

User's Manual

Computer on Module

COM Ports

Two USB Hosts

LCD

Ethernet

CompactFlash

MXM-6410 Windows CE 6.0 User's Manual

Version 2.04



Table of Contents

CHAPTER 1 MXM-6410/APC-6410 WINDOWS CE 6.0 FEATURES.....	5
1.1 BOARD SUPPORT PACKAGE (BSP)	5
1.2 DRIVERS	5
1.3 SERVICES	7
1.4 SPECIAL FEATURES	8
CHAPTER 2 SYSTEM INFORMATION.....	11
2.1 STARTING EVKM-MXM-6410.....	11
2.2 JUMPER SETTING	12
2.3 CONNECTORS.....	17
CHAPTER 3 USING WINDOWS CE 6.0	20
3.1 EBOOT	20
3.2. WINDOWS CE 6.0	26
3.2.1. <i>Setting the System Time</i>	26
3.2.2. <i>Touch Calibration</i>	27
3.2.3. <i>File System and Registry Basic</i>	28
3.2.4. <i>Networking Basics</i>	31
3.2.5. <i>Telnet Server</i>	33
3.2.6. <i>FTP Server</i>	38
3.2.7. <i>File Server</i>	46
3.2.8. <i>Web Server</i>	50
3.2.9. <i>Auto RUN</i>	57
3.2.10. <i>COM Ports</i>	58
3.2.11. <i>Software Installation</i>	60
3.3. CONFIGURE LCD PARAMETERS FOR DIFFERENT KINDS OF LCDS.....	62
3.3.1. <i>View Current LCD Parameters</i>	62
3.3.2. <i>Choose Default LCD Parameters</i>	63
3.3.3. <i>Set LCD Parameters of Different Types</i>	64
3.4. CONFIGURE GPIOs, BACKLIGHT_EN, LCD_PWREN AND VDDLCD_PWREN.....	67
3.4.1. <i>Configure GPIOs Setting at EBOOT</i>	67
3.4.2. <i>Configure GPIOs Setting at NK</i>	69
3.5. LOGO SPLASH SCREEN CUSTOMIZER	70
3.5.1. <i>Upload and save splash screen image into devices</i>	70
CHAPTER 4 USE MXM-6410/APC-6410 HARDWARE MFC MULTIMEDIA FUNCTION	75
4.1. WMV9 DECODER FOR WINDOWS MEDIA PLAYER.....	75
4.1.1. <i>Encode Video source as WMV9</i>	75

4.1.2. WMV9 Decoder for Windows Media Player	76
4.2. MFC DEVICE DRIVER'S API.....	76
4.2.1. CreateFile	77
4.2.2. DeviceIoControl.....	78
4.2.3. CloseHandle.....	79
4.2.4. Control Codes for DeviceIoControl()	80
4.2.5. Data Structure for Passing the IOCTL Arguments.....	83
4.3. OPENGLES USER'S MANUAL	84
4.3.1. Software Layers.....	84
4.3.2. Usage	86
4.3.3. Shader Compile.....	87
4.3.4. Dll location	87
4.3.5. Samples	87

Chapter

1

MXM-6410/APC-6410 Windows CE 6.0 Features

This Chapter gives an introduction to the Windows CE 6.0 features of MXM-6410 and APC-6410.

Section include :

- Board Support Package (BSP)
- Drivers
- Services

Chapter 1 MXM-6410/APC-6410

Windows CE 6.0 Features

This Chapter gives an overall picture in regarding to the Windows CE 6.0 features that Embedian provided with for the MXM-6410 computer on module and APC-6410 single board computer.

1.1 Board Support Package (BSP)

The Embedian Board Support Package for Windows CE 6.0 is one of the most advanced BSPs available on the market. Beside the standard Windows CE functionality, it includes a large number of additional drivers as well as optimized versions of standard drivers.

The Embedian bootloader and BSP were designed to be very configurable. This relieves the application-developer from the burden of creating an own image. Instead the necessary adaptations can be done by registry settings and adding files to the on-board flash file system.

The BSP is available at Embedian FTP site at

<ftp://ftp.embedian.com/pub/MXM-6410/WINCE60/>.

Customers who follow the readme instruction in the BSP folder for Microsoft's Platform Builder will enable you to build exactly the same Windows CE image as Embedian provides by default.

1.2 Drivers

The following drivers are integrated in the standard image that comes pre-installed with each MXM-6410 module or APC-6410 board.

Table 1.1 Drivers

Table 1.1	Drivers
Driver	Description
COM1	Support RXD/TXD/CTS/RTS
COM2	Support RXD/TXD
COM3	Support RXD/TXD
COM4	External UART that supports the full RS232 specification with all 9 signals.
COM5	External UART that supports the full RS232 specification with all 9 signals.
Ethernet	10/100Mbit driver for Davicom DM9000B
USB Host	4 Ports supporting mass storage devices (USB-stick, hard disc, CD-ROM...) and other devices as keyboard, mouse, USB hub...
USB OTG	Can be used as ActiveSync connection, mass storage or RNDIS
Audio	16Bit stereo output (up to 48kHz), mic input, line-In
Touch Screen	All 4-wire resistive screens supported
Display	All types of displays supported. Easy configuration in EBOOT
SD/SDHC	Memory cards, Wireless LAN, Bluetooth, GSM, GPS ...
CF	Memory cards, Wireless LAN, modem, Bluetooth, serial card, GPS, GSM, ...
GPIO	12 Configurable GPIO with Sample codes. Easy configuration at Eboot and NK
Flash File System	HIVE-based registry. Standard FAT as well as TFAT (transaction safe FAT) are supported
2D Engine	CPU 2D acceleration h/w supported
3D Engine	Support CPU 3D h/w acceleration. Support OpenGL ES 1.1/2.0 and follow the Khronos naming rules
TV-Out	TV output driver for NTSC/PAL format is supported
Camera	Camera drivers for various camera modules
MFC Engine	Support h/w H.263, H.264, VC1
SPI	High speed SPI supported
RTC	Real time clock driver

1.3 Services

The following services are integrated in the standard image that comes pre-installed with each MXM-6410 module or APC-6410 board. User can add their application oriented service from the BSP as well.

Table 1.2 Services

Table 1.2	Services
Service	Description
FTP Server	<i>Access device using the FTP File Transfer Protocol</i>
Telnet Server	<i>Access device using the Telnet protocol</i>
HTTP Server	<i>Access device using the HTTP protocol</i>
File Server	<i>Access device using the Samba protocol</i>
Time Server / Client	<i>Synchronize time with the network using the NTP protocol</i>
Shell	<i>Allow MXM-6410/APC-6410 to run batch file</i>

1.4 Special Features

In addition to the standard Windows CE functionality, Embedian has extended the operating system by following features:

Table 1.3 Special Features

Table 1.3	Special Features
Special Features	Description
EBOOT	Configuration through RS232/USB, clear flash registry, download image to RAM, download image to Flash, format flash, set/save configured structures, define initial GPIO configuration and set LCD parameters. Debug Functionality: erases flash range, write/read 32bit value at address.
Direct SD Boot	Boot Directly from SD card. It is very useful for maintenance purposes.
LCD Customizer	Set all types of LCD resolution parameters in EBOOT and pass to NK.
Splash Screen	Display default start-up screen while Windows CE is booting.
ActiveSync	Support Microsoft Active Sync. via USB 2.0 or serial port
Hive-Based Registry	Support Hive-Based Registry
AutoStart	Automatically execute programs upon system boot or upon removable media insertion
AutoCopy	Automatically copy files to the RAM file system upon system boot or upon removable media insertion. This, for example, can be used to add items to the Windows CE START menu button.
GPIO Customizer	Each GPIO can be configured to act as input, output, high, low or an interrupt source at EBOOT and NK.
Flushing Registry on Changes	Registry will be saved to Flash immediately on changes. Registry will be kept anytime even when power failure.
SplashScreen Customizer	Show your own screen while Windows CE is booting

<i>NAND Flash Writer</i>	<i>To recover the firmware in NAND flash from NOR flash or SD by jumper setting.</i>
<i>Registry Editor</i>	<i>To edit registry locally or remotely</i>
<i>Remote Display</i>	<i>To see device desktop and control it without panel attached</i>

Chapter

2

System Information

This Chapter gives an introduction of system information to let users quickly pick up how to setting up the system.

Section include :

- System Setup
- Jumper Setting
- Firmware architecture in NAND flash

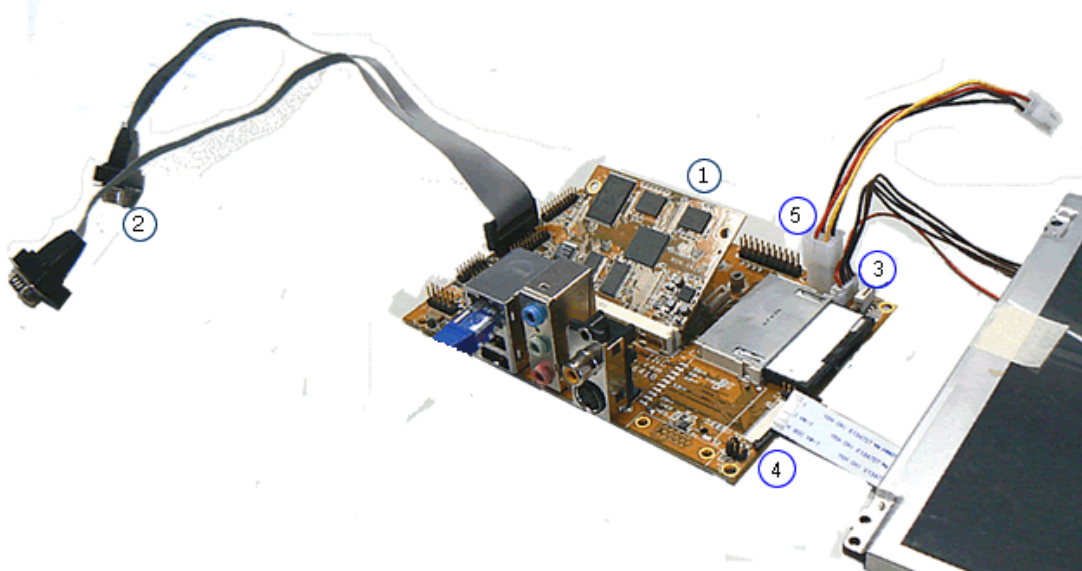
Chapter 2 System Information

This Chapter gives an introduction of hardware information so that users can quickly set up the system.

2.1 Starting EVKM-MXM-6410

Before starting, please check the jumper to make sure that the jumper configuration is matching your demand. Figure 2.1 shows the steps.

Figure 2.1 Starting EVKM-MXM-6410



1. Press the MXM-6410 module and screw it tightly.
2. Connect the console port (UART0, CN20 on figure 2.3) to the console cable, then to a null modem cable and then connect to your PC. Open up the DNW 0.6c (Click “**Connect**” at “**Serial Port**” tab) or Hyper Terminal and set the baud rate to 115200, 8N1, no hardware flow control.
3. Connect the backlight invert board power to CN9.
4. Connect the LCD FPC Cable (Top Contact) or LVDS cable.
5. Connect the power cable to power board.

2.2 Jumper Setting

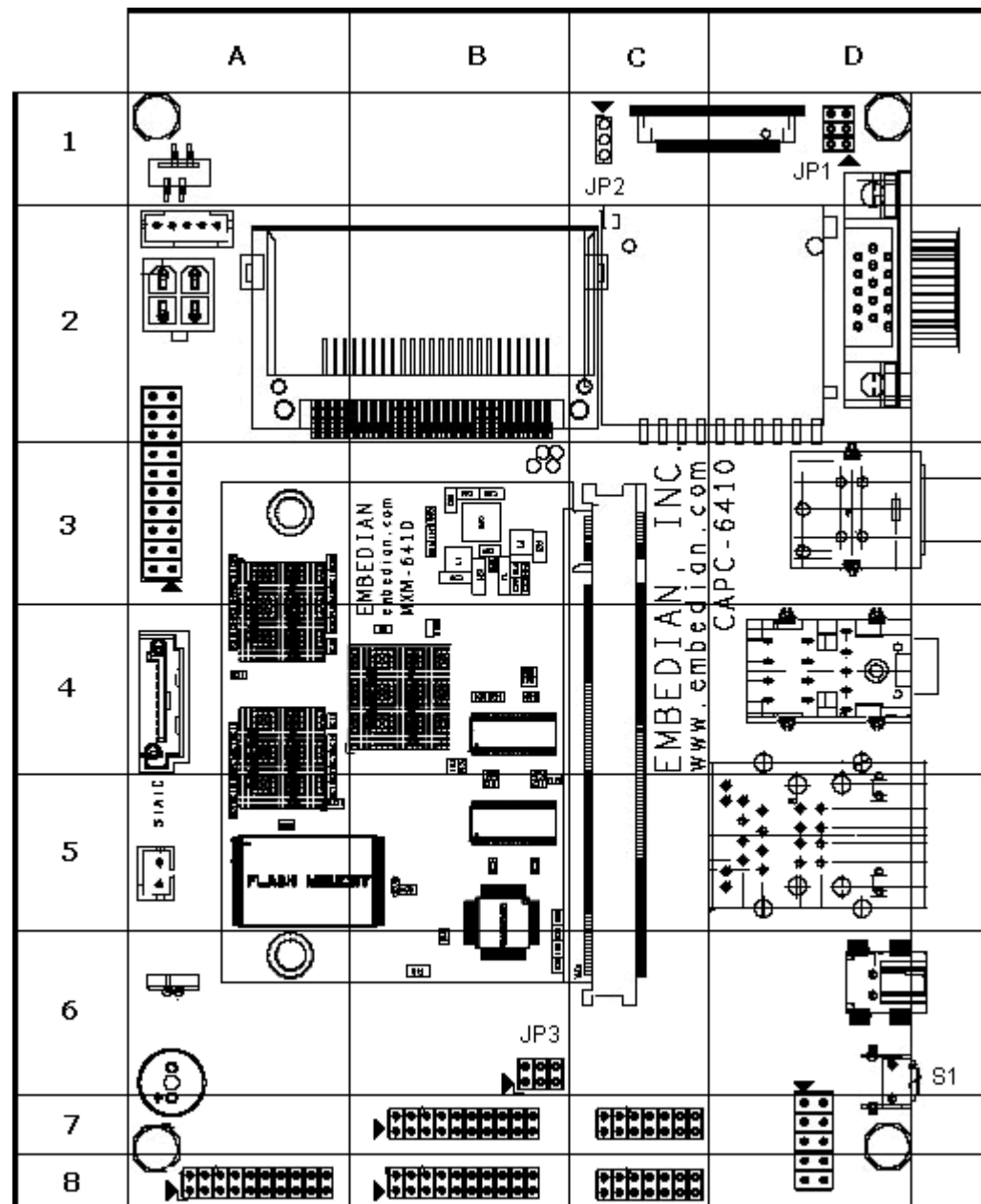
This section gives details of the hardware features of the EVKM-MXM-6410. These include a description of the switches, jumper settings, connectors and connector pin outs.

Jumpers

The EVKM-MXM-6410 SBC has a number of jumpers that allow you to configure your system to suit your application. All use 2mm shorting blocks (shunts) to select settings. Turn off power of the EVKM-MXM-6410 before changing the position of a shunt.

Jumper Location

Figure 2.2 Jumper Location



List of Jumpers

The table below lists the function of various jumpers.

Table 2.1 Jumpers

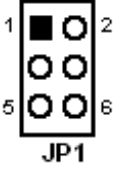
Table 2.1	Jumpers
Label	Function
JP1	NOR boot or NAND boot Setting and LCD Scan Direction Setting
JP2	LCD Power Setting (3.3V or 5V for both TTL and LVDS)
JP3	RS232/422/485 Setting
S1	Reset Button

Jumper Settings

The following tables describe how the jumper shunts to various configurations.

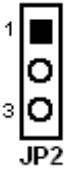
JP1: Location on Board, D1

Table 2.2 NOR Boot or NAND Boot and LCD Scan Direction Setting (JP1)

Table 2.2	NOR Boot or NAND Boot and LCD Scan Direction Setting	
	Setting	Function
	N.C.(Default)	NAND Boot
	JP1 (1-2)	NOR Boot
	JP1 (3-4)	Pull-High of CN14 Pin 30
	JP1 (5-6)	Pull-High of CN14 Pin 31

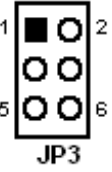
JP2: Location on Board, C1

Table 2.3 LCD VCC Power Setting for CN14 and CN15 (JP2) (from CPU LVDS and TTL)

Table 2.3	LCD VCC Power Setting for CN14 (JP2) (from CPU LVDS and TTL)	
	Setting	Function
	JP2(1-2)(Default)	3.3V
	JP2(3-4)	5V

JP3: Location on Board, B6


Table 2.4 RS232/422/485 Mode Setting

Table 2.4		RS232/422/485 Setting	
		Setting	Function
		JP3(1-2)(Default)	RS232
		JP3(3-4)	RS422/RS485 half duplex
		JP3(5-6)	RS422/RS485 full duplex

Note: If JP3 jumper set to be RS232, the RS232 connector will be CN20 (UART1) header. If JP3 jumper set to be RS422 and RS485, the connector will be CN22 header. User can only choose one (RS232 or RS422 or RS485) at the same time.

S1: Location on Board, D6

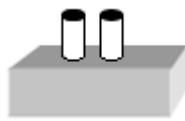
Table 2.5 Reset Button

Table 2.5		Reset Button	
		Setting	Function
		Press Button and Release Immediately	Reset CPU and IO

Setting Jumpers

You configure your board to match the needs of your application by setting jumpers. A jumper is the simplest kind of electric switch. It consists of two metal pins and a small metal clip (often protected by a plastic cover) that slides over the pins to connect them. To "close" a jumper you connect the pins with the clip.

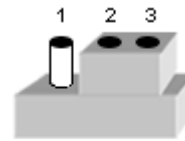
To "open" a jumper you remove the clip. Sometimes a jumper will have three pins, labeled 1, 2 and 3. In this case you would connect either pins 1 and 2 or 2 and 3.



Open



Closed



Closed 2-3

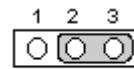
The jumper settings are schematically depicted in this manual as follows.



Open



Closed



Closed 2-3

A pair of needle-nose pliers may be helpful when working with jumpers. If you have any doubts about the best hardware configuration for your application, contact your local distributor or sales representative before you make any change.

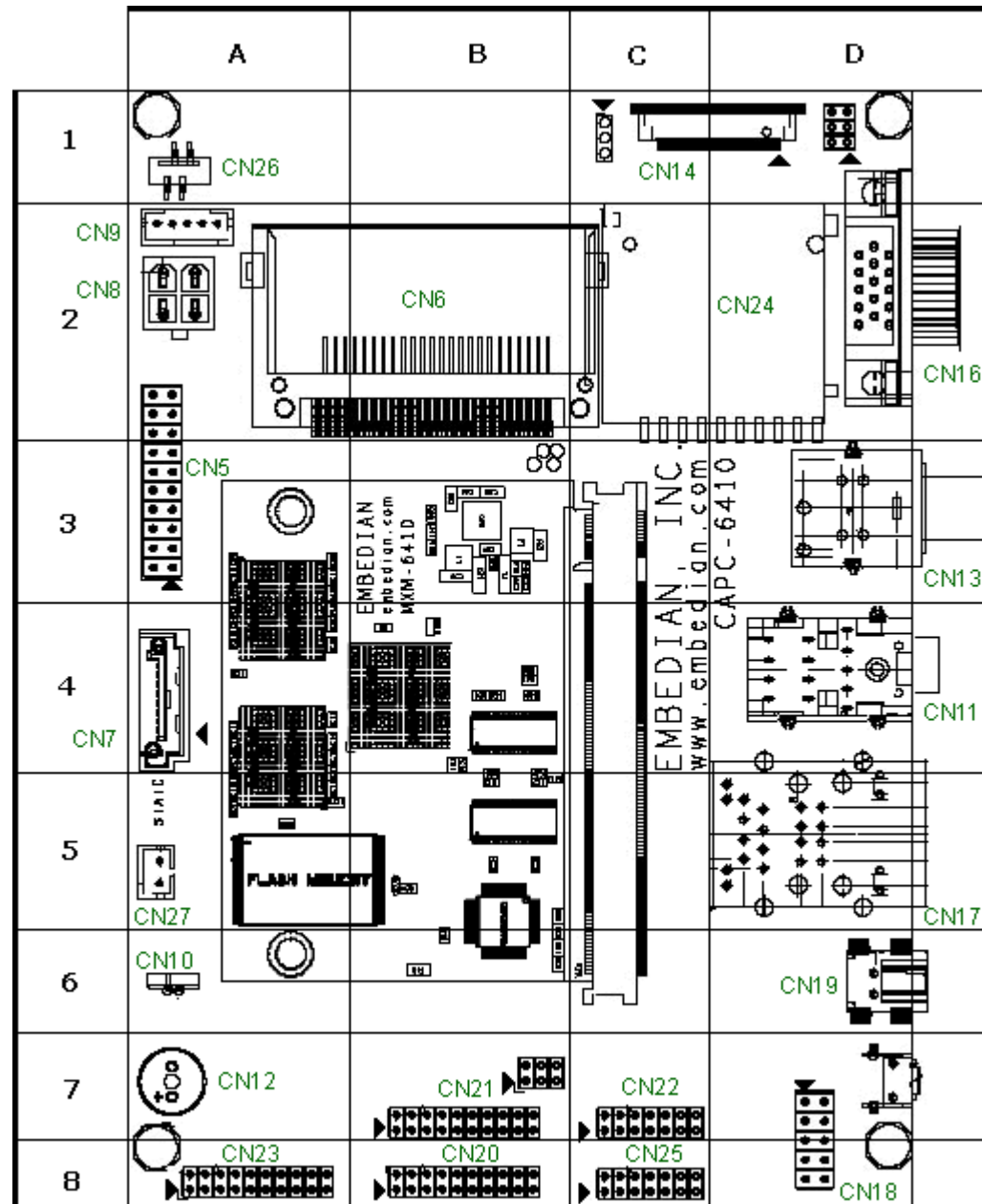
Generally, you simply need a standard cable to make most connections.

