UC-7400-CE User's Manual

First Edition, March 2006

www.moxa.com/product

Moxa Systems Co., Ltd.

Tel: +886-2-8919-1711
Fax:+886-2-8919-1722
Web: www.moxa.com

MOXA Technical Support
Worldwide: support@moxa.com.tw
The Americas support@moxa.com

UC-7400-CE User's Manual

The software described in this manual is furnished under a license agreement and may be used only in accordance with the terms of that agreement.

Table of Contents

1 Introduction

Microsoft® Windows® CE 5.0 is an open, scalable, 32-bit operating system (OS) that allows you to build a wide range of innovative, small footprint devices. A typical Windows® CE-based device is designed for a specific use, and often runs disconnected from other computers, or distributed as a front-end to a centralized host. Examples include enterprise tools, such as industrial controllers, communications hubs, point-of-sale terminals, and display devices, such as HMI, advertisement appliances, and interactive panels.

MOXA pays attention to the requirements of Windows® developers to provide the Windows® CE solution for the MOXA UC-7400 series ready-to-run embedded computer, including UC-7420, UC-7410 and UC-7408. The MOXA professional kernel development experience on embedded small footprint communication devices provides the intense technological skills required while porting the Windows® CE 5.0 kernel.

The following topics are covered in this chapter:

Overview

The MOXA UC-7400-CE Series (herein after referred to as UC-7400-CE) includes UC-7420-CE, UC-7410-CE, and UC-7408-CE. These RISC-based ready-to-run embedded computers are ideal for embedded applications. UC-7400-CE features 8 RS-232/422/485 serial ports, dual 10/100 Mbps Ethernet ports, 8 digital input channels and 8 digital output channels, a PCMCIA interface for wireless LAN communication, a CompactFlash port for flash disk expansion, and USB ports for adding additional memory (such as a USB Flash disk).

UC-7400-CE uses an Intel XScale IXP-422 266 MHz RISC CPU. Unlike the X86 CPU, which uses a CISC design, the IXP-422's RISC design architecture and modern semiconductor technology provide UC-7400-CE with a powerful computing engine and communication functions, but without generating a lot of heat. The built-in 32 MB NOR Flash ROM and 128 MB SDRAM give you enough memory to run your application software directly on UC-7400. As the dual LAN ports are built right into the IXP-422 CPU, UC-7400-CE makes an ideal communication platform for Network Security applications. If your application requires placing UC-7400-CE at a site that is not located near an Ethernet LAN connection, you can connect to the network by using UC-7400's PCMCIA port to attach a wireless LAN card.

UC-7400-CE series ready-to-run embedded computers are built on Microsoft® Windows® CE 5.0 operating system (OS). Building on a common, widely used environment makes them suitable for new system development and legacy system migration.

All the necessary device drivers, such as a PCMCIA Wireless LAN module and Keypad, LCM, and Buzzer control, are included with UC-7400-CE. The Operating System, device drivers, and the software you develop for your own application, can all be stored in UC-7400's Flash memory.

Model Descriptions and Package Checklist

The basic features of each UC-7400-CE product are described below:

UC-7420-CE
RISC-based ready-to-run embedded computer with 8 Serial Ports, Dual Ethernet, PCMCIA, CompactFlash, USB, WinCE 5.0

UC-7410-CE
RISC-based ready-to-run embedded computer with 8 Serial Ports, Dual Ethernet, WinCE 5.0

UC-7408-CE

RISC-based Data Acquisition embedded computer with 8 Serial Ports, 8 DI Channels, 8 DO Channels, Dual Ethernet, PCMCIA, CompactFlash, WinCE 5.0

UC-7400-CE is shipped with the following items:

- 1 UC-7400 series embedded computer
- Wall-Mounting Kit
- DIN-Rail Mounting Kit
- Quick Installation Guide
- Document & Software CD
- Cross-over Ethernet cable
- CBL-RJ45M9-150: 150 cm, 8-pin RJ45 to Male DB9 serial port cable
- CBL-RJ45F9-150: 150 cm, 8-pin RJ45 to Female DB9 console port cable
- Universal Power Adaptor
- Product Warranty Booklet

NOTE: *Notify your sales representative if any of the above items is missing or damaged.*