2.3 Connectors

Onboard connectors link the EVKM-MXM-6410 to external devices such as LCD panel, a keyboard, an audio headset or CompactFlash and to external communication such as 802.11, USB or Ethernet link. The table below lists the function of each of the board's connectors.

Connector Location

Figure 2.3 Connector Location



List of Connectors

Table 2.6 List of Connectors

Table 2.6	Connector
Label	Function
CN5	JTAG
CN6	CF Type I/II Connector
CN7	SATA Connector
CN8	Power Connector
CN9	LCD Backlight Inverter Connector
CN10	RTC Battery Connector
CN11	Audio (Microphone in, Headphone out, Line in) Connector
CN12	Buzzer
CN13	S-Video and CVBS Video Connector
CN14	TTL Level LCD FPC Connector
CN15	LVDS LCD Connector
CN16	VGA DSub-15 Connector
CN17	Ethernet RJ45 and Double USB Host Type A Connector
CN18	Double USB Host Header
CN19	USB OTG Type B Connector
CN20	UART 0 and UART 1 Header
CN21	UART 2 and UART 3 Header
CN22	SPI and RS422/485 Header
CN23	UART 4 and UART 5 Header
CN24	SD/SDHC Connector
CN25	GPIO Header
CN26	4-Wire Touch Panel Connector
CN27	CAN Bus Connector

For more details, users can refer to the hardware manual..

Chapter

3

Using Windows CE 6.0

This Chapter details how to use the Windows CE 6.0 of MXM-6410 computer on module or APC-6410 single board computer.

Section include :

- EBOOT
- Windows CE 6.0
- Configure LCD parameters for different kinds of LCDs

Chapter 3 Using Windows CE 6.0

This chapter gives three topics to introduce how to use the Windows CE 6.0 system on MXM-6410 computer on module or APC-6410 single board computers. The first is to introduce the EBOOT menu configuration and the second is to introduce the Windows CE 6.0 systems. The last section will teach users how to set the parameters for different kinds of LCDs without re-building from the sources or PB (Platform Builder).

Before going on, users need to install the “wceusbsh” USB drivers that Embedian provided with on your Windows PC first. It is the Windows CE USB RNDIS driver that is required for active sync. (When plug the USB cable to the Windows PC at first time, Windows will ask you to install the driver. Just point to the directory where your USB driver placed and the installation will be done.) Connect the console port to your Windows PC as described in section 2.1. Make sure that the jumpers are properly configured.

3.1 EBOOT

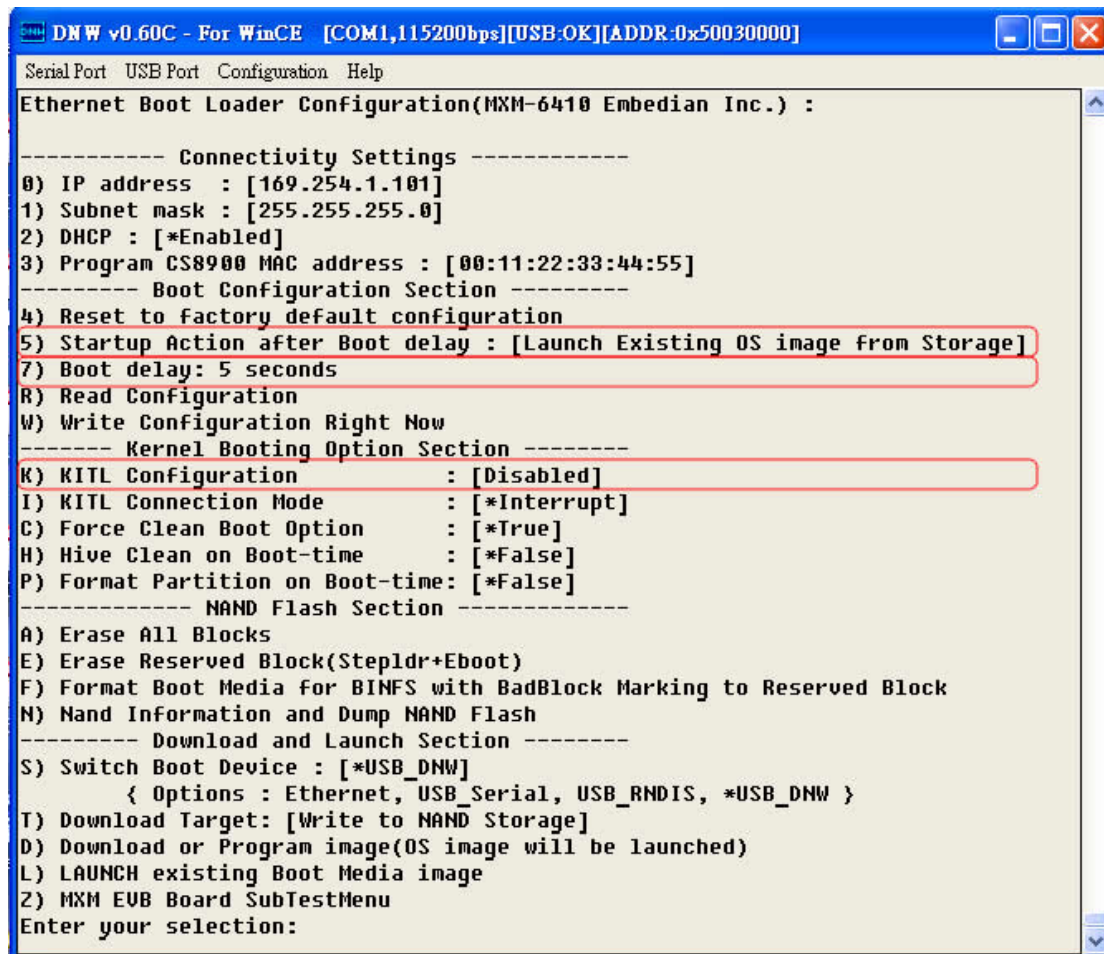
Turn on the power of the device. You should be able to hear a short beep. If you attached a LCD to the device, you will be able to see the splash screen as in figure 3.1.

Figure 3.1 EBOOT Splash Screen



To enter EBOOT menu configuration, press [**Space**] bar of your keyboard when booting. And you will see the following screen as shown in figure 3.2.

Figure 3.2 EBOOT Menu Configurations



You need to make sure that the following two items have been configured correctly.

First

Press "5" to change from

"5) Startup Action after Boot delay : [*Download New image]"

to

"5) Startup Action after Boot delay : [Launch Existing OS image from Storage]"

This is to tell EBOOT to load the Windows CE 6.0 NK.bin image from NAND flash to DDR RAM.

Second

Press "K" to disable the KITL configuration. That is to change from

"(K) KITL Configuration : [*Enabled]"

to

"(K) KITL Configuration : [Disabled]"

Users can also set up the boot delay interval between EBOOT and NK by pressing “7”.

Third

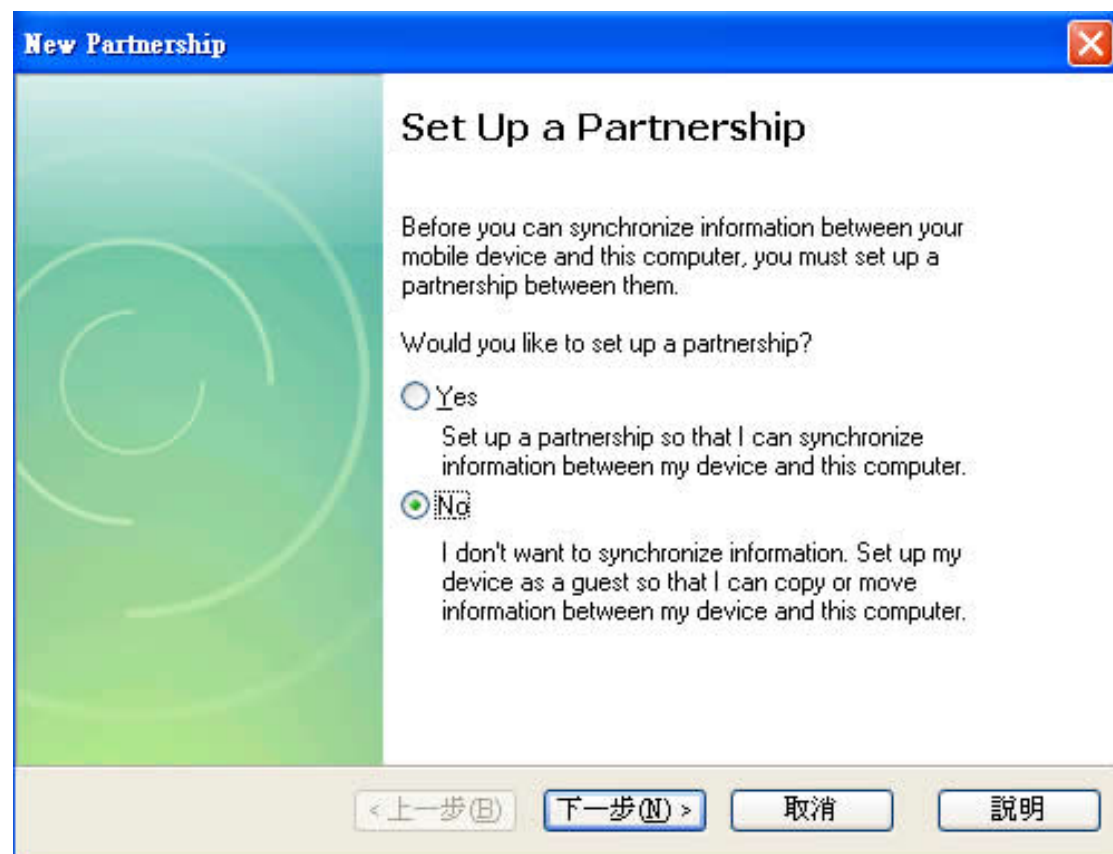
Before exit EBOOT menu, remember to press “W” to save the new configuration.

The Ethernet connectivity settings in EBOOT menu is for setting the network parameters for Ethernet download images from Platform Builder. User need to choose Ethernet for both download and transport protocols in their PB target connectivity options. And click on the setting button next to download and select the device when it shows up in the “Active Target Devices” window. But Embedian suggests users use USB to download images because of the download speed is faster. The IP address setting here will not be the same as Windows CE 6.0 NK image that is stored as HIVE registry.

After saving the EBOOT configuration by pressing “W”, users can reset the system or press “L” to launch the Windows CE 6.0 NK.bin image from NAND flash.

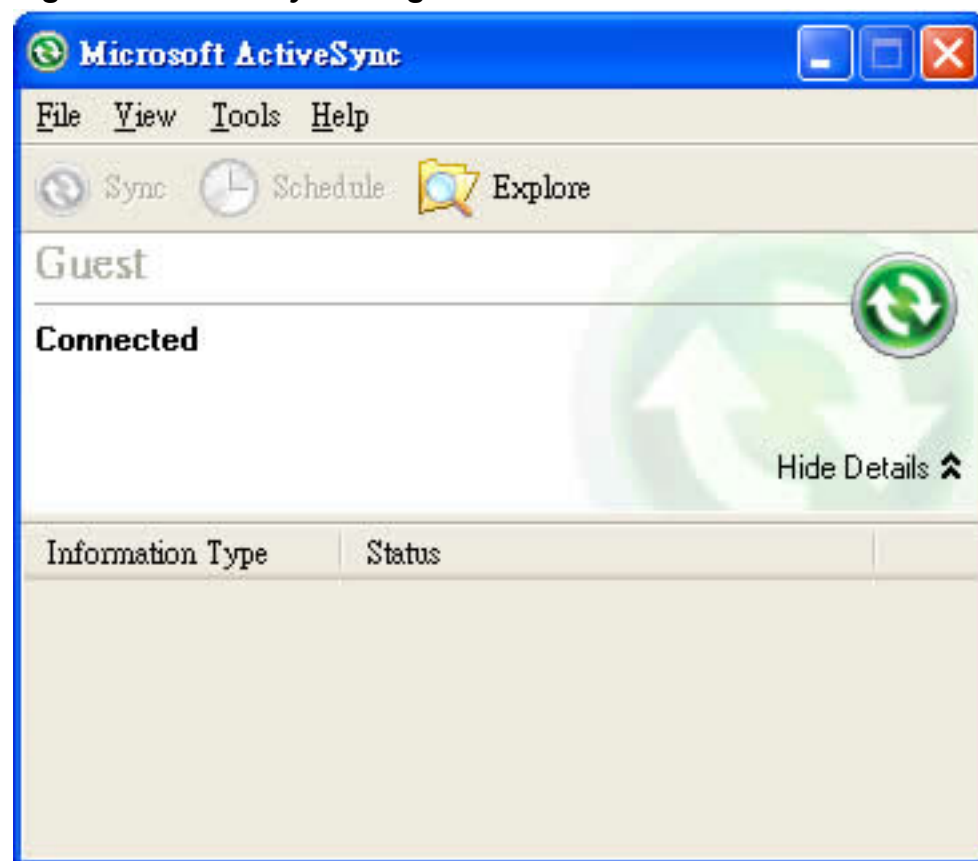
If you have ActiveSync program (user can download this program for free from Microsoft’s website) installed in your Windows PC and the USB cable is connected to the device, you should be able to see the ActiveSync program pop out and ask you to set up a partnership during loading the NK image as shown in figure 3.3.

Figure 3.3 ActiveSync Program



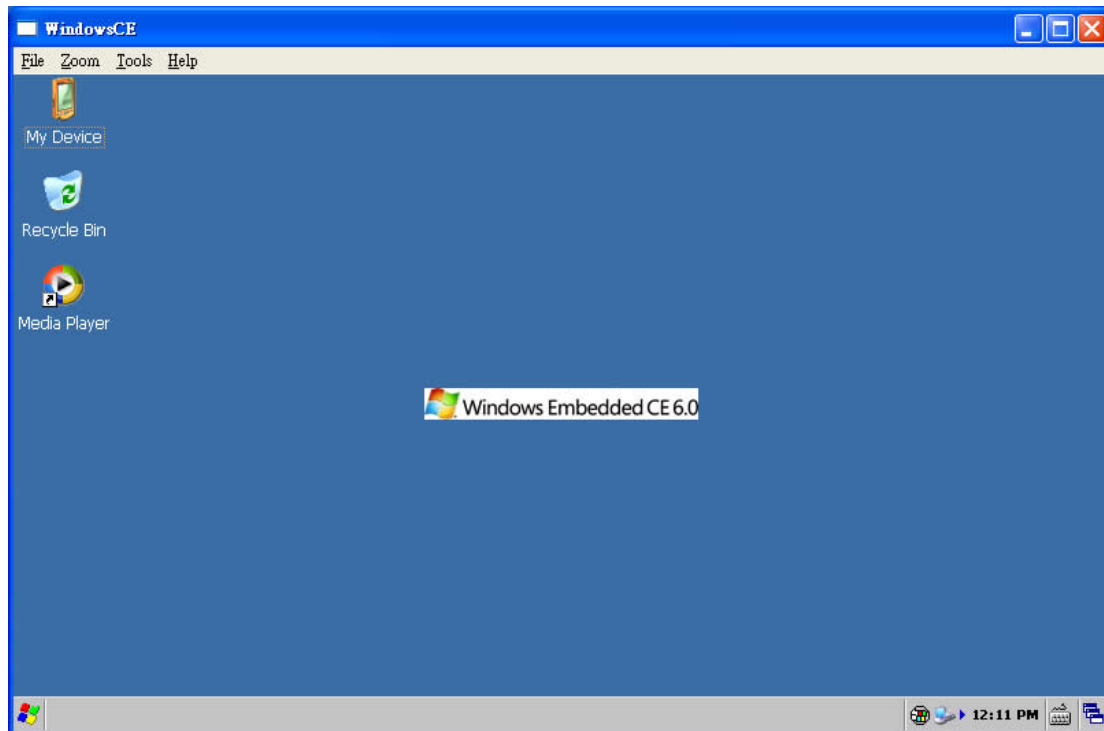
Select "**No**" and click "**Next**" of your ActiveSync program. You will see the screen shown as in figure 3.4.

Figure 3.4 ActiveSync Program



Users should be able to see the Windows CE 6.0 desktop on the LCD screen or remote tools as shown in figure 3.5.

Figure 3.5 Windows CE 6.0 Desktop



Next section, we will introduce the Windows CE 6.0 system.

3.2. Windows CE 6.0

This section will introduce the Windows CE 6.0 system.

3.2.1. Setting the System Time

The APC-6410 includes a battery-backed real-time clock.

To set the System Time:

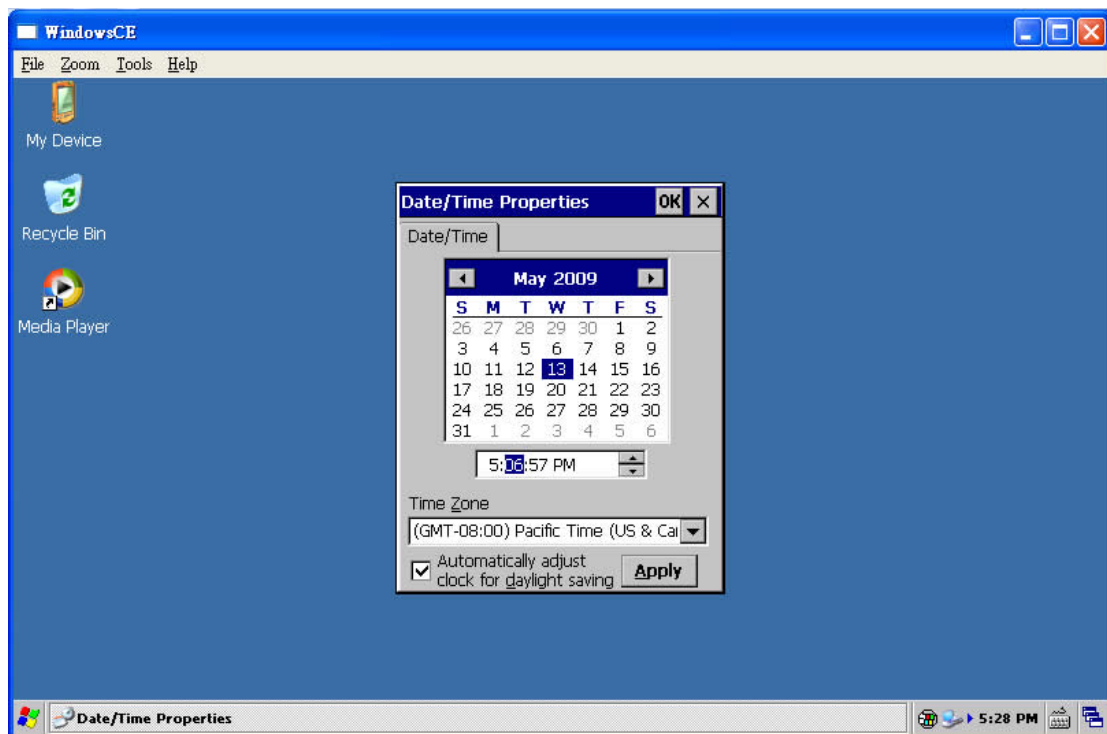
1. Double-tap on the time icon at the bottom right of the taskbar (shown below in figure 3.6).

Figure 3.6 Windows CE Time Icon



2. Within the Date/Time Properties Dialog (shown below in figure 3.7), set the current date and time then press the OK button to save these settings.

Figure 3.7 Date/Time Properties Dialog



APC-6410 comes with a RTC battery pack. When the external DC power is exist, the RTC battery will be charged. And when the external DC power is removed, the RTC battery will supply the RTC to keep the time for few days. It is therefore; users don't have to set up the system time every time when the external DC power is temporarily gone.