UC-7400 Product Features

- Intel XScale IXP-422 266 MHz Processor
- On-board 128 MB RAM, 32 MB Flash ROM
- Eight RS-232/422/485 serial ports
- 8 digital input channels and 8 digital output channels (UC-7408 only)
- Dual 10/100 Mbps Ethernet
- USB 2.0 host for mass storage devices (UC-7420 only)
- PCMCIA, wireless LAN expansion (supports 802.11b/802.11g)
- CompactFlash for storage expansion (UC-7420/7408 only)
- LCM display and keypad for HMI (UC-7420/7410 only)
- Ready-to-run WinCE 5.0 .NET platform
- DIN-Rail or wall mounting installation
- Robust, fanless design

UC-7420/UC-7410

UC-7408

UC-7420-CE and UC-7408-CE each add a PCMCIA slot and a CompactFlash interface for wireless communication and flash storage expansion. Furthermore, UC-7420-CE makes itself an all-in-one device by integrating two USB hosts, a USB device port, and a LCM display for mass storage capability and others. UC-7408-CE, without LCM and USB support, features an additional 8 digital input and 8 digital output channels, making it an ideal yet cost-effective embedded computer for data acquisition systems. Finally, UC-7410-CE is a shrink-down version of UC-7420-CE without PCMCIA, CompactFlash, and USB functions in order to meet low-cost requirements for smaller applications. You can refer to the following comparison table of product hardware specifications to get more detail.

Product Hardware Specifications

	UC-7420	UC-7410	UC-7408
CPU	Intel XScale IXP-422 266MHz	Intel XScale IXP-422 266MHz	Intel XScale IXP-422 266MHz
RAM	128 MB	128 MB	128 MB
Flash	32 MB	32 MB	32 MB
LAN	Auto-sensing 10/100 Mbps × 2 with built-in 1.5 KV magnetic isolation protection, RJ45 Connector		
Serial Port	RS-232/422/485 × 8 RJ45 Connector	RS-232/422/485 × 8 RJ45 Connector	RS-232/422/485 × 8 RJ45 Connector
Serial Protection	15 KV ESD for all signals		
Data Bits	5, 6, 7, 8		
Stop Bits	1, 1.5, 2		
Parity	none, even, odd, space, mark		
Flow Control	RTS/CTS, XON/XOFF, RS-485 ADDC™		
Speed	50 bps to 921.6 Kbps		
Serial Console	RS-232 × 1 RJ45 Connector	RS-232 × 1 RJ45 Connector	RS-232 × 1 RJ45 Connector
DI/DO	N/A	DI × 8, DO × 8	N/A
USB 2.0 Hosts	2	N/A	N/A
USB 1.1 Client	1*	1*	1*
PCMCIA	Cardbus × 1	N/A	Cardbus × 1
Storage Expansion	Compact Flash × 1**	N/A	Compact Flash × 1**
LCM	128 × 64 dots	128 × 64 dots	N/A
Keypad	5	5	N/A
Real Time Clock	Yes	Yes	Yes
Buzzer	Yes	Yes	Yes
Rest Button	HW Reset × 1 Reset to Default × 1	HW Reset × 1 Reset to Default × 1	HW Reset × 1 Reset to Default × 1
Power Input	12 to 48 VDC	12 to 48 VDC	12 to 48 VDC
Power Consumption	12W	10W	8W
Dimension (W x D x H)	197 × 125 × 44 mm	197 × 125 × 44 mm	197 × 125 × 44 mm
Weights	875 g	810 g	870 g
Operating temperature	-10 to 60°C, (14 to 140°F), 5 to 95% RH		
Storage temperature	-20 to 80°C, (-4 to 176°F), 5 to 95% RH		
Regulatory Approvals	EMC: CE Class A, FCC Class A Safety: UL, cUL, TUV		
Warranty	5 years	5 years	5 years

* USB Client function is reserved for future enhancement

** CompactFlash is designed for Flash memory card or Microdrive

UC-7400-CE Software Features

UC-7400-CE ready-to-run embedded computers are network centric / head-less computers that are designed to be programmable for embedded communication applications. The following are the software features of UC-7400-CE:

Applications Development Environment

To make UC-7400-CE an easy-to-use programming environment, its Windows® CE environment provides the following common, popular application development features that make programming convenient and easy as in a PC environment.