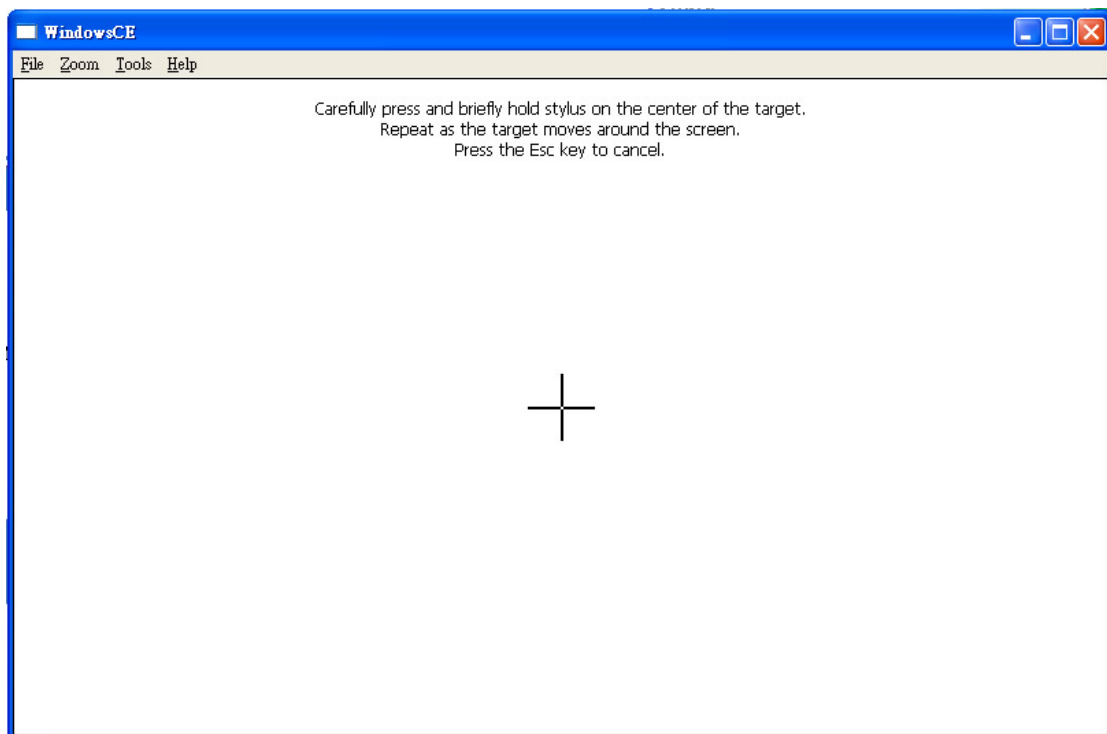
3.2.2. Touch Calibration

If your MXM-6410/APC-6410 device is attached to a touch screen, you will need to calibrate your touch screen at first boot. The calibration value will be stored as HIVE. That means you don't need to calibrate again at next boot.

To calibrate your touch screen:

1. Go to **Start->Settings->Control Panel->Stylus**
2. Choose **"Calibration"** Tab and click on **"Recalibrate"**. You will see below as shown in figure 3.8.

Figure 3.8 Calibrate touch panel



3. Carefully press and briefly hold stylus on the center of the cross target. Repeat as the cross target move around the screen.

3.2.3. File System and Registry Basic

3.2.3.1. Making Files Persistent

Files written or copied to **\NandFlash** are persistent from one boot to the next. Files written to **SD** memory cards (files in **\Storage Card**) or USB Compact Flash cards (files in **\Hard Disk**) are also persistent. Files written anywhere else in the system at run time are volatile.

3.2.3.2. Re-formatting the NAND Flash Folder

\NandFlash default has two partitions Part00 and Part01. Part00 is formatted as a BINFS filesystem by default and stores the system firmware including of NK.bin and boot HIVE information. Part01 is formatted as FAT filesystem by default and is reserved for user storage. NAND Flash would be automatically formatted and partitioned and mounted during the boot sequence. It would also create the "Documents and Settings" directory in the **\NandFlash** folder which is required for Hive-based Registry support (system.hv and user.hv) as explained later.

Manually, **\NandFlash** can be reformatted as follows. (Note that in case

the Part00 of the \NandFlash folder is manually formatted, users need to go to eboot menu to recover the NK at next boot.)

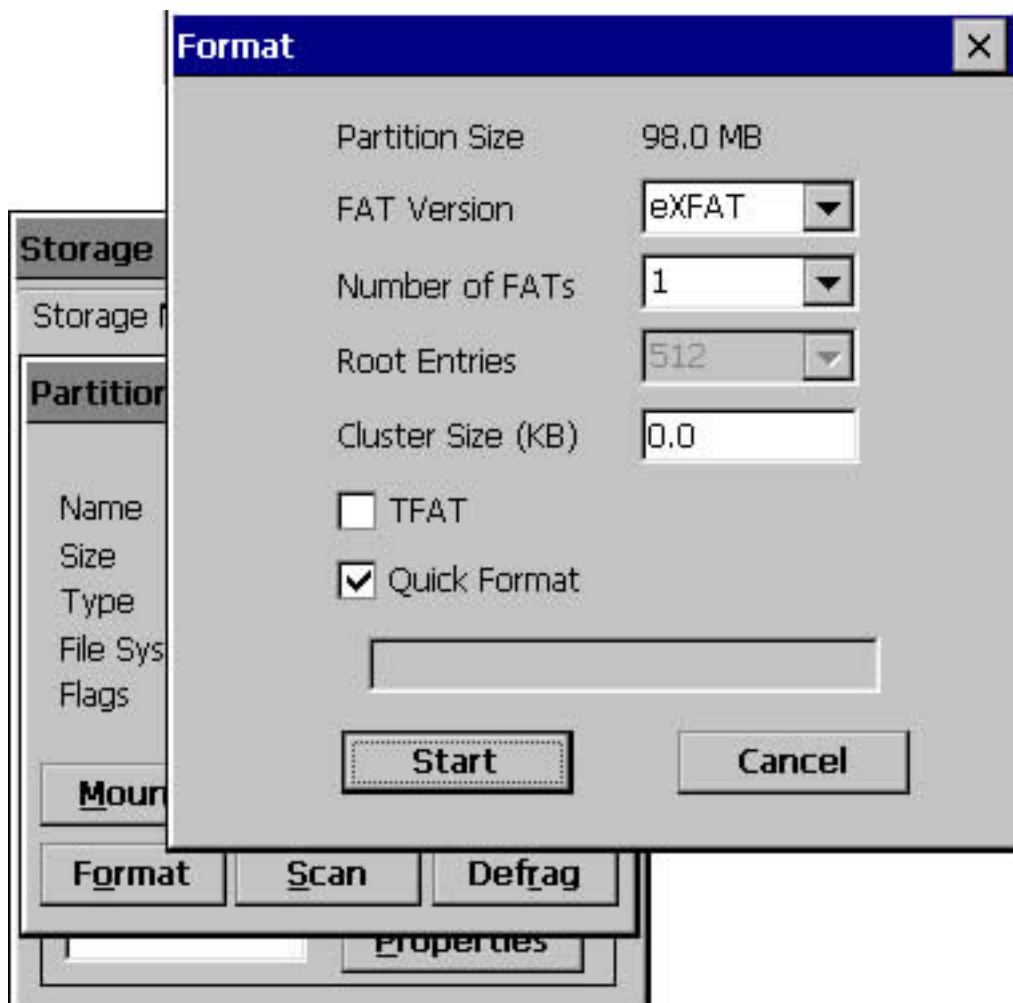
1. Go to **Start->Settings->Control Panel->Storage Manager**
2. Choose the one label with “DSK1: Microsoft Flash DISK” from **Store Info** and choose “Part01 *” (The asterisk sign represents that this partition has been mounted.) from **Partitions**.

Figure 3.9 Storage Manager Control Panel



3. Click **Properties**.
4. Click **Dismount**.
5. Click **Format**.

Figure 3.10 Format Part01 of NAND Flash



6. Click **Start**.
 7. Click **Yes**.
 8. Click **OK**.
 9. Click **Mount**
 10. Click **OK**
 11. Exit out of Storage Properties by Clicking **OK**.
- The user partition of NAND flash has been formatted now.

3.2.3.3. Persisting the Registry

WinCE Hive-Based Registry has been well implemented. The hive-based registry stores the registry data in files, or hives, which can be kept on any file system. This removes the need to backup and restore on power off. For detailed information, please refer to *Windows CE 6.0 Help* on relating topic. Embedian's CE 6.0 HIVE-based design will be automatically flushing the registry on changes. Users don't need

to flush registry manually.

All the hive-based registries are configured to be stored in **\NandFlash\Documents and Settings**. So if the **\NandFlash** folder is manually formatted, all the registry settings would be gone and it would be restored back to the default registry after the next reboot. Therefore, it is best to reboot the device if you have formatted the **\NandFlash** folder before proceeding with any operation.

To restore the default registry (it may interfere with software development), you have to go into the EBOOT, and press “4) Reset to factory default configuration” or just re-flash the NK again.

3.2.4. Networking Basics

3.2.4.1. Getting the IP Address

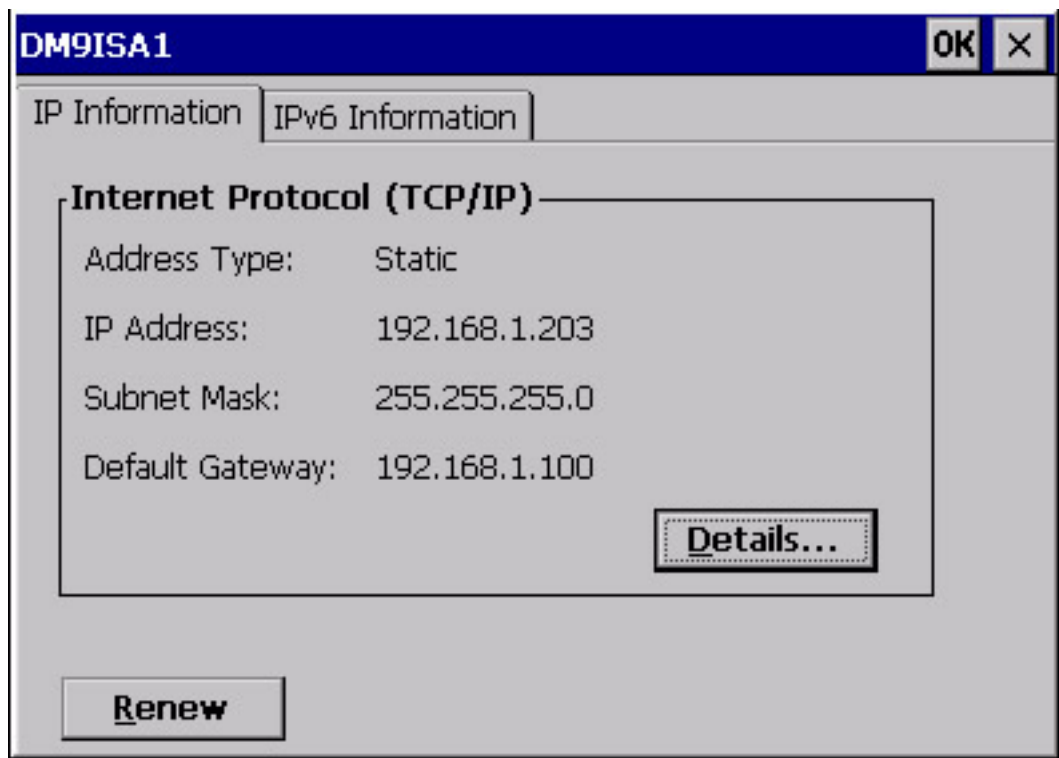
To view your current IP address, simply double click on the network icon located on the MXM-6410/APC-6410 as shown in figure 3.11.

Figure 3.11 Network Icon



After double-clicking on this Icon a dialog box will be shown with available IP information as shown in figure 3.12.

Figure 3.12 IP Information Dialog



3.2.4.2. Setting Up IP Address or DHCP

If networking hardware such as a hub or switch is available, an Ethernet patch cable can be used to connect the MXM-6410/APC-6410 to this hardware. Because of MXM-6410/APC-6410 support AUTO-MDIX Ethernet protocol, there is no need of crossover cable even it is direct link. Setting static addresses or DHCP on both MXM-6410/APC-6410 is described below. Once the addresses are properly set, the host PC will be able to communicate with the device.

On the MXM-6410/APC-6410 desktop, tap

Start->Settings->Network and Dial-up Connections.

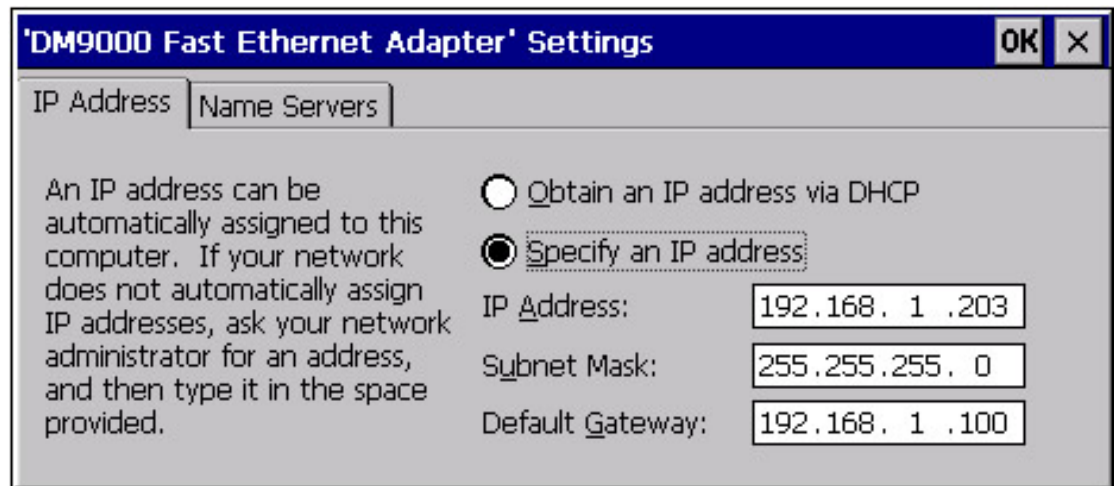
Double-tap the network connection icon (named after the DM9ISA1 driver used) to open the **Network Settings** dialog box.

Figure 3.13 Network Connection Icon



Change the appropriate settings and tap **OK**. Note that the settings will take effect immediately.

Figure 3.14 DHCP or Static IP Address Setting



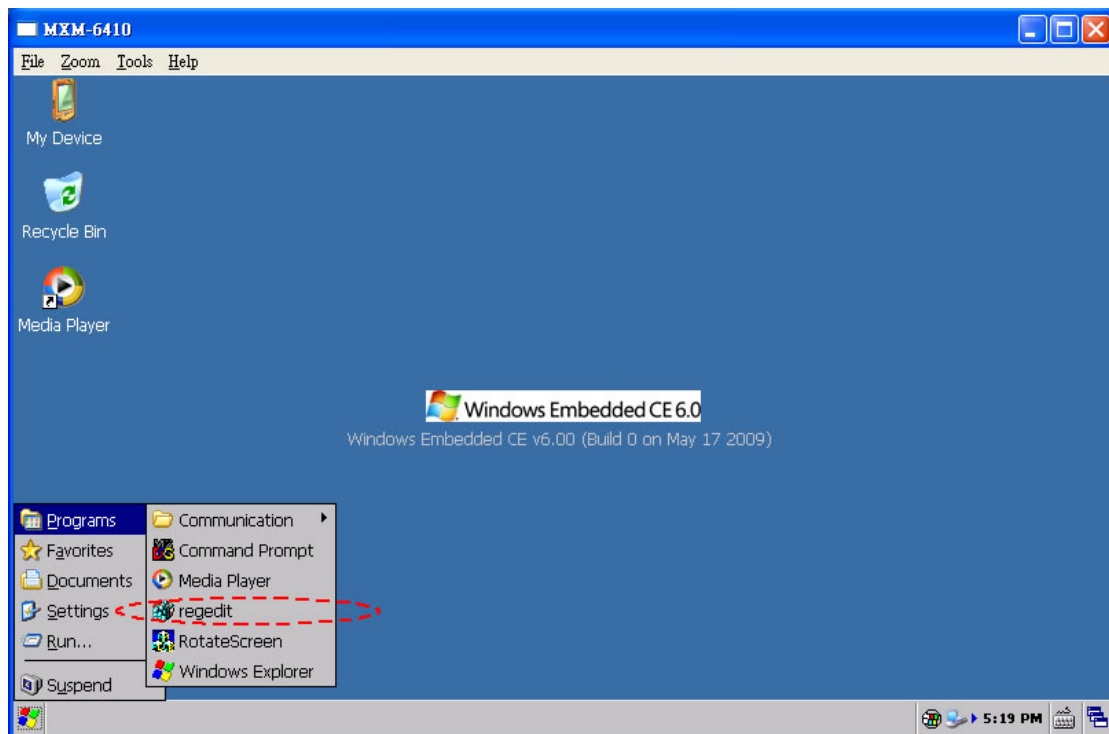
Once the IP has been modified, it will be saved to HIVE registry and the IP and Ethernet configuration will be kept at next boot.

3.2.5. Telnet Server

The Telnet server in the factory-installed image is configured with no access restrictions. You could disable the Telnet server or restrict access to it before deploying the product or connecting to an unsecured network. The Telnet server is not started on MXM-6410/APC-6410 hardware at boot by default. To turn on the telnet service, users need to enable the "*IsEnabled*" registry key setting under the **HKEY_LOCAL_MACHINE\Comm\TELNETD** registry key. Figure 3.15 and figure 3.16 show how to enable the telnet service.

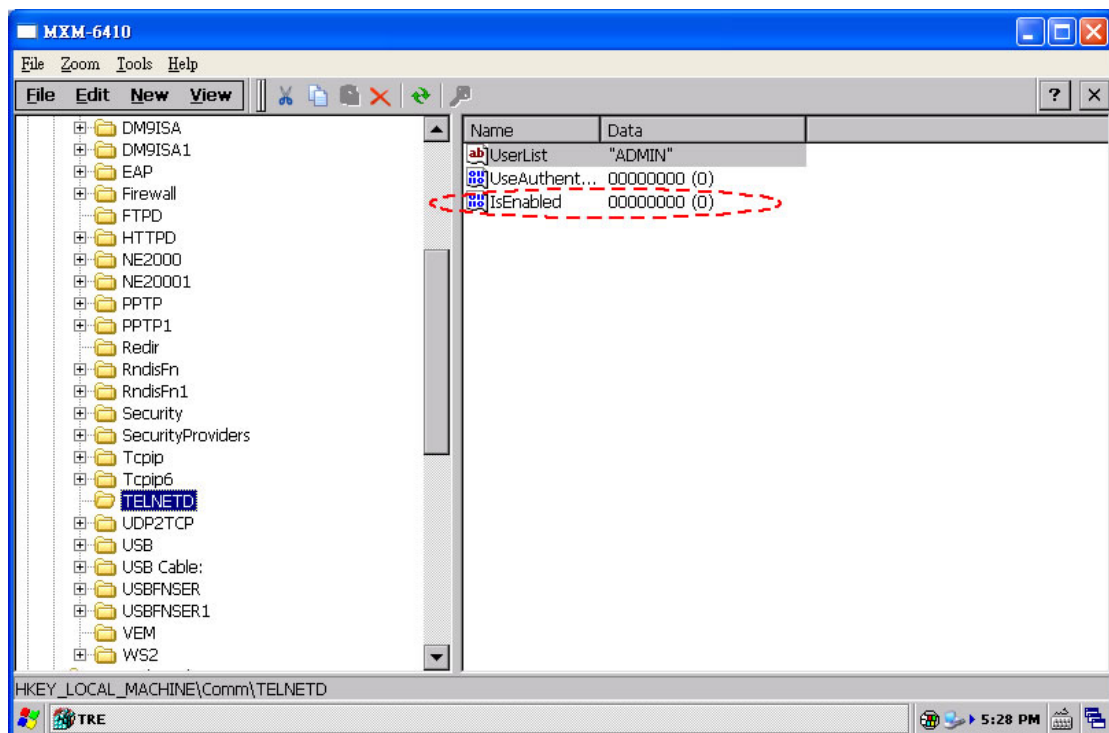
Go to **Start → Programs → regedit**

Figure 3.15 regedit Utility



The telnet registry settings are under the **HKEY_LOCAL_MACHINE\Comm\TELNETD** registry key. Double click the *IsEnabled* registry key set it from “0” to “1”.

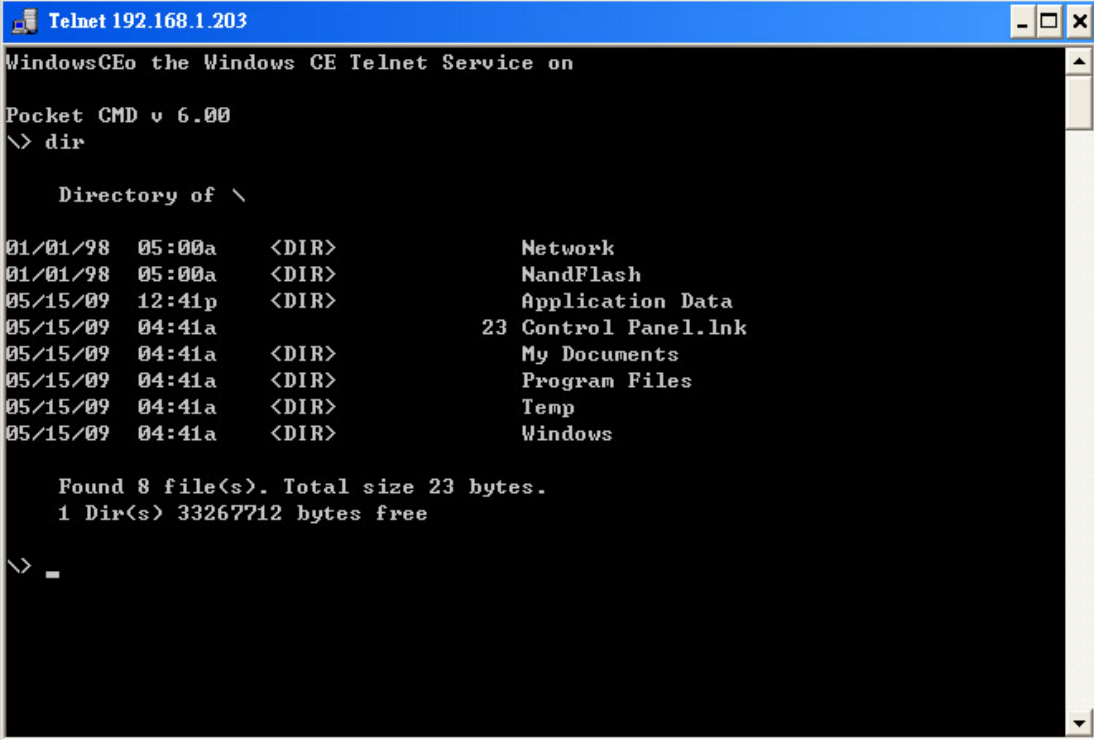
Figure 3.16 Enable Telnetd Service



Note: You will need to reboot the device to take effect the registry key enabled.