- **C Libraries and Run-times -** Compared to the C libraries and run-times used on a desktop PC running Windows®, the C libraries and run-times on a UC-7400-CE is a subset of the WIN32 APIs. It supports full ANSI C run time, standard input/output library, standard input/output ASCII library and standard ASCII string functions. In addition, it supports compiler C++ exception handling equivalent and Run-Time Type Information (RTTI) equivalent to the desktop C++ compilers.
- **Component Services (COM and DCOM) -** The Common Object Model (COM) is an operating system-independent, object-oriented system for creating binary software components that can interact with other COM-based components in the same process space, in other processes, or on remote machines.
- **Microsoft® Foundation Classes (MFC) -** MFC is a comprehensive class library and complete object-oriented application framework designed to help build applications, COM components, and controls.
- **SOAP Toolkit -** SOAP is an XML-based protocol for object exchange and remote procedure calls. Microsoft® Windows® CE 5.0 provides functionality similar to the SOAP Toolkit version 2 on the desktop. It provides a layer that allows COM objects to use SOAP as the transport protocol for remote procedure calls and to interact with Web services.
- **Microsoft® .NET Compact Framework 2.0 -** It offers a choice of languages, initially Microsoft® Visual Basic® and Microsoft® Visual C#, and eliminates the common problems faced with language interoperability.
- **XML-** Provides the Document Object Model (DOM) for base XML functionality, support for XML Query Language (XQL) and XPATH, Extensible Style Sheet Language Transformations (XSLT) that enables you to transform one class of XML document to another, SAX2 support for event-based parsing of XML documents and includes MSXML Writer, and parsing based on Simple API for XML (SAX) for resource-constrained target devices.
- **Winsock 2.2 -** Provides enhanced capabilities over Winsock 1.1, including installable service providers for additional third-party protocols, as well as Media sense.

Networking and Communications Capabilities

For network centric embedded application usage, UC-7400-CE, not only provides powerful communication hardware interfaces including dual Ethernet and 3-in-I serial ports, but also supports the networking and communications capabilities that are built-in to Windows® CE 5.0 OS. The features that are well supported are listed as below.

- **Simple Network Management Protocol (SNMP) -** Monitors remote connections to the network.
- **Simple Network Time Protocol (SNTP) Client -** Provides support for synchronizing the device's system time with a SNTP server, and supports Daylight Saving Time.
- **Serial Communications -** In addition to the 16550 UART driver bound to a debug port and the console port, it includes a special driver for 8 additional MOXA home-made serial ports.
- **Network Utilities (IpConfig, Ping, Route) -** Utilities for troubleshooting various network problems.
- **TCP/IP -** Includes IP, Address Resolution (ARP), Internet Control Message (ICMP), Internet Group Membership (IGMP), Transmission Control (TCP), User Datagram (UDP), name resolution and registration, and DHCP.

Supporting Servers and Daemons

In addition to the development and communication capability, UC-7400-CE embeds the services and daemons as stated next. These common and easy-to-use application servers help users to migrate the UC-7400-CE embedded computer to the industrial communication application very easily and conveniently.

- **Telnet Server -** A sample server that allows remote administration through a standard telnet client.
- **FTP Server -** A sample server used for transferring files to and from remote computer systems over a network using TCP/IP.
- **Web Server (HTTPD) -** Includes ASP, ISAPI Secure Socket Layer support, SSL 2, SSL 3, and Transport Layer Security (TLS/SSL 3.1) public key-based protocols, and Web Administration ISAPI Extensions.
- **Dial-up Networking -** Consists of RAS client API and the Point to Point Protocol (PPP). RAS and PPP support Extensible Authentication Protocol (EAP) and RAS scripting.

Learning Firmware Build Versions

There are three ways to obtain the firmware version of the UC-7400-CE computer. This information is particularly important for the purpose of feature identification.

- Examine the welcome message after you logon the computer.
- Logon the Web-based management system (described in a later chapter) to view the system information.

In addition, for a UC-7420-CE or UC-7410, you can obtain the firmware version by pressing function key F1 and then examining this information on the LCM display.

Memory and File Systems

From the 128M bytes of SDRAM space, the main memory has a capacity of about 112M bytes in which the operating system and user applications run. The kernel image occupies the rest of the space.

The internal file system in the UC-7400-CE computer controls access to ROM and also provides file storage in the object store, which is in the RAM. The ROM file system provides persistent storage for applications and their related data even when the main power supply is lost. It integrates the read-only files that are stored in Flash ROM with the read/write files of both an application and a user. In the UC-7400-CE computer, a child directory named "NORFlash" under the root indicates the ROM storage of the flash memory of size 15.5M bytes.