To access the Telnet server on a MXM-6410/APC-6410 device, simply enter *telnet <Your Device IP Address>*. See the example below.

Figure 3.17 Telnet Prompt



```
Telnet 192.168.1.203
WindowsCEo the Windows CE Telnet Service on
Pocket CMD v 6.00
\> dir

    Directory of \

01/01/98  05:00a  <DIR>      Network
01/01/98  05:00a  <DIR>      NandFlash
05/15/09  12:41p  <DIR>      Application Data
05/15/09  04:41a  23 Control Panel.lnk
05/15/09  04:41a  <DIR>      My Documents
05/15/09  04:41a  <DIR>      Program Files
05/15/09  04:41a  <DIR>      Temp
05/15/09  04:41a  <DIR>      Windows

    Found 8 file(s). Total size 23 bytes.
    1 Dir(s) 33267712 bytes free

\> _
```

Note: If you used telnet client like putty program to telnet in, the first few commands might give you a “command not found” message. Just a couple of trials will be fine. But Windows telnet command prompt will not appear to see this.

3.2.5.1. Default Registry Setting

It's necessary to be aware of registry settings that impact security. The telnet registry settings are under the

HKEY_LOCAL_MACHINE\Comm\TELNETD registry key.

Telnet server reads the values in the registry before serving each request. Therefore, changes made to the registry take affect immediately and do not require the telnet server to be restart. The registry change will be kept on next reboot.

Table 3.1. TELNETD Registry Value

HKEY_LOCAL_MACHINE\Comm\TELNETD	
Value	Description
IsEnabled : REG_DWORD	Indicates if the Telnet server is enabled. Set this value to a non-zero value to enable the server, and to 0 to disable the server. If the sample Telnet server is started and this value has not been set, it defaults to accepting connections.
UseAuthentication: REG_DWORD	To require a password check on the user, set this value to 1; otherwise, set it to 0. By default, the Telnet sample requires authentication.
UserList : String	Provides a comma-separated list of allowed users. Requires UseAuthentication to be enabled.

3.2.5.2. Changing the Telnet Server Access Privileges

To change the Telnet server access privileges, the Registry Editor tool under the **Windows** folder (or **Start → Programs → regedit**) will be used.

To modify the Telnet server access privileges:

Step 1 Click the regedit.exe located at \Windows folder.

Step 2 Browse to the **HKEY_LOCAL_MACHINE\Comm\TELNETD**.

Step 3 Ensure that the **IsEnabled** value is **1**.

Step 4 Ensure that the **UseAuthentication** value is **0**.

Step 5 Set the **UserList** value to a semi-colon separated list of users. (For example, **user1;user2**).

Refer to the topic **Telnet Server User Lists** in the Platform Builder documentation for more detailed information on this subject.

Examples:

[HKEY_LOCAL_MACHINE\COMM\TELNETD]

UserList="GladysL;AnnD;-KimY;@SomeGroup;-@villains"

Allows GladysL, AnnD, and the SomeGroup group, but restricts the KimY user and the villains group.

The asterisk or the at sign and the asterisk (* or @*) allows all users, and the combination of the hyphen and the asterisk, or the combination of the hyphen and the at sign and the asterisk (-* or -@*) denies all users.

The string is interpreted sequentially. In other words, if GladysL is member of the Finance group, the following strings will allow GladysL.

GladysL

@ Finance

@ Finance;GladysL

GladysL;@ Finance

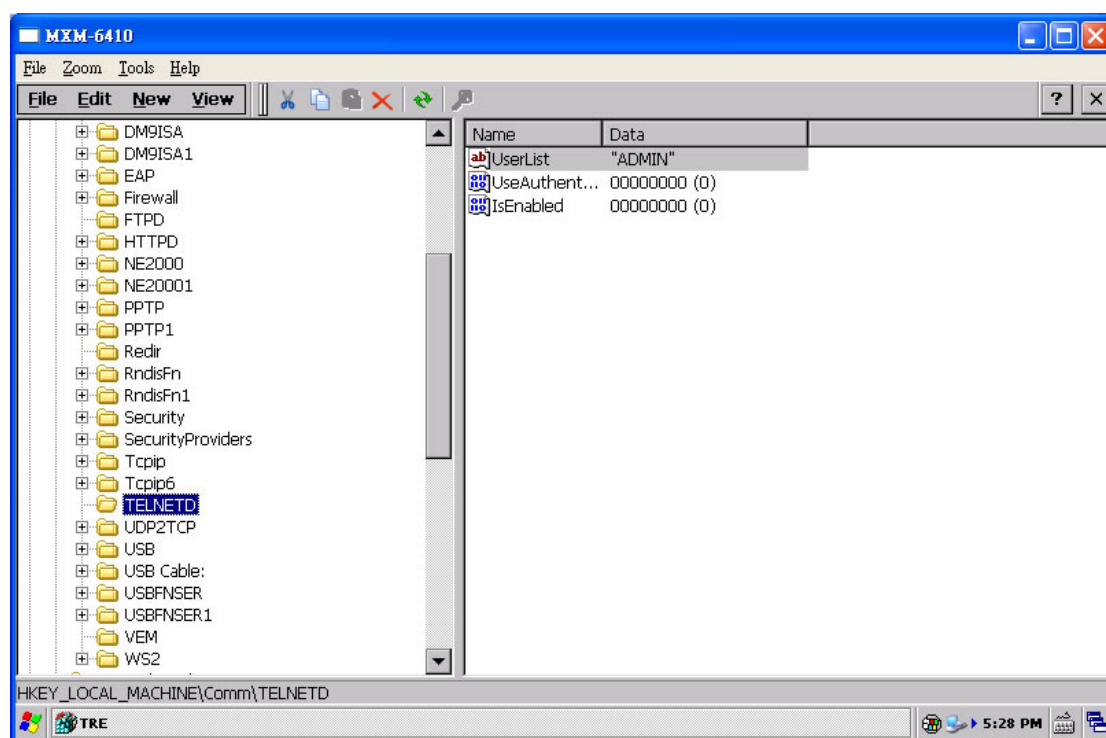
@ Admins;-GladysL

GladysL;-@ Finance

GladysL;-GladysL

*;-GladysL;-@ Finance

Figure 3.18 Telnet Registry



3.2.5.3. Security Note

The security on the Telnet server is very light and vulnerable to security attacks. Even if the Telnet server is configured to require password authentication, the password is sent in plain text across the network and is therefore vulnerable to packet sniffing. A malicious user could obtain the password to MXM-6410/APC-6410 by watching packets sent back

and forth between the Telnet server and the client during the authentication stage. If a malicious user could log on to the device, they would have completed over it. This could involve deleting or modifying key system files and the registry.

Because of these serious security risks, it is strongly recommended you only run the Telnet server for development and debugging purposes, on a controlled, private network where you trust the users. It is strongly recommended that you do not deploy this Telnet server on a public network such as the Internet.

3.2.5.4. Recommendations

Set the user list and domain variables to prevent hacker attacks on your device. If Telnet server is used without appropriate values set for the User List and Domain variables, your Telnet server will be vulnerable to hacker attacks.

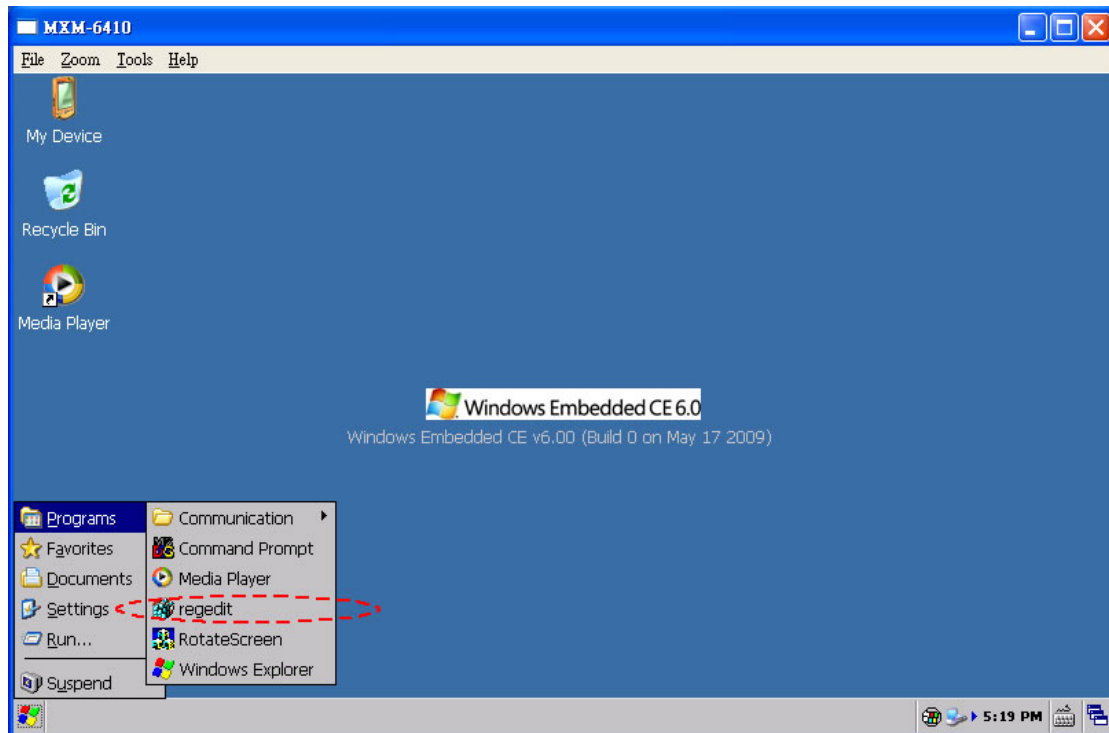
To prevent such attack, the user name is the **UserList** registry value must be set for each of the servers that are currently running. The user will then need to log in with the specified user name and appropriate password to use the server. You can also set the domain variable in the **DefaultDomain** registry value, which is located under **HKEY_LOCAL_MACHINE\Comm\Redir** registry key.

3.2.6. FTP Server

The FTP server in the factory-installed image is configured with no access restrictions. You should disable the FTP server or restrict access to it before deploying the MXM-6410/APC-6410 or connecting to an unsecured network. The FTP server is not started on MXM-6410/APC-6410 hardware at boot by default. To turn on the FTP service, users need to enable the “*IsEnabled*” registry key setting under the **HKEY_LOCAL_MACHINE\Comm\FTPD** registry key. Figure 3.19 and figure 3.20 show how to enable the ftp service.

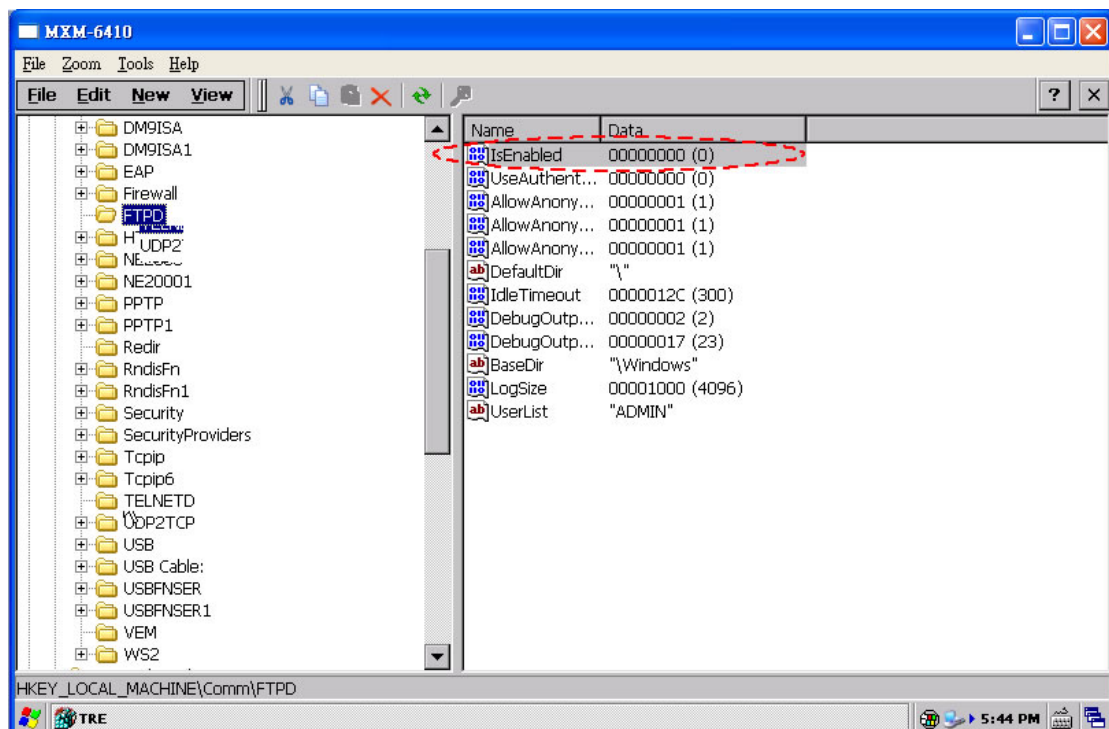
Go to **Start** → **Programs** → **regedit**

Figure 3.19 *regedit Utility*



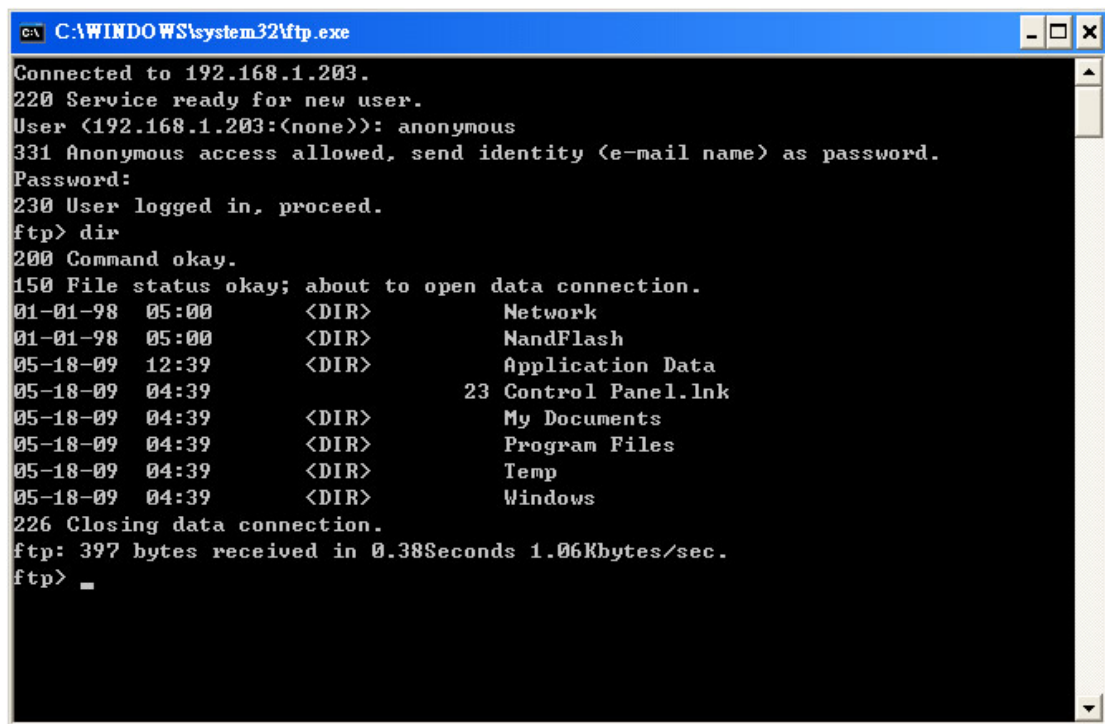
The ftpd registry settings are under the **HKEY_LOCAL_MACHINE\Comm\FTPD** registry key. Double click the “*IsEnabled*” registry key set it from “0” to “1”.

Figure 3.20 *Enable Ftpd Service*



To access the FTP server on a MXM-6410/APC-6410 device, simply enter *ftp <Your Device IP address>* at your PC command prompt. See the example below.

Figure 3.21 FTP Prompt



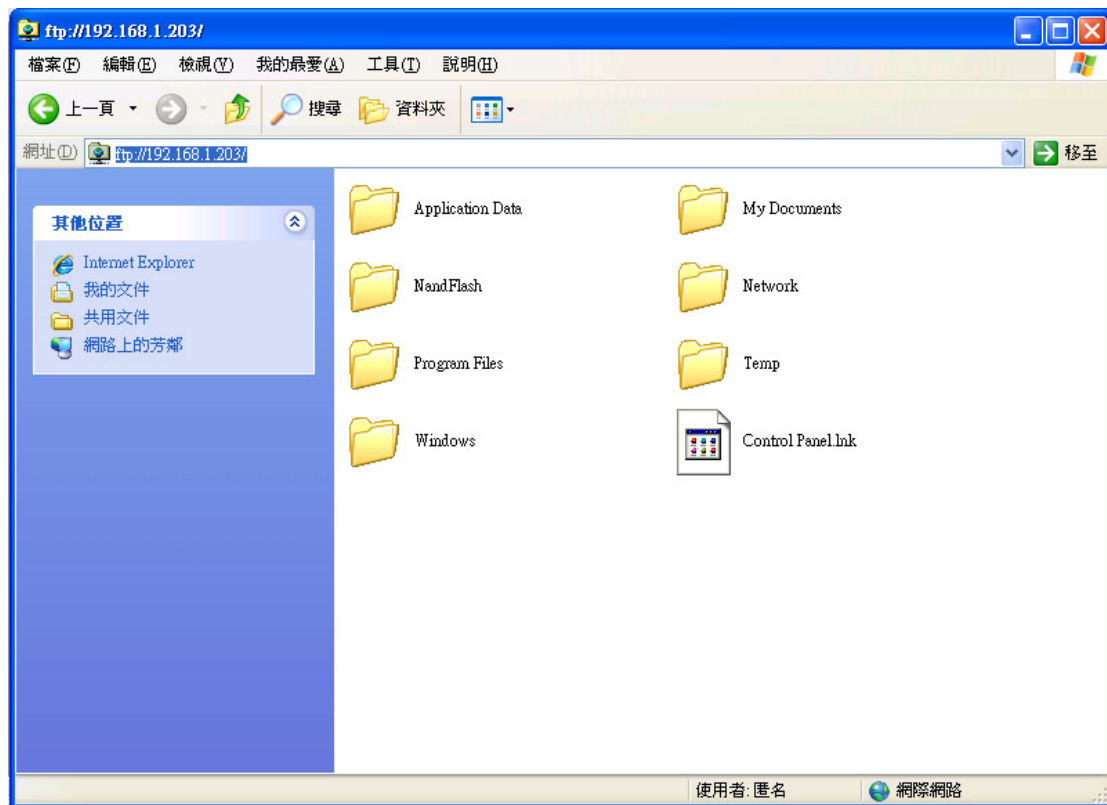
```
G:\ C:\WINDOWS\system32\ftp.exe
Connected to 192.168.1.203.
220 Service ready for new user.
User (192.168.1.203:(none)): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in, proceed.
ftp> dir
200 Command okay.
150 File status okay; about to open data connection.
01-01-98 05:00 <DIR> Network
01-01-98 05:00 <DIR> NandFlash
05-18-09 12:39 <DIR> Application Data
05-18-09 04:39 23 Control Panel.lnk
05-18-09 04:39 <DIR> My Documents
05-18-09 04:39 <DIR> Program Files
05-18-09 04:39 <DIR> Temp
05-18-09 04:39 <DIR> Windows
226 Closing data connection.
ftp: 397 bytes received in 0.38Seconds 1.06Kbytes/sec.
ftp> _
```

Once this *ftp <Your Device IP address>* command has been entered, the FTP server will respond by asking for a username and password. It is currently configured to allow any user in with any password, including **anonymous**.

Use File Browser to access the FTP Server

A regular file browser in your Windows PC may be used to access the FTP server as well. To use the FTP server this way, use *ftp://the device IP address*.

Figure 3.22 Use File Browser to access FTP Server



MXM-6410/APC-6410 implementation of FTP server enables you to transfer files from a desktop computer using a TCP/IP connection. The implementation of FTP server in MXM-6410/APC-6410 is based on RFC 959. This included FTP server supports the minimum implementation of the FTP protocol defined in RFC 959. This minimum implementation includes configuration values, transfer parameters, and ASCII and Image data types, and allows FTP to operate with a minimum of error messages.

3.2.6.1. Default Registry Setting

It's necessary to be aware of registry settings that impact security. The telnet registry settings are under the

HKEY_LOCAL_MACHINE\Comm\FTPD registry key.

FTP server reads the values in the registry before serving each request. Therefore, changes made to the registry take affect immediately and do not require the ftp server to be restart. The registry change will be kept on next reboot.

Table 3.2. FTPD Registry Value

HKEY_LOCAL_MACHINE\Comm\FTPD	
Value	Description
AllowAnonymous : REG_DWORD	Default set to 1. Possible values are 0 (false) or 1 (true). Determines whether the server will allow anonymous access.
AllowAnonymousUpload : REG_DWORD	Default set to zero (0). Possible values are 0 (false) or 1 (true). Determines whether authorization is required to upload files to the server, delete files from the server, and rename files.
AllowAnonymousVroots : REG_DWORD	Default set to zero (0). Possible values are 0 (false) or 1 (true). Specifies whether access to virtual roots is granted or denied to anonymous users.
DefaultDir : String	Default root directory. Directory and subdirectories of this key are accessible remotely. If this value is not set in the registry, the default is \Temp.
IsEnabled : REG_DWORD	Default set to 1. Possible values are 0 (false) or 1 (true). Determines whether or not the server will accept incoming connections. This value is typically used to keep the server disabled at boot time.
UserList : REG_MULTI_SZ	Provides a comma-separated list of allowed users.
NoSystemInfo : REG_DWORD	No default set in registry. Uses value of 0 if no value is set. If NoSystemInfo is set to 1, and a remote FTP client requests the Operating System name and version from the FTP server (via the "SYST" command), the FTP server will not indicate it.

3.2.6.2. Changing the FTP Server Access Privileges

To change the FTP server access privileges, the Registry Editor tool under the **Windows** folder (or **Start → Programs → regedit**) will be used.

To modify the FTP server access privileges:

Step 1 Click the regedit.exe located at **Windows** folder.

Step 2 Browse to the **HKEY_LOCAL_MACHINE\Comm\FTPD**.

Step 3 Ensure that the *IsEnabled* value is **1**.

Step 4 Ensure that the *UseAuthentication* value is **0**.

Step 5 Set the **UserList** value to a semi-colon separated list of users. (For example, **user1;user2**).

Refer to the topic **Telnet Server User Lists** in the Platform Builder documentation for more detailed information on this subject.

Examples:

```
[HKEY_LOCAL_MACHINE\COMM\TELNETD]
```

```
UserList="GladysL;AnnD;-KimY;@SomeGroup;-@villains"
```

Allows GladysL, AnnD, and the SomeGroup group, but restricts the KimY user and the villains group.

The asterisk or the at sign and the asterisk (* or @*) allows all users, and the combination of the hyphen and the asterisk, or the combination of the hyphen and the at sign and the asterisk (-* or -@*) denies all users.

The string is interpreted sequentially. In other words, if GladysL is member of the Finance group, the following strings will allow GladysL.

```
GladysL
```

```
@ Finance
```

```
@ Finance;GladysL
```

```
GladysL;@ Finance
```

```
@ Admins;-GladysL
```

```
GladysL;-@ Finance
```

```
GladysL;-GladysL
```

```
*;-GladysL;-@ Finance
```

Step 6 Set the **AllowAnonymous** value to 1 if you wish to enable anonymous FTP logins.

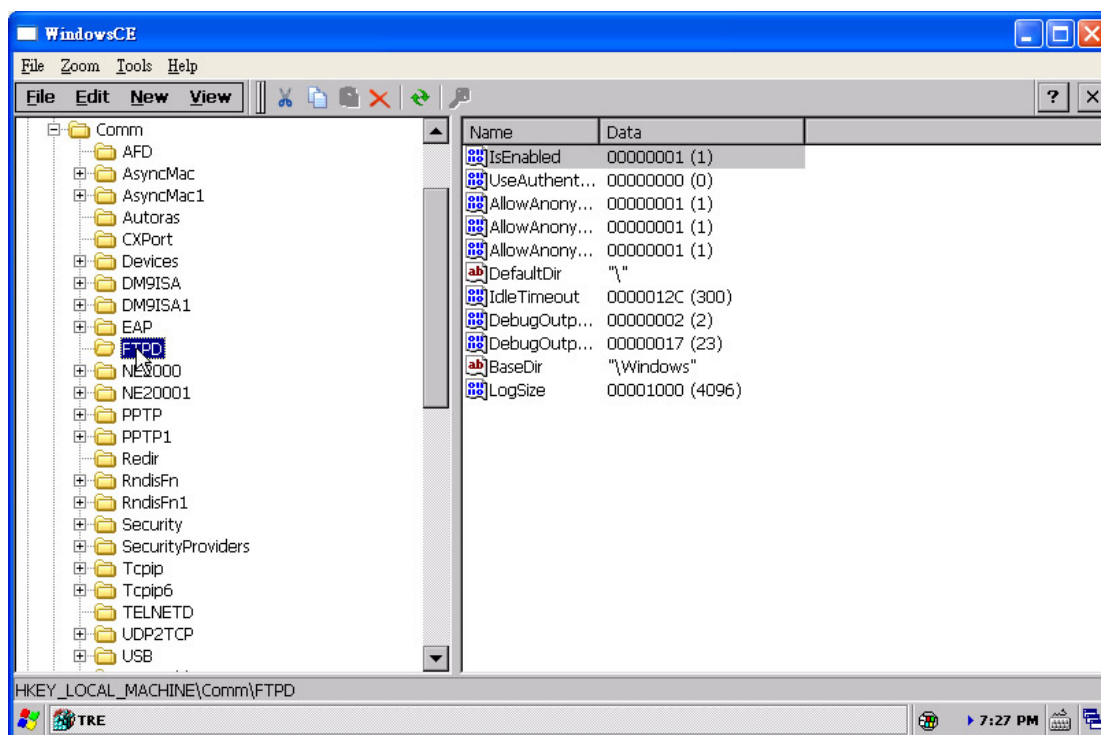
Step 7 Set the **AllowAnonymousUpload** value to 1 if you wish to

enable anonymous users from uploading files to the FTP server.

Step 8 Set the **DefaultDir** string to whichever directory you wish to use as the FTP root directory. Users will not be able to go outside of this directory.

Further information may be found in the Platform Builder documentation on the FTP server, located in **Application Development > FTP Server**.

Figure 3.23 FTP Registry



3.2.6.3. Security Notes

If **AllowAnonymous** is set to true, it will allow users to connect the server without providing verifiable credentials. Anyone can log in using the username "anonymous" and any password to gain access. It is recommended that you set this value to false and use the **UserList** registry setting to specify all allowed users.

If **AllowAnonymousUpload** is set to true, unauthenticated users will be able to copy files to, and delete files from, your server. This can be very dangerous because attackers might upload dangerous applications and documents, or they might delete important system files. It is not recommended to allow upload permission for anonymous users.

If **AllowAnonymousVroots** is set to false, anonymous users will only be able to access the main FTP shares. If this value is set to true, unauthenticated users will also be able to access VROOTs as well as the main share. Therefore you should use this setting with caution.

Setting **UseAuthentication** to false enables clients to connect to the server without providing credentials. It is therefore strongly recommended that you do not set this value to false. Change this setting only if you have anonymous clients that must access the server but cannot or will not send USER and PASS credentials.

It is recommended that you set this value to a list of users who should have access to the server and its member VROOTs. Specifying the allowed users in **UserList** and setting **AllowAnonymous** to false will help protect the device from most attacks and keep your files available only to those users who need to see them.

3.2.6.4. Recommendations

Set the user list and domain variables to prevent hacker attacks on your device. If FTP server functionality is used without appropriate values set for the User List and Domain variables, the FTP server will be vulnerable to hacker attacks.

These variables are not set by default. A hacker must only guess the device's password, the way it is set in Control Panel, to obtain access to server.

To prevent such attack, the user name is the **UserList** registry value must be set for each of the servers that are currently running. The user will then need to log in with the specified user name and appropriate password to use the server. You can also set the domain variable in the **DefaultDomain** registry value, which is located under **HKEY_LOCAL_MACHINE\Comm\Redir** registry key. Setting the **DefaultDomain** registry value will require FTP clients to have valid domain credentials to log in.

3.2.7. File Server

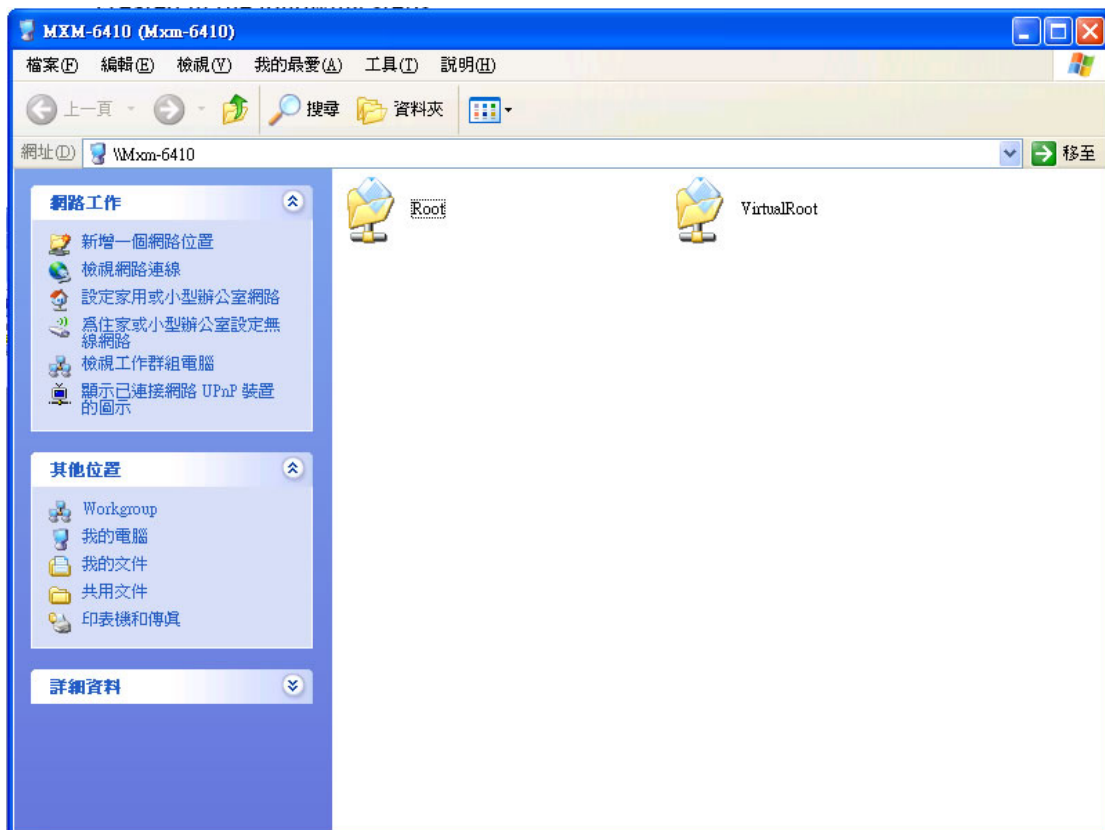
File server in the factory-installed image is configured with no access restrictions. You should disable the file server or restrict access to it before deploying the MXM-6410/APC-6410 or connecting to an unsecured network. The File Server enables clients to access files and other resources, such as printer, from a server over using TCP/IP. File Server uses the Common Internet File System (CIFS). This is an extension to the Server Message Block (SMB) file sharing protocol. CIFS enables a network-enabled application to access and manipulate files and directories on a remote server in the same way that the application accesses and manipulates files and directories on the local system.

To use the file server, users can access and browse the shared folders that created in the default registry in the following steps.

1. On the PC, from the **Start** menu, choose **Run**.
2. In the Open box, type **\\MXM-6410**, and then choose **OK**.

You can now browse the shared files in devices as showed in figure 3.24.

Figure 3.24 Access File Server

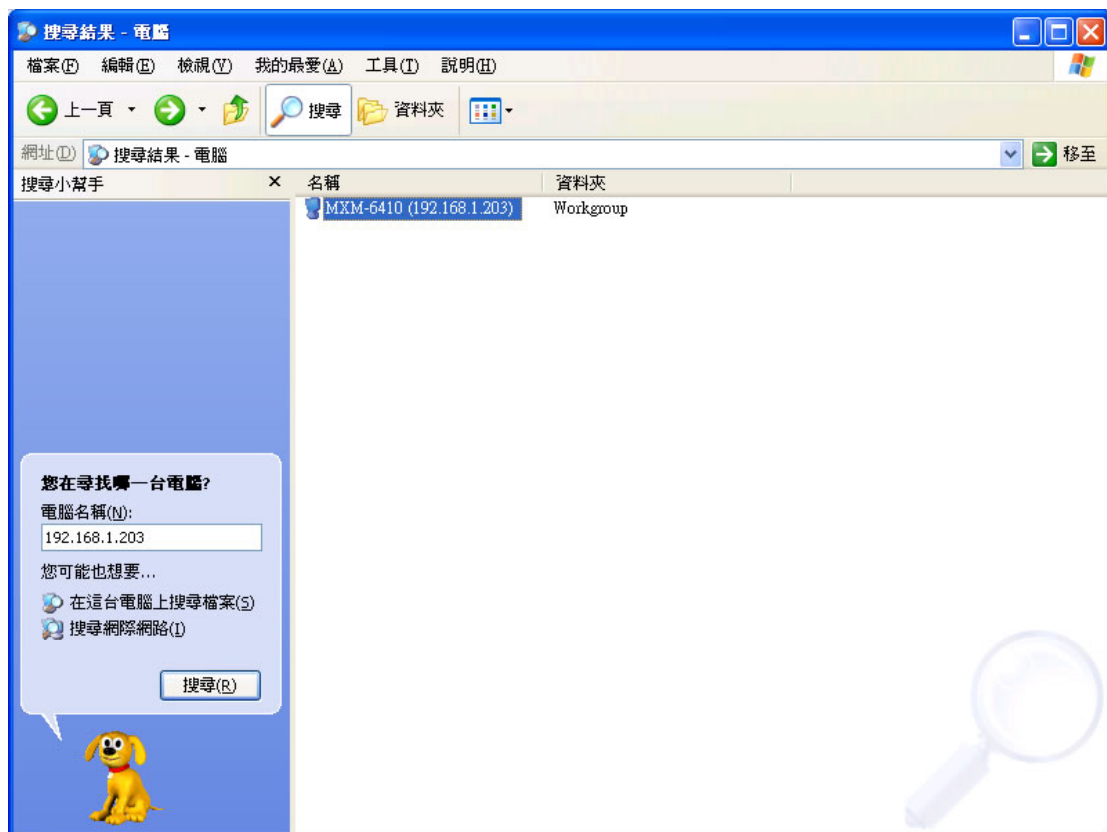


The alternative way to access to shared files in the file server is described as follows.

1. On the PC, from the **Start** menu, choose **Search**.
2. In the searching menu, choose **Computer or People**, and then choose **Computer on the network**.
3. In the Open box, type **MXM-6410** or **<Device IP Address>**, and then click **Search**.

You can now browse the shared files in devices as showed in figure 3.25

Figure 3.25 The other way to access file server from PC



3.2.7.1. Default Registry Settings

The registry stores information necessary to configure the system for applications and hardware devices. The registry also contains information that the operating system continually references during operation. MXM-6410/APC-6410 enables you to create virtual file server directories. To users who access your file server share, virtual directories appear as subdirectories of the file server share, although these directories may be located in a different folder. You can create a virtual root directory called "VirtualRoot" that points to \Windows directory of the device and a root directory called "Root" that points to \NandFlash directory of the device by specifying the following registry key:

HKEY_LOCAL_MACHINE\Services\SMBServer\Shares\Root and
HKEY_LOCAL_MACHINE\Services\SMBServer\Shares\VirtualRoot

Note the maximum length of the virtual root directory and root directory is 12 characters.

Table 3.3. File Server Registry Value

HKEY_LOCAL_MACHINE\Services\SMBServer\Shares	
Value	Description
UseAuthentication: REG_DWORD	No default set. Setting this value to 0 will disable the authentication on the file server. The file server will be accessible to all users on the network.

Table 3.4. VirtualRoot/Root of File Server Registry Value

HKEY_LOCAL_MACHINE\Services\SMBServer\Shares\VirtualRoot HKEY_LOCAL_MACHINE\Services\SMBServer\Shares\Root	
Value	Description
Path : String	Specifies the path to be shared.
Type: REG_DWORD	Setting this value to 1 designates this as a print server share, setting this value to 0 (zero) designates this as a file server share.
UserList : String	Provides a comma-separated list of allowed users. Requires UseAuthentication to be enabled.

3.2.7.2. Security Note

The default setting of authentication on the file server is disabled. However, it is not recommended that you disable authentication on the file server and you share the \Windows or root directory.

You can specify a list of folders that cannot be shared. You can use any name for each folder you specify in the exclusion list. Setting the **HKEY_LOCAL_MACHINE\Services\SMBServer\Shares\ExcludePaths** registry key prevents the configuration functions from creating the specified shares, so that they cannot be accessed by an un-trusted application.

Table 3.5 An example to exclude the folders to be shared

HKEY_LOCAL_MACHINE\Services\SMBServer\Shares\ExcludePaths	
Value Name	Value
"Windows"	"\\Windows"
"My Documents"	"Documents and Settings"

3.2.8. Web Server

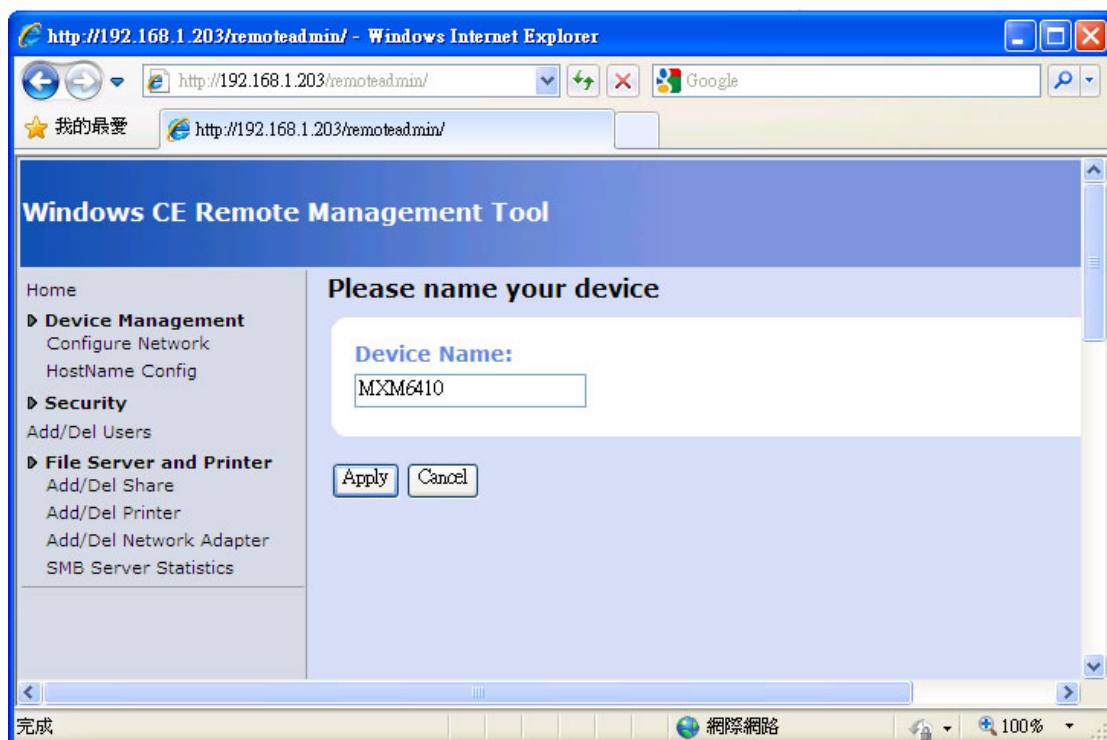
Web server facilitates the use of the Internet for communication between MXM-6410/APC-6410 device and network printers, scanners, and other shared equipment. The Web server applications send Hypertext Markup Language (HTML) pages to a requesting browser. Users only need to have an Internet connection and a browser to be able to make use of the Web server functionality. The Web server supports IPv6 and also supports the use of Active Server Pages (ASP).

The HTTP server is not started on MXM-6410/APC-6410 hardware at boot by default. To turn on the telnet service, users need to enable the "*IsEnabled*" registry key setting under the

HKEY_LOCAL_MACHINE\Comm\HTTPD registry key.

To access the Web server, just type <IP Address> at the Web browser. The default Web service configuration setting is remote administration page. It is very useful to use when the device is without display. At first connection to the Web server, users will be asked to input the password and re-type to confirm. After that, figure 3.24 shows the remote administration pages.

Figure 3.24 Remote Administration Page



Note: If users would like to use a own custom webpage, users need to modify the “Default” registry key setting under the **HKEY_LOCAL_MACHINE\Comm\HTTPD\ROOTSV** registry key from “\$REDIRECT” to “\windows\www\wwwpubl”

3.2.8.1. Default Registry Settings

It is necessary to be aware of the registry settings that impact security. The Web server settings are located under the **HKEY_LOCAL_MACHINE\Comm\HTTPD** registry key. If you make changes to the Web server registry settings, it is necessary to stop the Web server and restart it to make the changes take effect. The “IsEnabled” registry value is default set to disable the Web service and is checked only when the Web server is initially loaded. If the registry value is set to zero (0), the Web server does not start. Changing this value to zero (0) while the Web server is running has no effect. You also must stop the Web server to make it stop accepting connections.