The root directory is a RAM file system of size 15.5M bytes. It can be used for storing temporary files for your applications. However, do not place persistent files or applications in the root directory because they will be wiped out when the system is shutdown. Instead, place them under the directory "NORFlash".

The internal file system supports TFAT (Transaction-Safe File Allocation Table), which is a safer mechanism than FAT. TFAT protects the file system from write corruption in critical times such as power loss. During power loss, the file state is rolled back. In addition, TFAT is a superset of FAT. This means that the internal file system of the computer provides the ability to mount external FAT file systems.

The additional file systems on USB and CompactFlash storage devices are placed at the root of the internal file system. If you intend to use these devices to port data between your PC and the UC-7400-CE computer, please format them as the FAT file system on your PC. These devices need to be formatted as FAT as otherwise the PC may not recognize the TFAT format.

Hive-Based Registry—Contrast to RAM-Based Registry

The registry for the UC-7400-CE is a hive-based registry in contrast to a RAM-based registry. The hive-based registry stores registry data inside files, or hives, which can be kept on any file system. This removes the need for performing backup and restore on power off.

Inserting a CompactFlash Card into a UC-7400-CE Computer

The UC-7400-CE (except for UC-7410-CE) is equipped with a CompactFlash slot of type II which supports cards of both types - I and II. A mass storage card is considered to be a standard attachment to the computer. Thus, when an empty mass storage card is inserted into the slot, the computer automatically formats it to the FAT system. This process takes a few minutes to complete. The UC-7400-CE, when a mass storage card is inserted, creates a directory named "CFFloder" under the root directory and the newly created directory serves a link to the storage.

Inserting a USB Mass Storage into a UC-7400-CE Computer

When an empty USB storage device is plugged into the USB slot at the back of the UC-7400-CE (only for UC-7420-CE), the computer automatically formats it to the FAT system instead of the TFAT system as the USB storage device is considered to be highly portable between your PC and the computer.

When the first USB mass storage device is plugged in, a directory named "USBDisk" under the root directory is created as a link to the storage, on the internal file system. The directory created for the second plugged in USB device is "USBDisk2".

Eight RS-232/422/485 Serial Ports

The UC-7400-CE computer comes with two embedded serial ports, named "COM1" and "COM2". The former is hidden inside the box case for debugging purposes when developing firmware. The latter is used as the console port on which you can connect to the computer via a serial wire. Though the console port is available for you to develop applications such as data acquisition and control, we suggest that you implement them on the eight serial ports located on the front of the device.

These eight additional serial ports, from left to right, have the most common names starting from "COM3" to COM10". They are designed to provide reliability, high-speed and 3-in-1 (i.e., RS-232, RS-422, and RS-485) operation mode switch for your diverse applications. Each of these ports supports baudrate settings up to 921600 bps.

2
Getting Started

In this chapter, we explain how to operate a UC-7400-CE computer via a PC near you. For clarity, this PC is called a development workstation and the UC-7400-CE computer is called a target computer.

In addition, manual steps are described to facilitate operations such as system time adjustment, troubleshooting network connectivity, etc. Some of these operations can be done via system commands after gaining access to the computer and others can be done by a Web-based management system, which is described in a later chapter.

The following topics are covered in this chapter:

Starting Your UC-7400-CE Computer

Connect the SG wire to the shielded contact located at the upper left corner of the target computer, and then power on it by connecting it to the power adaptor. It takes about 30 to 60 seconds for the system to boot up. Once the system is ready, the “Ready” LED will light up and stay on till you shutdown the computer. For a computer with a LCM display, the network address settings and other system information will appear on the LCM.

Resetting Your UC-7400-CE Computer

Warm/Cold Start: You seldom need to perform a warm-start or a cold-start on a UC-7400-CE computer. In some cases where the computer stops responding, an application locks up or when you upgrade the firmware, it may be necessary to perform a warm-start or a cold-start.

Warm-Start: In power-on state, insert a pin into a “Reset” hole next to the serial ports and hold the pin for 1 or 2 seconds. The computer reboots itself.

Cold-Start: Switch off and then switch on the power again. The computer reboots itself right away.

Resetting to Factory Defaults: If the computer is not working properly, and you want to reset it back to factory default settings, press and hold the “Reset to Default” button for at least 5 seconds. The buzzer sounds while the factory default settings are loaded. After the factory default has been loaded, the computer reboots itself. Do not confuse this with the “Reset” button.