Table 3.6. HTTPD Registry Key and Named Values

HKEY_LOCAL_MACHINE\Comm\HTTPD	
Value	Description
BasicRealm : String	Specifies the string that the Web Server will send to clients as its Basic realm when performing basic authentication. If this registry value is not set, the Web Server will default to using the string "Microsoft-WinCE".
IsEnabled : REG_DWORD	If the value is not set in the registry, the Web Server is enabled. If the value is set to zero (0), the Web Server does not accept connections from the network, even from the local host.
Port : REG_DWORD	Default setting is 80. This port receives HTTP connections. Do not set the port to zero (0). When the Web Server is running from Services.exe, which is the default, this value is ignored and Services.exe becomes the super server.
Basic : REG_DWORD	Default setting is zero (0). If this value is nonzero, the Web Server uses Basic authentication for client connections. Enables Basic authentication, which enables the client browser to send the user identifier and password to the server.
NTLM : REG_DWORD	Default setting is 1. If this value is set to nonzero, the Web Server uses NTLM authentication for client browser connections. Also, if this value is nonzero, the failure of Basic authentication forces NTLM authentication. If the value is not set in the registry, NTLM is not used.
DirBrowse : REG_DWORD	Default setting is zero (0). If this value is set to nonzero, directory browsing is allowed. If this value is not set in the registry, directory browsing is turned off. Turns on the Web Server's ability to provide local directory browsing.
Filter DLLs : String	Default not set in the registry. Specifies a list of DLL names, separated by commas, that identifies the filters to use.

DefaultPage : String	<p>Default not set in the registry. If the value is not present in the registry, the Web Server will use "default.htm;index.htm".</p> <p>Specifies a list of page names, separated by semicolons, that indicates names interpreted by the Web Server to be default pages. When browsing a directory, the Web Server traverses this list searching for a file of the same name in the directory. If the file exists, it is sent to the client. If no matching file exists, the Web Server sends a directory listing or returns an error, depending on whether directory browsing is enabled. If more than one DefaultPage file name is matched, the Web Server uses the first matching file name.</p>
AdminUsers : String	<p>Default not set in the registry.</p> <p>Specifies a list of user names, separated by semicolons. A user who has gained user access must be listed in this key to gain Administrator access.</p>
LogFileDirectory : String	<p>Default setting is "\windows\www" directory. If the name is not set or if the specified directory is inaccessible, no logging is performed.</p> <p>Specifies the name of the directory where the logging files are created.</p>
PostReadSize : REG_DWORD	<p>If the value is not set in the registry, PostReadSize will default to 48 KB. The Web Server uses a minimum value of 8192 bytes (8 KB). If the value in the registry is less than 8 KB, the value is ignored and the Web Server will use 8 KB.</p> <p>Specifies the maximum number of bytes that the Web Server reads when receiving POST data. To read more data, you must use a raw data filter or call ReadClient in an ISAPI extension.</p>
MaxLogSize : REG_DWORD	<p>Default setting is 32 KB. If this value is not set in the registry, or if it is set to zero (0), no logging is performed.</p> <p>Specifies the maximum size, in bytes, that a log file can become before it is rolled over.</p>

MaxHeaderSize : REG_DWORD	<p>Default setting is 48 KB in the registry.</p> <p>Specifies the maximum number of bytes that the Web Server will read of HTTP headers. If the header size exceeds this value, the Web Server will terminate the session and return a message to the client: 400 - Bad Request.</p>
MaxConnections : REG_DWORD	<p>Default is not set in the registry. If the value is not set in the registry, MaxConnections will default to 10.</p> <p>Specifies the maximum number of simultaneous connections to the Web site. After the maximum number of connections is established, additional client requests will be sent a message: 503 – Server Too Busy.</p>
ServerID : String	<p>Default is not set in the registry. If the value is not set in the registry, ServerID will default to "Microsoft-WinCE/X.Y", where X is the major version and Y is the minor version of Windows CE-based device.</p> <p>If ServerID is set, the Web Server returns the specified server name in the response header.</p> <p>Identifies the server name that is included when the Web Server generates HTTP response headers. The response header includes a field name "Server: ". Optionally, you can include the software version number or any similar information in the string.</p>

3.2.8.2. Security Note

When using **Basic** authentication, the client browser sends the user identifier and password to the server in clear text. In addition, all data sent between the client and browser is in clear text and therefore vulnerable to packet sniffing. You should consider using SSL to help protect sensitive information.

NTLM Although the client browser sends the password to the server in encrypted format, all data sent between the client and the browser is in clear text and therefore vulnerable to packet sniffing. You should using SSL to help protect sensitive information.

DirBrowse turns on Web server's ability to provide local directory browsing. This exposes the local file system to remote browser through HTTP. Users can view file lists and download files depending on virtual root and authentication registry settings. Enabling directory browsing increases the potential attack surfaces, therefore you should enable directory browsing only when necessary.

AdminUsers User names in this list identify the administrators of the site who have access to all virtual roots hosted on this Web site, including the restricted sites. Choose these users carefully and ensure that they set proper password, otherwise their accounts could be used to gain access to restricted sites.

MaxConnections Setting the value too small can block user access to the site. However, if the value is too large, the Web server will consume more system resources. Based on your deployment model, choose this number appropriately.

ServerID To avoid revealing the server software information to malicious users, you may want to create a custom server name that obfuscates the Web Server and operating system versions.

3.2.8.3. Recommendations

A typical deployment uses a Web server in a private network to provide a remote user interface to configure a headless device. The registry defines the number of connections and when the **MaxConnections** registry value is not set, the registry limits the number to 10.

A typical deployment uses the Web server to display status information or to host a family or community Web site. You should not use the Web server to perform critical operations, such as machine control or financial processing.

Use NTLM and/or Basic authentication mechanism to limit access to known users only. You can set the option in the

HKEY_LOCAL_MACHINE\COMM\HTTPD registry key.

SSL protocol helps to protect data from packet sniffing by anyone with physical access to the network.

Carefully choose your virtual roots and limit access to the appropriate files by providing appropriate user access lists. Anonymous users with access to the virtual root may be able to access files and directories within that virtual root.

You can set the options in

HKEY_LOCAL_MACHINE\COMM\HTTPD\ROOTS registry key.

3.2.9. Auto RUN

When Windows CE begins loading, the kernel starts the file system and examines the **HKEY_LOCAL_MACHINE\Init** registry key to identify what applications to run. To control which applications run at system startup, create launch registry values. Launch registry values do not need to be stored in the registry, although you can specify dependencies. You can specify up to 32 applications.

Table 3.7. Named values of HKEY_LOCAL_MACHINE\Init Key

HKEY_LOCAL_MACHINE\Init	
Value	Description
Launchnn : String	Specifies the application to launch in order “ nn ”
Dependnn : Binary	Launch nn registry values have optional dependencies as denoted by the Depend nn registry value.

Depend nn registry values specify applications that Windows CE must be running before the **Launch nn** applications run.

Depend nn registry values begin with the keyword **Depend**, followed by the same decimal number as the **Launch nn** registry value.

The **Depend nn** registry values define an order in which Windows CE launches applications. One or more dependent applications can be specified per **Depend nn** value. Dependent applications are specified as a series of Words in hexadecimal notation.

Table 3.8. Typical Init Registry Entry Using Dependencies

HKEY_LOCAL_MACHINE\Init	
Value Name	Value
Launch20	device.dll
Launch30	gwes.dll
Depend30	2 length binary value
Launch50	explore.exe
Depend50	4 length binary value
Launch60	serviceStart.exe
Depend60	2 length binary value

If your applications have dependencies with other applications, use the registry editor to setup your applications manually. After finishing the registry setup, the system will automatically flush the registry on changes to HIVE.

3.2.10. COM Ports

3.2.10.1. Introduction

The MXM-6410/APC-6410 names the serial ports as “COMx”. Note the **COM_IOCTL_SET_OPERATION_MODE** call will be returned with error if trying to configure a non-configurable port.

Table 3.9. Related Win32 APIs to Configure COMx

Device Name: COMx:	
Win32 Functions	Description
CreateFile	Opens COMx devices.
CloseHandle	Closes COMx device.
DeviceIoControl	Calls a customized IOCTL function.

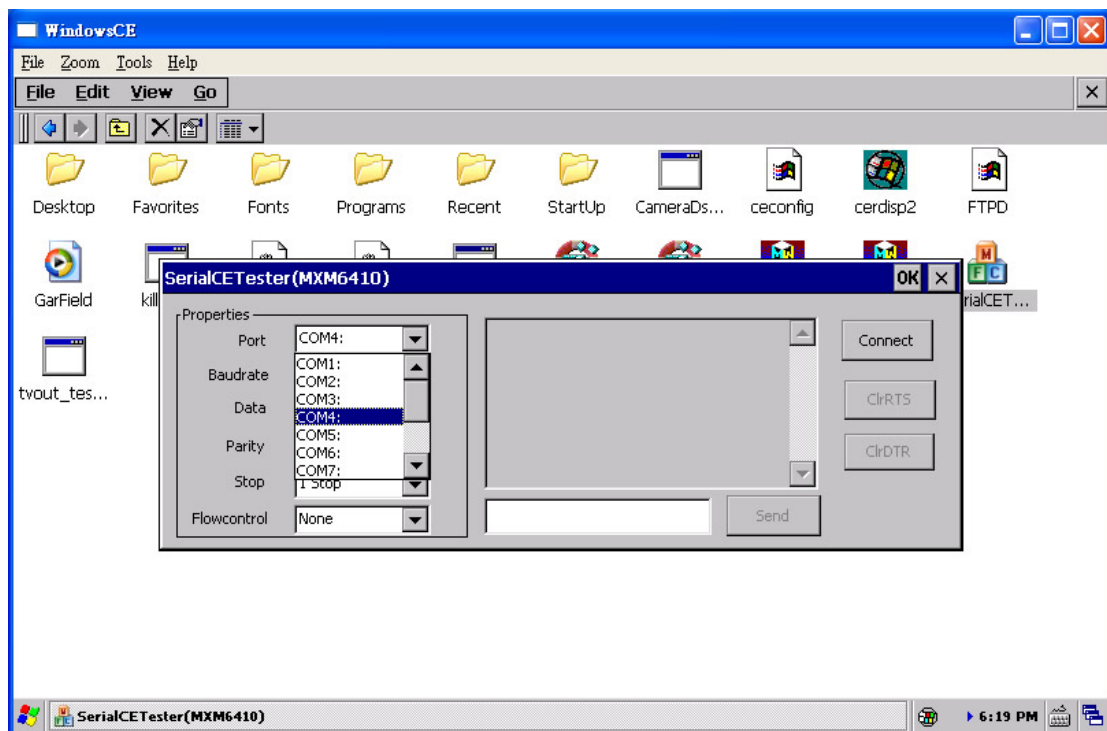
Table 3.10. IO Control Codes for COMx

Device Name: COMx:	
IO Control Codes	Description
COM_IOCTL_GET_OPERATION_MODE	This IOCTL is used to get the operation mode.
COM_IOCTL_SET_OPREATION_MODE	This IOCTL is used to set the operating mode.

3.2.10.2. Test COM Ports

To test COM ports, user ActiveSync program to put the SerialCETester.exe that Embedian offered into device \Windows directory and open it. You will see as shown figure 3.25.

Figure 3.25 COM Port Test Program



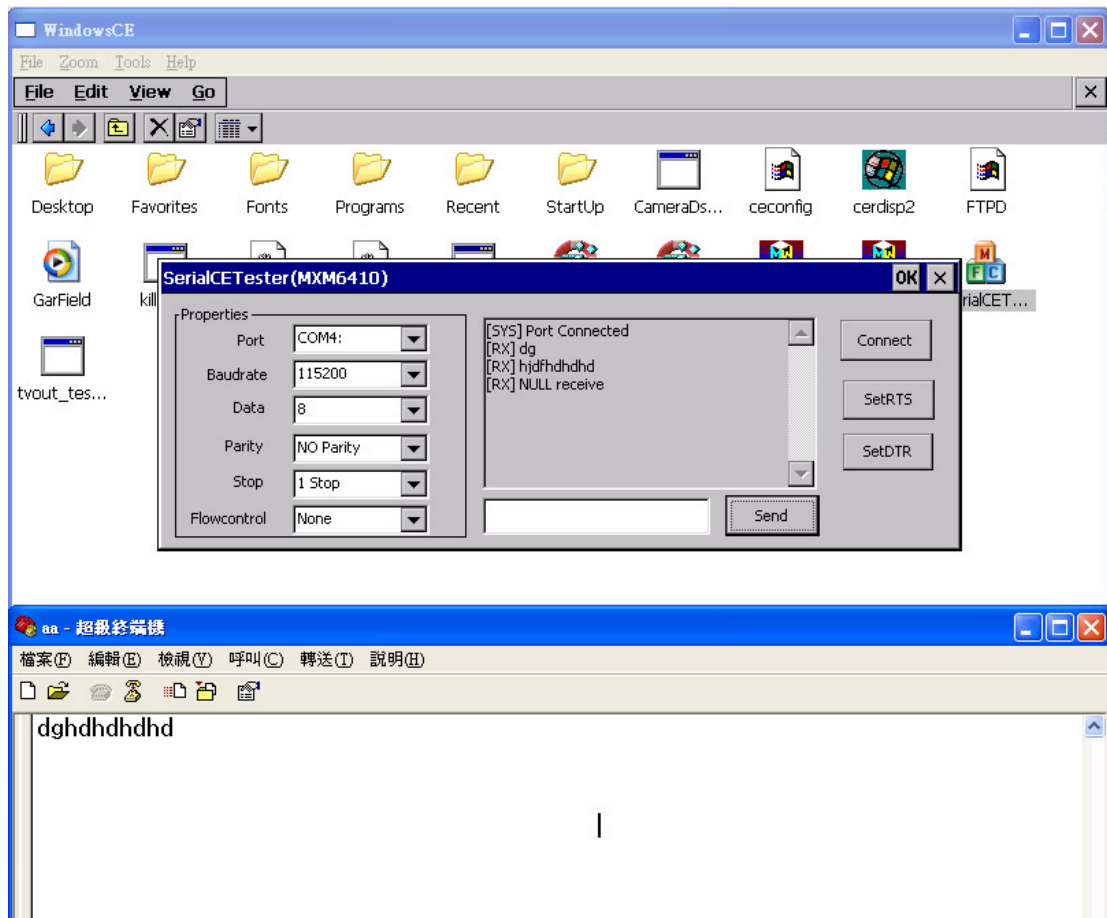
At the PC side, open the hyper terminal program and set the COM port baud rate, configuration and hardware flow control.

At device side, choose which COM port you would like to test and set the COM port baud rate, configuration and hardware flow control exactly the same as that you set in PC Hyperterminal.

Press "Connect" and type and characters from PC. You should be able to receive those characters from the CE device.

Type any characters in the dialog box of the CE test program and then click "Send". You should be able to receive them in the PC Hyperterminal program. This is shown in figure 3.26.

Figure 3.26 COM Port Test Result



3.2.11. Software Installation

Applications installed on to the Embedian Windows CE products will not remain on the device after rebooting the system. This is due to the applications being installed to the \Windows and \Program Files folders. The \Windows and \Program Files folders are not persistent. For this reason, users can integrate the software from the platform builder and integrate the software into the NK or users can use the batch file to implement that. The following steps must be taken to restore the files to the \Windows and \Program Files folders.

3.2.11.1. Software Installation using a Batch File

Software Installation

1. Follow the standard installation of the software application to the Windows CE device.
2. Test the software application.
3. From the Desktop run the application and Save the Registry.
4. Connect the Windows CE device to a PC with ActiveSync.
5. Create a folder on your hard drive of the PC.
6. Copy the files from the Windows CE device to the folder on the hard drive.
 - Directories to check:
 - \Windows
 - \Program Files
 - \Windows\Programs
7. Create a batch file to copy the files to the Windows CE device.
 - Be sure to copy the files to where they were placed during the installation.
 - Folders will need to be created on the Windows CE device to match the ones created during the install.
8. Copy all of the files and the batch file on to a SD/SDHC card.
9. Place the SD/SDHC card in to the Windows CE device.
 - To debug the batch file place a pause as the last line in the file. Then when the batch file runs you will see the results on the Windows CE device.
10. On the Windows CE device run the batch file.
11. Test your software application.
12. Create a Startup folder in the **\NandFlash** folder.
13. Paste a Shortcut to the batch file on the **\StorageCard**.
14. From the Desktop run the application and Save the Registry.
15. Reboot the system.
 - The shortcut should run the batch file from the SD/SDHC card.
16. Test your software application.

SAMPLE.BAT

```
md \Windows\Programs
Copy <filename> \Windows\Programs
md \Program Files\<appname>
Copy <filename> \Program Files\<appname>
Pause
<filename> = the file you wish to copy
<appname> = the application name
md = Make Directory command line function (DOS)
```

Note1: Once the batch file is debugged remove the Pause from the last line and add Echo off as the first line. Echo off will not display the batch file running.

Note2: The only way to make the applications permanent on the Windows CE device is to create a new Windows CE Operating Systems Image. User can custom your own Windows CE Images by the BSP that Embedian supplied.

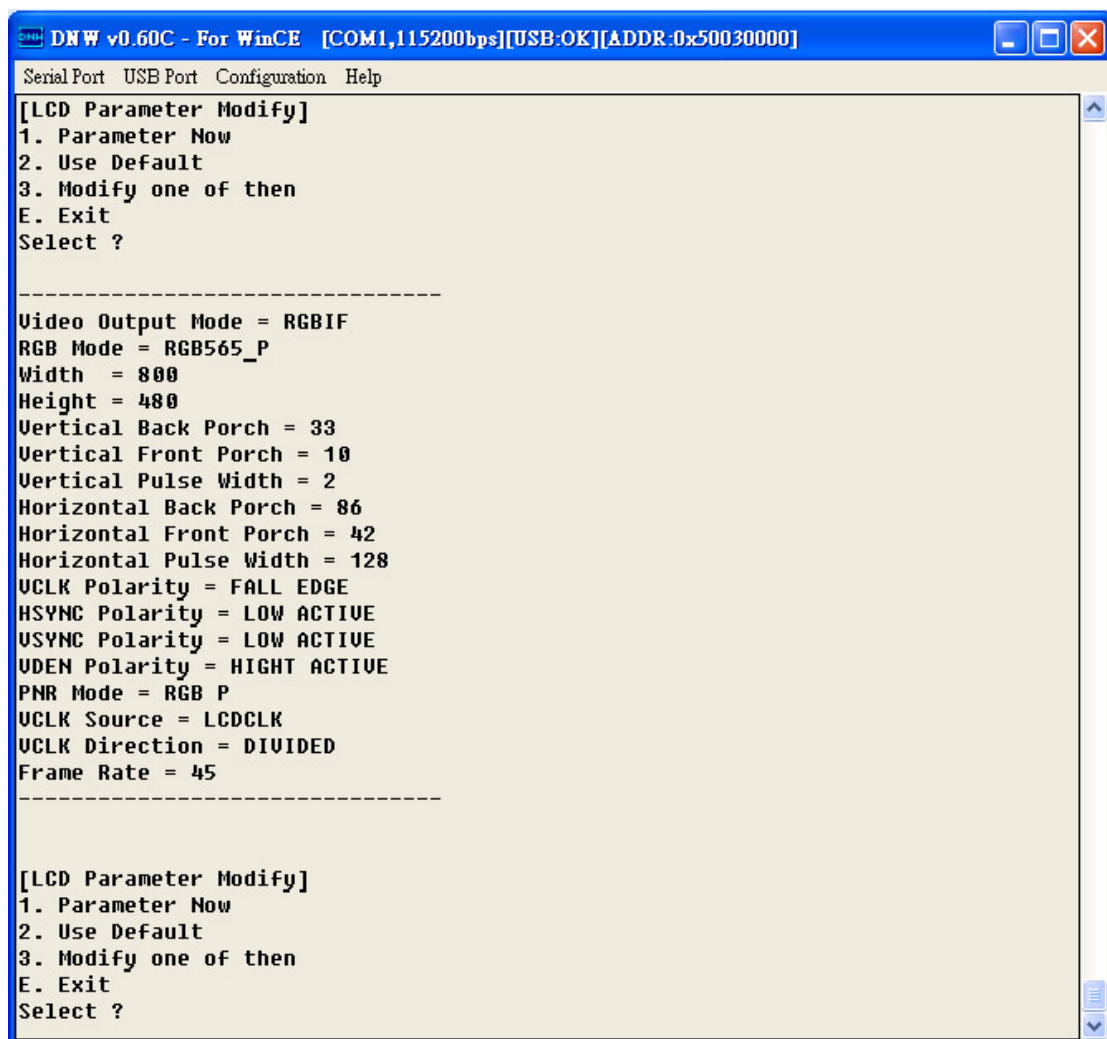
3.3. Configure LCD parameters for different kinds of LCDs

Users don't need to build from BSP to configure LCD parameters of various types. Embedian allow users to configure the LCD parameters at EBOOT menu, after configuring, EBOOT will pass those parameters to NK. Embedian also reserves some very common types LCD parameters as default setting. Users can just choose it instead of setting parameters each by each. Embedian also allow users to view the current LCD parameter settings.