Operating UC-7400-CE Computer Via Serial Console

The serial console port (next to two LAN ports) gives users a convenient way of connecting the development workstation to the console utility of the target computer. This method is particularly useful when using the computer for the first time.

After you have wired a serial cable, go back to the development workstation and start a terminal program (e.g., HyperTerminal) by using the settings shown below for the serial console port.

Baud rate	115200bps
Parity	None
Data bits	8
Stop bits	1
Flow Control	None
Terminal	VT100

After a successful connection, type the login name and password as requested to logon the computer. The default values are both **“admin”**.

Login: admin

Password: admin

Changing the Network Settings

The UC-7400-CE computer comes with two network interfaces. The default IP addresses and netmasks of the network interfaces are as follows:

	Default IP Address	**Netmask**
LAN 1	192.168.3.127	255.255.255.0
LAN 2	192.168.4.127	255.255.255.0

Normally, you are required to change them because they are located on a different local network from that of your development workstation. Without changes, you cannot connect to them directly. Use the ***netconfig*** utility to complete the task. Type ***netconfig -h*** to get help on this utility.

```
\> netconfig -h
Usage: netconfig –n <"LAN1" or "LAN2"> [-m <netmask>] [-d <DNS server>] [-g <gateway>]
[-i <IP address>]
```

For example, your development workstation has a LAN port at 192.168.1.x and the Domain Name Server (DNS) is at 192.168.2.6. Execute the following command.

```
\> netconfig –n LAN1 –i 192.168.1.5 –m 255.255.255.0 –g 192.168.1.254 –d 192.168.2.6
```

Use command ***netconfig*** to view the new settings.

```
\> netconfig
LAN1 Interface Configuration:
IP Address:   192.168.1.5
SubNet Mask:  255.255.255.0
Gateway:    192.168.1.254
DNS:     192.168.2.6

LAN2 Interface Configuration:
IP Address:   192.168.4.127
SubNet Mask:  255.255.255.0
Gateway:
DNS:
```

Operating Your UC-7400-CE Computer Via Telnet Client

Before operating your UC-7400-CE computer using the Telnet client, we suggest that you change the network settings of the computer (see the earlier section) to have at least one of the two network ports situated in the same LAN as your development workstation.

Use a cross-over Ethernet cable to directly connect your development workstation to the target computer, or a straight-through Ethernet cable to connect the computer to a LAN hub or switch. Next, use a telnet client in your development workstation to connect to the telnet console utility of the target computer. Upon a successful connection, type the login name and password as requested to logon to the computer.

After logging in via the console port or a telnet client, you have a list of busybox commands available to operate the computer. Use ***HELP*** to display all the commands, or type ***HELP [command name]*** to display extended help for the selected command. Some of these commands such as ***DATE*** and ***TIME*** are very useful to you to easily manage the system time of the computer. Others commands such as ***DIR*** and ***MKDIR*** are good utilities for file management. For example, to inspect the file structure of the root directory, type ***DIR***

```
\> dir /b
NORFlash
My Documents
Program Files
Temp
Windows
```

User/Group Management

User Group: You should assign specific services, such as ftp and telnet, to defined user groups such that these services are accessible only by the users within the permissible user group. Three user groups, namely "ftpd", "telnetd", and "httpd", are already created by default for your convenience.

Adding a Group: Use the command ***useradd –g <groupName>*** to create a user group.

```
\> useradd –g yyyy
group yyyy has been added.
```

Deleting a Group: To remove a group, use the command ***userdel –g <groupName>***.

```
\> userdel –g yyyy
group yyyy has been removed.
```

Adding a User: Use the command ***useradd <newUserID>*** to add a user for accessing the system. The user's password, by default, is the same as the user name.

```
\> useradd xxxx
user xxxx has been added.
```

In addition, you can permit this user to access a particular service by typing ***-g*** followed by the user group name of the service, i.e., ***useradd –g <groupName> <newUserID>***. For example,

```
\> useradd –g telnetd xxxx
user xxxx is existent
group telnetd is existent
user xxxx has been added to group yyyy
```

Deleting a User: Use the command ***userdel <userID>*** to delete a user from the system. User "admin" **CANNOT** be deleted.

```
\> userdel xxxx
user xxxx has been deleted
```

You can also just remove a user from a user group by using the command ***userdel –g <groupName> <newUserID>***. For example,

```
\> userdel –g yyyy xxxx
user xxxx has been removed from group yyyy
```

Changing the Password: Use the command ***"passwd <userID>"*** to change your login password. Please type, The user's password, by default, is the same as the user name.

```
\> passwd xxxx
Current password:
New password:
Retype new password:
Password has been changed
```

Adjusting System Time and RTC Time

The UC-7400-CE computer has two time settings, the system time (or CPU clock) and the RTC (Real Time Clock) time. The system time regulates the execution of instructions and the RTC keeps track of the time even when the computer is turned off. RTC time runs on a special battery that is not connected to the normal power supply. Do not confuse a computer's real-time clock with its CPU clock.

Setting the System Time Manually: Use the ***date***, and ***time*** commands to query the current system date/time or to set a new system date/time.

```
\> date
The current date is: Tuesday, November 22, 2005
Enter the new date (mm-dd-[yy]yy): 12-23-05

\> date /T
Wednesday, November 23, 2005

\> time
The current time is: 5:27:17 PM
Enter the new time (hh:mm:ss): 16:02:00

\> time /T
4:02:04 PM
```

Adjusting RTC Time: Use the command ***hwclock -w*** to convert the system time to the RTC time.

```
\> hwclock -w
```

Use the command ***hwclock –w YYYY-MM-DD hh:mm:ss*** to manually update the RTC time.

```
\> hwclock –w 2005-12-23 15:00:00
```

Use the command ***hwclock*** to query the updated RTC time.

```
\> hwclock
2005-12-23 15:00:00
```

Starting and Stopping Services

After booting up, the UC-7400-CE computer runs several services continuously to serve requests from users or other programs. Notable services include telnet ("TEL0:"), console ("CON0:"), world wide web HTTP ("HTP0:"), file transfer FTP ("FTP0:") etc. You seldom need to care about these services. However, you still can start up or stop a service with its associated name by using the command "services". For example,

Start the FTP service by

```
\> services start FTP0:
```

Stop the FTP service by

```
\> services stop FTP0:
```

The default services in UC-7400-CE are listed as below:

TEL0:	Telnet Service
FTP0:	FTP Service
CON0:	Console Service

Troubleshooting Network Connectivity

The ipconfig tool prints the TCP/IP-related configuration data of a host including the IP addresses, gateway and DNS servers.

```
\> ipconfig /all
Windows IP configuration
Ethernet adapter Local Area Connection:
IP Address: 192.168.4.127
Subnet Mask: 255.255.255.0
Adapter Name: IXP425ETHNPE2
Description: IXP425ETHNPE2
Adapter Index: 2
Address: 80 86 33 33 34 12
DHCP Enabled: NO
Ethernet adapter Local Area Connection:
IP Address: 192.168.14.202
Subnet Mask: 255.255.248.0
Default Gateway: 192.168.15.254
Adapter Name: IXP425ETHNPE1
Description: IXP425ETHNPE1
Adapter Index: 3
Address: 78 56 34 91 cc dd
DHCP Enabled: NO

Host name: UC7420CE
Domain Name:
DNS Servers: 192.168.1.6
NODETYPE: 8
Routing Enabled: NO
Proxy Enabled: NO
```

To troubleshoot network connectivity, reachability, and name resolution, use the ping command. This command verifies IP-level connectivity to another TCP/IP computer by sending Internet Control Message Protocol (ICMP) Echo Request messages. The corresponding return Echo Reply messages are displayed, along with round-trip times. For more information, type ping without parameters.

```
\> ping www.moxa.com
Pinging Host www.moxa.com [192.168.1.16]
Reply from 192.168.1.16: Echo size=32 time<1ms TTL=126
Reply from 192.168.1.16: Echo size=32 time<1ms TTL=126
Reply from 192.168.1.16: Echo size=32 time<1ms TTL=126
```

The **route** utility allows you to view or modify network routing tables. Type this command without parameters to view a list of functions.

```
\> route
```

To view current routing items in the tables,

```
\> route PRINT
```

To add a routing item on network interface 1,

```
\> route ADD 192.168.0.0 MASK 255.255.0.0 192.168.15.254 IF 2
```

To delete a routing item,

```
\> route DELETE 192.168.0.0
```

Simple Network Management Protocol (SNMP)

SNMP is the Internet Standard protocol for network management and part of the TCP/IP protocol suite. SNMP was developed to monitor and manage networks. It uses a distributed architecture that consists of agents and managers:

SNMP agent

The SNMP agent is an SNMP application that monitors network traffic and responds to queries from SNMP manager applications. The agent also notifies the manager, by sending a trap, when significant events occur.