3.3.1. View Current LCD Parameters

To view current LCD parameters, go to EBOOT menu and press "**Z) MXM EVB Board SubTestMenu**" and then press "**2. LCD Panel Parameter Modify**" and then press "**1. Parameter Now**" as shown in figure 3.27

Figure 3.27 View Current LCD Parameters

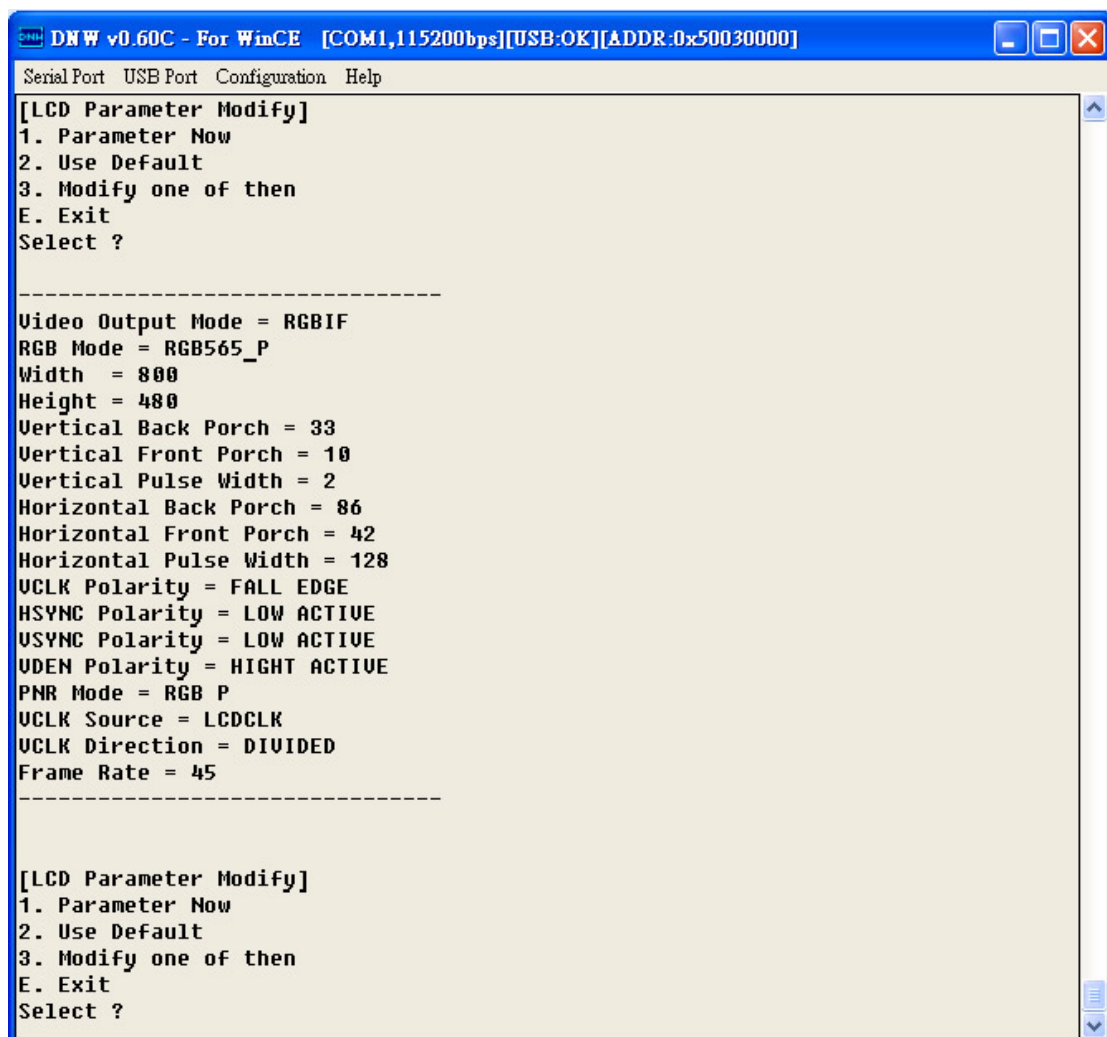


3.3.2. Choose Default LCD Parameters

Embedian chooses Data Image 5.7-inch FG050605DNSWAG01 (320x234) and PVI 7-inch PM070WT3 (800x480) as default LCD parameters because they are also compatible to many LCDs with same resolutions.

To set the default LCD parameters, go to EBOOT menu and press "**Z) MXM EVB Board SubTestMenu**" and then press "**2. LCD Panel Parameter Modify**" and then press "**2. Use Default**". You will see two types of LCD for your choices. Choose the correct type that you will use. Following figure shows the default LCD parameters.

Figure 3.28 Default LCD Parameters



To save the configuration, press “E” exit to main menu and press “**W) Write Configuration Right Now**” to save it.

3.3.3. Set LCD Parameters of Different Types

Embedian provides users with an easy way to configure parameters for various types of LCDs.

To configure your LCD parameters, go to EBOOT menu and press “**Z) MXM EVB Board SubTestMenu**” and then press “**2. LCD Panel Parameter Modify**” and then press “**3. Modify one of then**”. You will see the following menu

[LCD Parameter Modify by each]

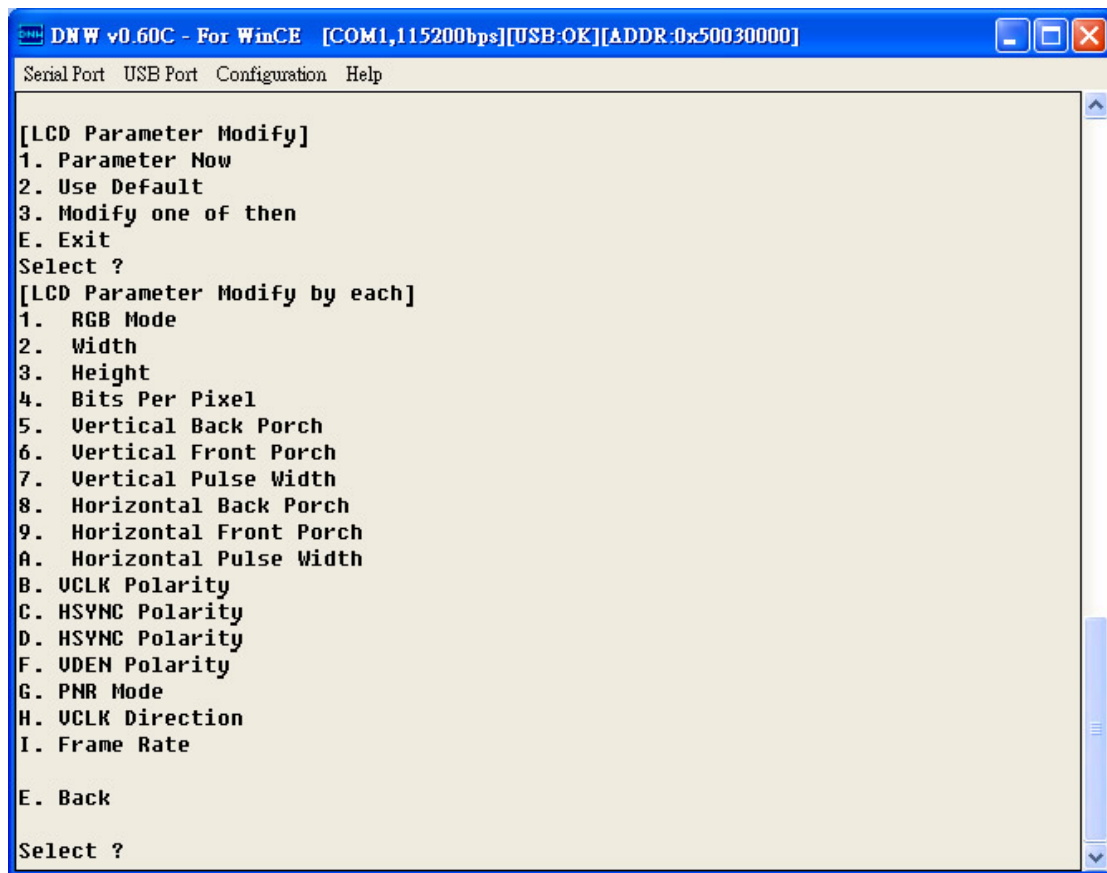
- 1. RGB Mode**
- 2. Width**

- 3. Height
- 4. Bits Per Pixel
- 5. Vertical Back Porch
- 6. Vertical Front Porch
- 7. Vertical Pulse Width
- 8. Horizontal Back Porch
- 9. Horizontal Front Porch
- A. Horizontal Pulse Width
- B. VCLK Polarity
- C. HSYNC Polarity
- D. HSYNC Polarity
- F. VDEN Polarity
- G. PNR Mode
- H. VCLK Direction
- I. Frame Rate

E. Back

You can set each of them by just choose them. Following figure shows the setting of each parameter.

Figure 3.29 Configure LCD Parameters



After done with the setting, press “E” to exit to the main menu and press “W” to save the parameters. The EBOOT will pass the new LCD parameters to NK and you should see the new LCD configuration at next boot.

3.4. Configure GPIOs, BACKLIGHT_EN, LCD_PWREN and VDDLCD_PWREN

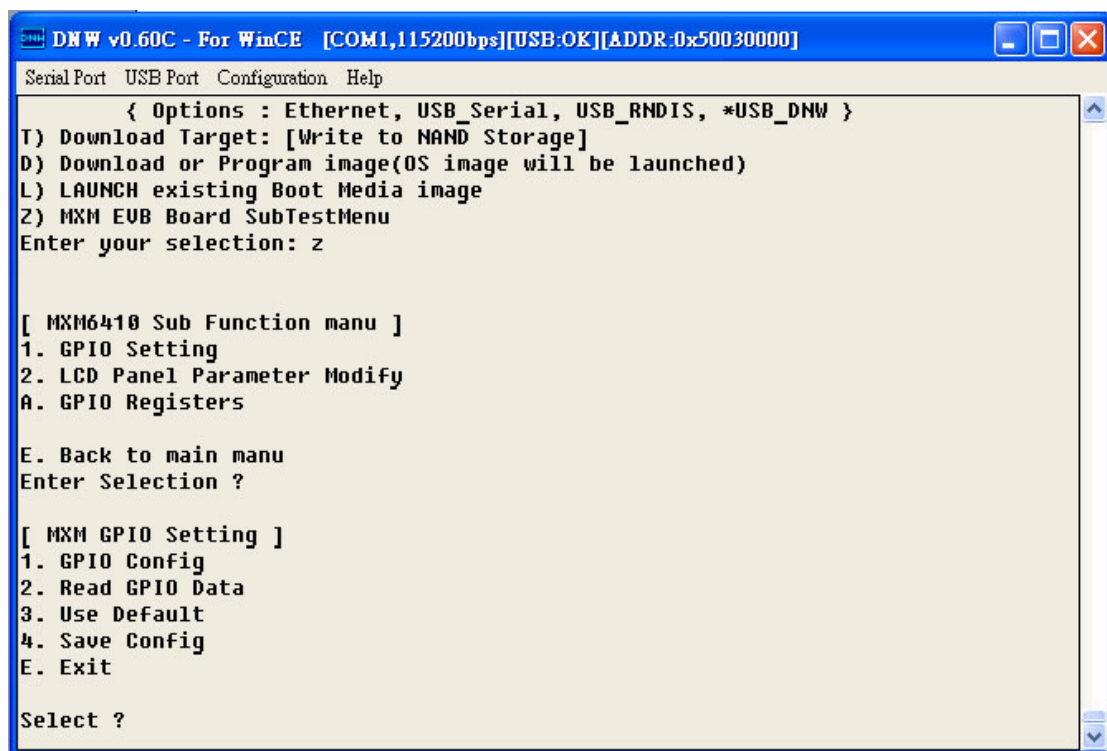
Embedian reserves 12 free available GPIOs as well as BACKLIGHT_EN signal, LCD_PWREN signal and VDDLCD_PWREN for user configuration. Users can set the GPIOs as input or output, high or low, pull-up or pull-down at Eboot or during the NK.

3.4.1. Configure GPIOs Setting at EBOOT

Embedian provides users with an easy way to configure GPIO settings for free available GPIOs at EBOOT.

To configure the GPIO settings, go to EBOOT menu and press “**Z)** MXM EVB Board SubTestMenu” and then press “**1. GPIO Setting**”. Users will see GPIO configuration menu as shown in figure 3.30.

Figure 3.30 GPIO Setting Menus



The “**1. GPIO Config**” option can let users configure each GPIO pin as input or output, pull-up or pull-down and high or low.

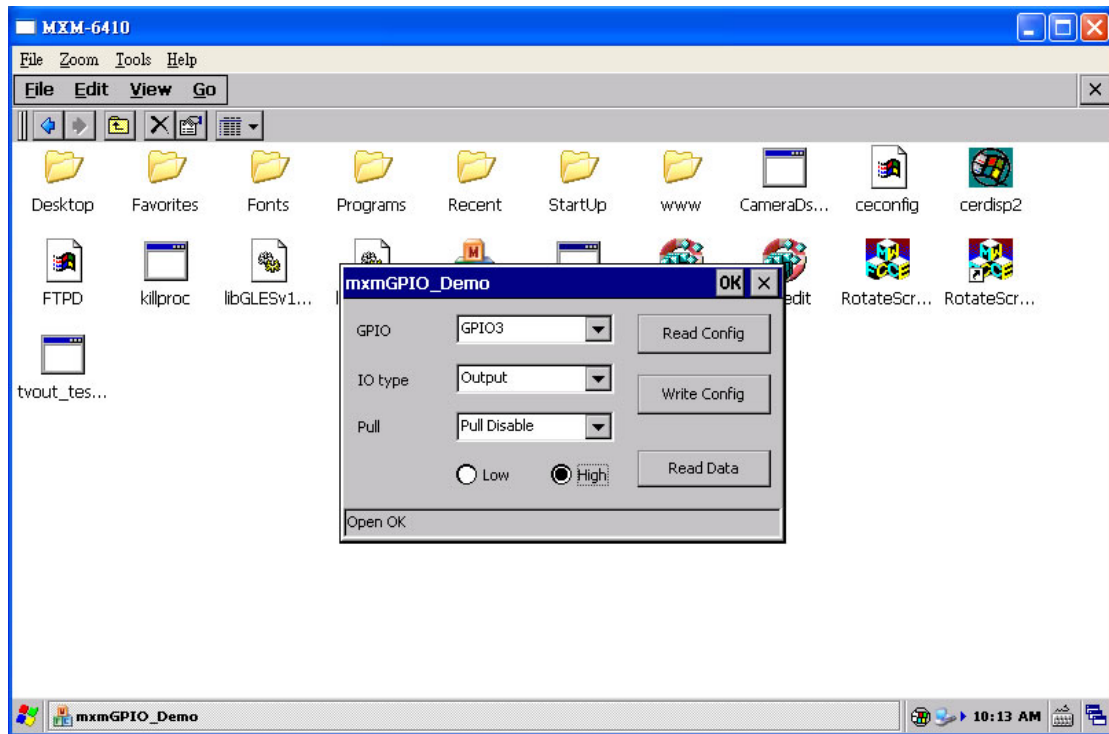
The “**2. Read GPIO Data**” allows users to read the GPIO data if the GPIO is set to be input and connect to the outer world or if the GPIO is set to be output.

The “**3. Use Default**” will set the GPIO 1 to GPIO 12 input pull-down. BACKLIGHT_EN, LCD_PWREN and VDDLCD_PWREN signals are set to output pull-disable high by default.

3.4.2. Configure GPIOs Setting at NK

Embedian provides users with a library and sample demo code to allow user application change the GPIO setting at NK. The control applet named “*mxmGPIO_demo.exe*” provides a simple user interface to diagnostic the functionality of the digital input and digital output channels.

Figure 3.31 GPIO Demo Program



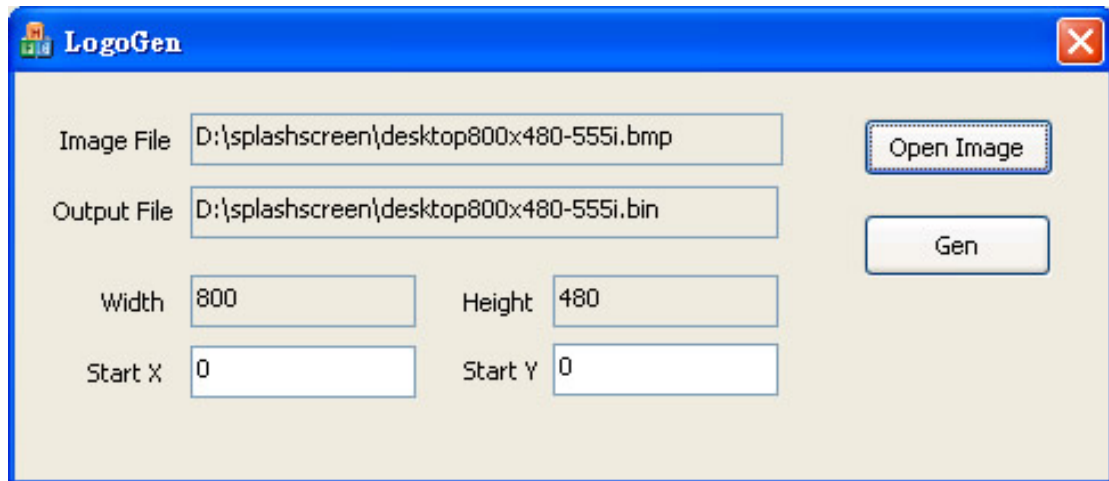
Note:

1. The sample code of the GPIO demo program is available at Embedian FTP site.
2. The “Read Config” button at the right hand side of the program will read back the current GPIO setting.
3. If the IO type is “Output”, the “Write Config” button at the right hand side of the program will set the port to “Low” first by default and it will change to high when user click the “High” radio button.
4. When the IO type is set to “Input”, users can know the setting of outer world by clicking “Read Data” button.

3.5. Logo Splash Screen Customizer

Embedian provides a way to allow users to customize their own boot splash screen at EBOOT. To achieve that, users need to prepare for a bmp 555i file format first. And then use the “LogoGen.exe” program than Embedian provided to convert the 555i file format into an EBOOT readable .bin file format.

Figure 3.32 LogoGen Program

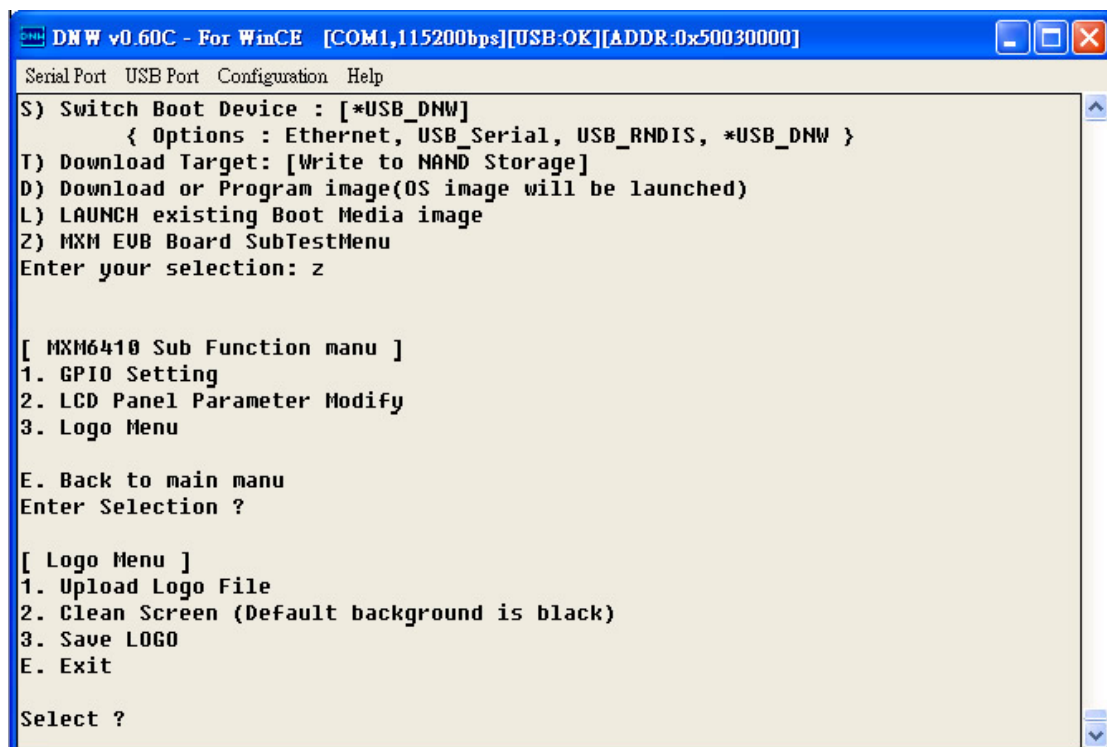


The program allows users set the image width, height, startX and startY parameters and convert to a EBOOT readable .bin file format.

3.5.1. Upload and save splash screen image into devices

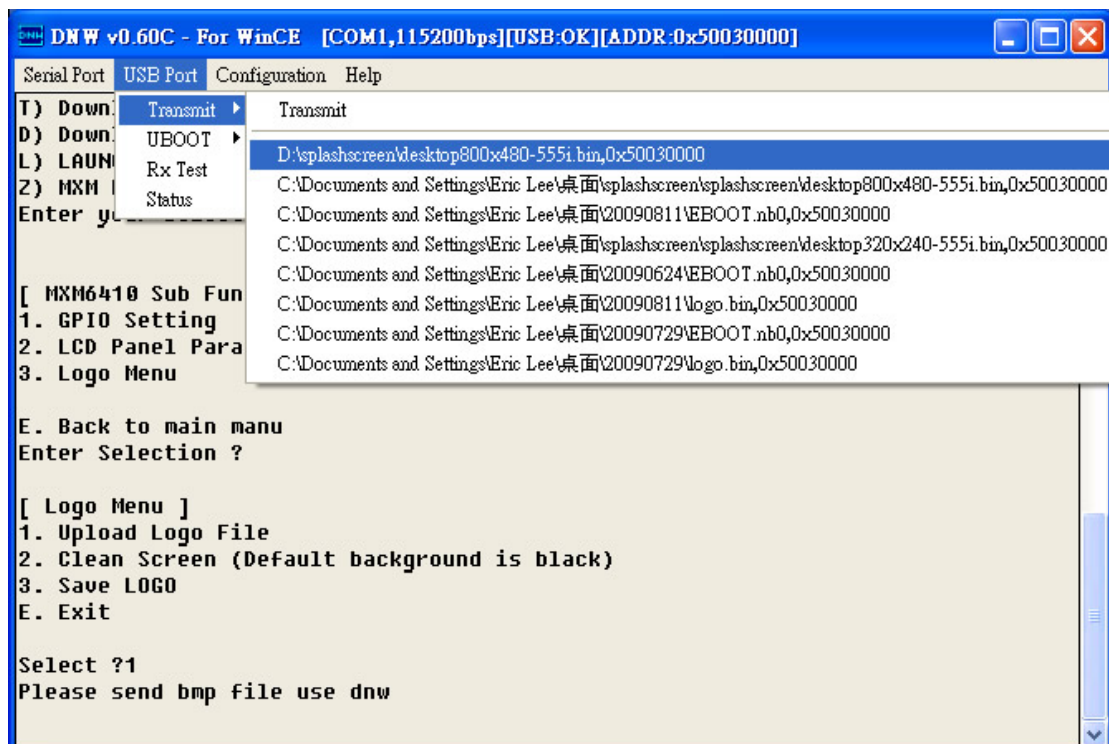
To upload the splash screen image into device, go to EBOOT menu and press “**Z) MXM EVB Board SubTestMenu**” and then press “**3. Logo Menu**”. Users will see sub menu as shown in figure 3.33.

Figure 3.33 EBOOT LOGO Menu



The “**1. Upload Logo File**” option can let users upload the splash screen that just created into the device and shown on the LCD screen immediately. To achieve that, at “**USB Port**” → “**Transmit**” of DNW program, choose the splash screen that just created and you will see the new splash screen shown on the LCD screen.

Figure 3.34 Upload Splash Screen



The “**2. Clean Screen (Default background is black)**” allows users to clean the splash screen that you just uploaded.

The “**3. Save LOGO**” will save the splash screen to NAND flash permanently.

Note: The maximum size of splash screen now is limited at $800 \times 600 \times 2 = 768000$ Byte. For file larger than this size will erase the EBOOT image.

Reboot the device and you should be able to see the new splash screen.

Figure 3.35 New Splash Screen



Chapter

4

Use MXM-6410/APC-6410 Hardware MFC Multimedia Function

This Chapter gives how to use the hardware MFC multimedia function.

Section include :

- WMV9 Decoder for Windows Media Player
- MFC Device Driver's API
- OpenGL ES User's Guide

Chapter 4 Use MXM-6410/APC-6410 Hardware MFC Multimedia Function

This Chapter gives topics related to hardware MFC multimedia function in MXM-6410/APC-6410.

4.1. WMV9 Decoder for Windows Media Player

The Windows media player in Windows CE 6.0 devices supports WMV9 hardware decoder. And Microsoft provides with a free WMV9 PC encoding tool to encode the video sources. First of all, we need to encode the video source as WMV9 format.

4.1.1. Encode Video source as WMV9

Users can download the free Windows Media Encoding tool that Microsoft provided with from

<http://www.microsoft.com/downloads/details.aspx?FamilyID=5691ba02-e496-465a-bba9-b2f1182cdf24&displaylang=en>

The recommended resolutions for MXM-6410/APC-6410 hardware MFC is 640x480 or 720x480. And the recommended bit rate is around 2000Kbps or less and frame rate is 30fps. It will take a while to encode the video sources. After encoding, you should be able to see a .wmv output file at the target directory of your PC.

4.1.2. WMV9 Decoder for Windows Media Player

Using ActiveSync. upload the wmv file into the device \NandFlash or \Storage Card directory. At devices, go to **My Device** → **NandFlash** or **My Device** → **Storage Card** and double click the file that you just upload. You will see the Media Player software playing the video.

Note: Please remember to close remote display tool when playing video because the remote display tool takes a lot of bandwidth.

4.2. MFC Device Driver's API

API Functions	Description
CreateFile	Create the MXM-6410 MFC instance.
DeviceIoControl	IOCTL_MFC_MPEG4_DEC_INIT IOCTL_MFC_MPEG4_ENC_INIT IOCTL_MFC_MPEG4_DEC_EXE IOCTL_MFC_MPEG4_ENC_EXE IOCTL_MFC_H264_DEC_INIT IOCTL_MFC_H264_ENC_INIT IOCTL_MFC_H264_DEC_EXE IOCTL_MFC_H264_ENC_EXE IOCTL_MFC_H263_DEC_INIT IOCTL_MFC_H263_ENC_INIT IOCTL_MFC_H263_DEC_EXE IOCTL_MFC_H263_ENC_EXE IOCTL_MFC_VC1_DEC_INIT IOCTL_MFC_VC1_DEC_EXE IOCTL_MFC_GET_LINE_BUF_ADDR IOCTL_MFC_GET_RING_BUF_ADDR IOCTL_MFC_GET_FRAM_BUF_ADDR
CloseHandle	Close the 6410 MFC instance.

4.2.1. CreateFile

CreateFile	
<i>Syntax</i>	<pre>HANDLE WINAPI CreateFile(LPCTSTR lpFileName, DWORD dwDesiredAccess, DWORD dwShareMode, LPSECURITY_ATTRIBUTES lpSecurityAttributes, DWORD dwCreationDisposition, DWORD dwFlagsAndAttributes, HANDLE hTemplateFile);</pre>
<i>Description</i>	This function creates the 6410 MFC instance. Several MFC instance can be made simultaneously. This means that CreateFile function can be called several times in a process (task).
<i>Parameters</i>	<p>lpFileName [IN] : MFC's device driver name. (L"MFC1:")</p> <p>dwDesiredAccess [IN] : GENERIC_READ GENERIC_WRITE</p> <p>dwShareMode [IN] : 0</p> <p>lpSecurityAttributes [IN] : NULL</p> <p>dwCreationDisposition [IN] : OPEN_EXISTING</p> <p>dwFlagsAndAttributes [IN] : FILE_ATTRIBUTE_NORMAL</p> <p>hTemplateFile [IN] : NULL</p>
<i>Returns</i>	HANDLE of the MFC instance. If it fails, it returns INVALID_HANDLE_VALUE.

4.2.2. DeviceIoControl

DeviceIoControl	
<i>Syntax</i>	BOOL WINAPI DeviceIoControl(HANDLE hDevice, DWORD dwIoControlCode, LPVOID lpInBuffer, DWORD nInBufferSize, LPVOID lpOutBuffer, DWORD nOutBufferSize, LPDWORD lpBytesReturned, LPOVERLAPPED lpOverlapped);
<i>Description</i>	Most of functions are developed in ioctl. This system call has many functions which is separated by dwIoControlCode
<i>Parameters</i>	hDevice [IN] : HANDLE returned by CreateFile() function dwIoControlCode [IN] : The control code for the operation. Detailed information will explain below. lpInBuffer [IN] : Structure of the MFC argument nInBufferSize [IN] : Size of MFC argument structure lpOutBuffer [OUT] : NULL nOutBufferSize [OUT] : 0 lpBytesReturned [OUT] : NULL lpOverlapped [IN] : NULL
<i>Returns</i>	If the operation completes successfully, the return value is nonzero. If the operation fails or is pending, the return value is zero.

4.2.3. CloseHandle

CloseHandle	
<i>Syntax</i>	BOOL WINAPI CloseHandle(HANDLE hDevice);
<i>Description</i>	Closes an open MFC's handle.
<i>Parameters</i>	[IN] hDevice - HANDLE returned by CreateFile() function
<i>Returns</i>	If the function succeeds, the return value is nonzero. If the function fails, the return value is zero

4.2.4. Control Codes for DeviceIoControl()

<i>IOCTL_MFC_MPEG4_DEC_INIT</i> <i>IOCTL_MFC_H263_DEC_INIT</i> <i>IOCTL_MFC_H264_DEC_INIT</i> <i>IOCTL_MFC_VC1_DEC_INIT</i>	
<i>Syntax</i>	BOOL WINAPI DeviceIoControl(HANDLE hDevice, DWORD dwIoControlCode, LPVOID lpInBuffer, DWORD nInBufferSize, LPVOID lpOutBuffer, DWORD nOutBufferSize, LPDWORD lpBytesReturned, LPOVERLAPPED lpOverlapped);
<i>Description</i>	It initializes the MFC's instance with the configure stream.
<i>Parameters</i>	hDevice [IN] : HANDLE returned by CreateFile() function dwIoControlCode [IN] : IOCTL_MFC_MPEG4_DEC_INIT, IOCTL_MFC_H263_DEC_INIT, IOCTL_MFC_H264_DEC_INIT, IOCTL_MFC_VC1_DEC_INIT lpInBuffer [IN] : Pointer to MFC_DEC_INIT_ARG structure. nInBufferSize [IN] : sizeof(MFC_DEC_INIT_ARG) lpOutBuffer [OUT] : NULL nOutBufferSize [OUT] : 0 lpBytesReturned [OUT] : NULL lpOverlapped [IN] : NULL
<i>Returns</i>	If the operation completes successfully, the return value is nonzero. If the operation fails or is pending, the return value is zero.

<i>IOCTL_MFC_MPEG4_DEC_EXE</i> <i>IOCTL_MFC_H263_DEC_EXE</i> <i>IOCTL_MFC_H264_DEC_EXE</i> <i>IOCTL_MFC_VC1_DEC_EXE</i>	
<i>Syntax</i>	BOOL WINAPI DeviceIoControl(HANDLE hDevice, DWORD dwIoControlCode, LPVOID lpInBuffer, DWORD nInBufferSize, LPVOID lpOutBuffer, DWORD nOutBufferSize, LPDWORD lpBytesReturned, LPOVERLAPPED lpOverlapped);
<i>Description</i>	It decodes the stream in the LINE_BUF or RING_BUF.
<i>Parameters</i>	hDevice [IN] : HANDLE returned by CreateFile() function dwIoControlCode [IN] : IOCTL_MFC_MPEG4_DEC_EXE, IOCTL_MFC_H263_DEC_EXE, IOCTL_MFC_H264_DEC_EXE, IOCTL_MFC_VC1_DEC_EXE lpInBuffer [IN] : Pointer to MFC_DEC_EXE_ARG structure. nInBufferSize [IN] : sizeof(MFC_DEC_EXE_ARG) lpOutBuffer [OUT] : NULL nOutBufferSize [OUT] : 0 lpBytesReturned [OUT] : NULL lpOverlapped [IN] : NULL
<i>Returns</i>	If the operation completes successfully, the return value is nonzero. If the operation fails or is pending, the return value is zero.

IOCTL_MFC_GET_LINE_BUF_ADDR IOCTL_MFC_GET_RING_BUF_ADDR IOCTL_MFC_GET_FRAM_BUF_ADDR	
<i>Syntax</i>	BOOL WINAPI DeviceIoControl(HANDLE hDevice, DWORD dwIoControlCode, LPVOID lpInBuffer, DWORD nInBufferSize, LPVOID lpOutBuffer, DWORD nOutBufferSize, LPDWORD lpBytesReturned, LPOVERLAPPED lpOverlapped);
<i>Description</i>	It obtains the address of the LINE_BUF, RING_BUF or FRAM_BUF.
<i>Parameters</i>	hDevice [IN] : HANDLE returned by CreateFile() function dwIoControlCode [IN] : IOCTL_MFC_GET_LINE_BUF_ADDR, IOCTL_MFC_GET_RING_BUF_ADDR, IOCTL_MFC_GET_FRAM_BUF_ADDR lpInBuffer [IN] : Pointer to MFC_GET_BUF_ADDR_ARG structure. nInBufferSize [IN] : sizeof(MFC_GET_BUF_ADDR_ARG) lpOutBuffer [OUT] : NULL nOutBufferSize [OUT] : 0 lpBytesReturned [OUT] : NULL lpOverlapped [IN] : NULL
<i>Returns</i>	If the operation completes successfully, the return value is nonzero. If the operation fails or is pending, the return value is zero.

4.2.5. Data Structure for Passing the IOCTL Arguments

4.2.5.1. MFC_ENC_INIT_ARG

MFC_ENC_INIT_ARG	
int ret_code	[OUT] Return code
int in_width	[IN] width of YUV420 frame to be encoded
int in_height	[IN] height of YUV420 frame to be encoded
int in_bitrate	[IN] Encoding parameter: Bitrate (kbps)
int in_gopNum	[IN] Encoding parameter: GOP Number (interval of I-frame)
int in_frameRateRes	[IN] Encoding parameter: Frame rate (Res)
int in_frameRateDiv	[IN] Encoding parameter: Frame rate (Divider)

4.2.5.2. MFC_ENC_EXE_ARG

MFC_ENC_EXE_ARG	
int ret_code	[OUT] Return code
int out_encoded_size	[OUT] Length of Encoded video stream

4.2.5.3. MFC_DEC_INIT_ARG

MFC_DEC_INIT_ARG	
int ret_code	[OUT] Return code
int in_strmSize	[IN] Size of video stream filled in STRM_BUF
int out_width	[OUT] width of YUV420 frame
int out_height	[OUT] height of YUV420 frame

4.2.5.4. MFC_DEC_EXE_ARG

MFC_DEC_EXE_ARG	
int ret_code	[OUT] Return code
int in_strmSize	[IN] Size of video stream filled in STRM_BUF

4.2.5.5. MFC_GET_BUF_ADDR_ARG

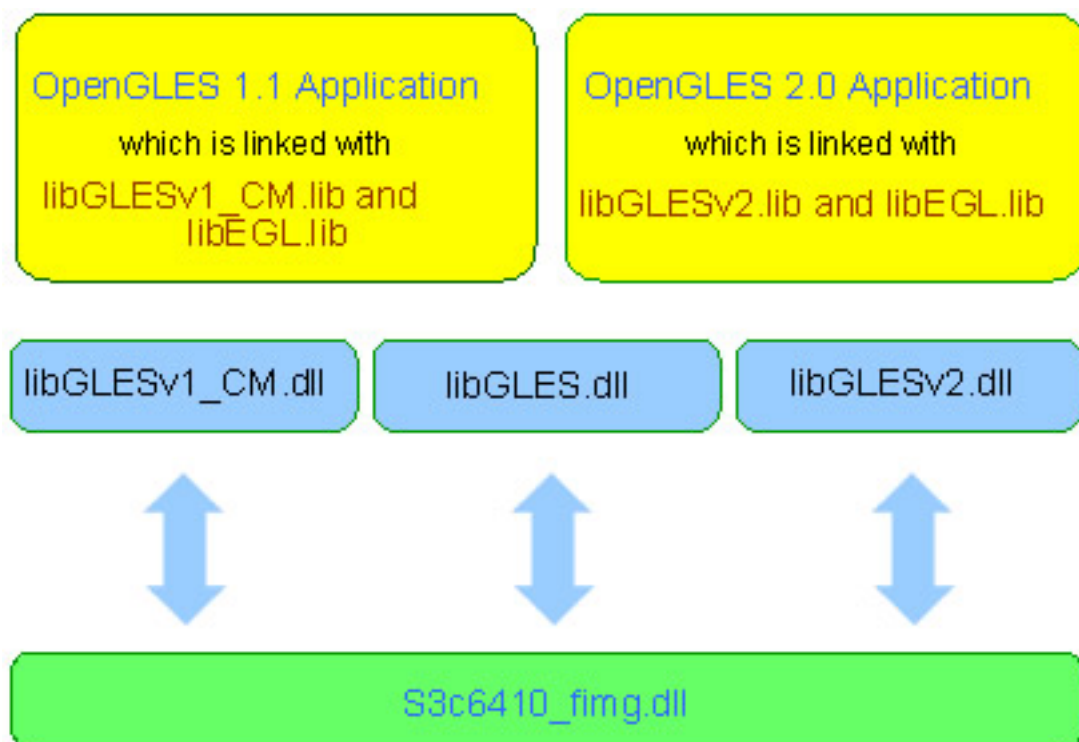
MFC_DEC_INIT_ARG	
int ret_code	[OUT] Return code
int in_usr_data	[IN] User data for translating Kernel-mode address to Usermode address
int out_buf_addr	[OUT] Buffer address
int out_buf_size	[OUT] Size of buffer address

4.3. OpenGL ES User's Manual

4.3.1. Software Layers

OpenGL ES driver has two layers. One is for mapping H/W and allocating memory. The other one is for OpenGL ES libraries.

Figure 4.1 MXM-6410/APC-6410 OpenGL ES Software Layer



The Figure 4.1 shows this driver structure. S3c6410_fimg.dll is mapping and allocating layer. This driver maps H/W address for library. The library uses this mapped address. And it also allocates physically continuous memory for

library and FIMG H/W. If the library requests some texture memory and depth buffer, then this driver dynamically allocates some memory and returns the address.

The libEGL.lib implements EGL 1.3. This will work with OpenGL ES1.1 and OpenGL ES2.0

The libGLESv1_CM.dll implements OpenGL ES1.1. If you want to know more about OpenGL ES 1.1, please visit www.khronos.org/opengl/ website.

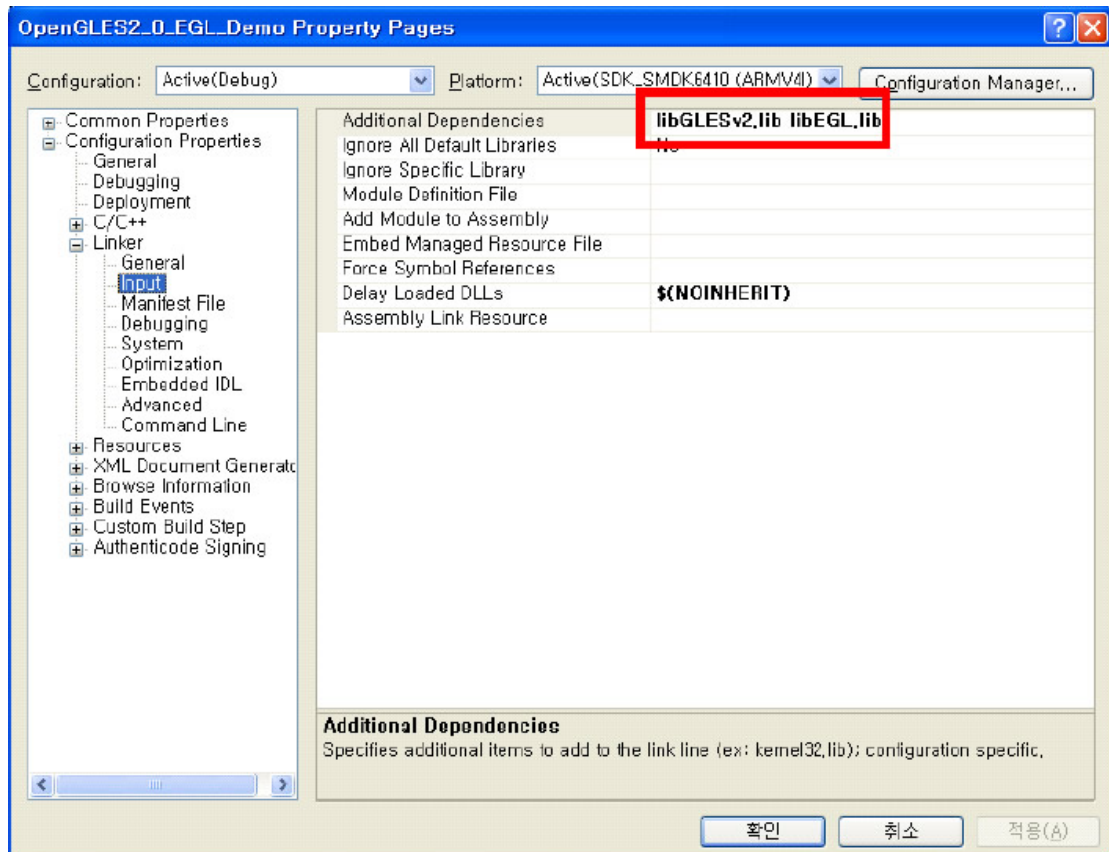
The libGLESv2.dll implements OpenGL ES2.0. If you want to know more about OpenGL ES 1.1, please visit www.khronos.org/opengl/ website.

4.3.2. Usage

For using OpenGL ES, You need to link libEGL.lib and libGLESv2.lib libraries.

In Project property of Visual Studio 2005, set Additional Dependency like the following figure 4.2.

Figure 4.2 Setting Property for OPENGLES2.0 in VS2005



If you need to use OpenGL ES 1.1, then you need to input libGLESv1_CM.lib instead of libGLESv2.lib.

4.3.3. Shader Compile

You need to use orion compiler for compiling shader. You can execute attached compiler on MS Windows XP and VISTA.

Ex)

orion -O -f <Fragment Shader File>

orion -O -v <Vertex Shader File>

orion -O -a -f <Fragment Shader ASM File>

orion -O -a -v <Fragment Shader ASM File>

If you type just “orion”, you can see more detail information about this shader compiler.

4.3.4. DLL location

libEGL.dll, libGLv1_CM.dll and libGLv2.dll should be in \Windows directory or same directory with your application. Second has higher priority.

4.3.5. Samples

4.3.5.1. OpenGL ES 1.1 Demo Application

The demo program is located at Embedian ftp site and it is a very simple opengles demo. You can get example about OpenGL ES application and compile environment from source code.

You can modify line 80 “int selectedDemo” from 0 to 7.

4.3.5.2. OpenGL ES 2.0 Demo Application

The demo program is located at Embedian ftp site and it is a very simple opengles demo. You can get example about OpenGL ES application and compile environment from source code.

You can modify line 59 “int selectedDemo” from 0 to 2.