SNMP Manager

An SNMP manager is an SNMP application that generates queries to SNMP-agent applications and receives traps from SNMP-agent applications.

The UC-7400-CE computer installs an SNMP agent to serve as an SNMP device. You should install the SNMP manager on the workstation computer (for example, a Linux system) that monitors the network. After installing the nodes, you need to configure the SNMP manager and agent.

To check SNMP agent capabilities in a target UC-7400-CE (e.g, network IP at 192.168.3.127) computer, please logon the workstation computer that, for example, a Linux-based computer, the SNMP manager resides and type

```
\> snmpwalk -v 2c -c public 192.168.3.127 system
SNMPv2-MIB::sysDescr.0 Microsoft Windows CE Version 5.0 (Build 1400)
SNMPv2-MIB::sysObjectID.0 SNMPv2-SMI::enterprises.8691.13.7420
SNMPv2-MIB::sysUpTime.0 1282929
SNMPv2-MIB::sysContact.0 Your System Contact Here
SNMPv2-MIB::sysName.0 WindowsCE
```

You will see a series of messages from the SNMP agent in the UC-7400-CE computer. From there, you can monitor and manage the computer.

3 Web-based Management System

Note: Internet Explorer 5.5 and above is required to use the web based management system.

The UC-7400-CE series ready-to-run embedded computers are network-centric platforms and are designed to serve as excellent front-ends for data acquisition and industrial control. Due to the distributed characteristics of the devices, that these computers control, they often reside in harsh areas as the devices themselves and are away from system administrators. To manage these computers, operations such as networking/server configuration, file management, and process (thread) monitoring/control become a critical area to consider.

To resolve these management issues and accordingly reduce the toil of system administration, a web-based management system is installed into the UC-7400-CE computer. This system incorporates often-used features into CGI pages and categorizes them on a menu bar.

Before operating the system, please make sure you have a network connection from your PC to the target computer and can open an Internet browser at your PC after the connection. Then, use the IP address of the target computer as a home page URL. After the main page comes out, click on **Web-Based Management**.

Provide your authentication data including user ID and password into the corresponding fields of the prompt (case sensitive) and then hit the enter key to request access to the management system. The system checks your data with the users previously defined in the computer and then determines the validity of your logon.

The default User ID and Password are as follows:

User ID: admin

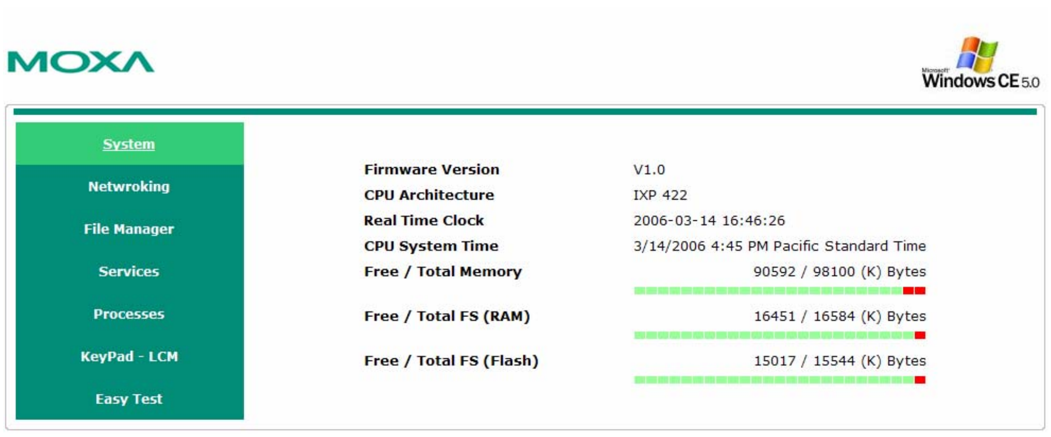
Password: admin

The following topics are covered in this chapter:

- ❑ **System Information**
- ❑ **Netowrking/Server Configuration**
- ❑ **Process (Thread) Monitoring/Control**
- ❑ **Launching Processes Automatically**
- ❑ **Services Monitoring/Control**
- ❑ **Binary/Text File Management**

System Information

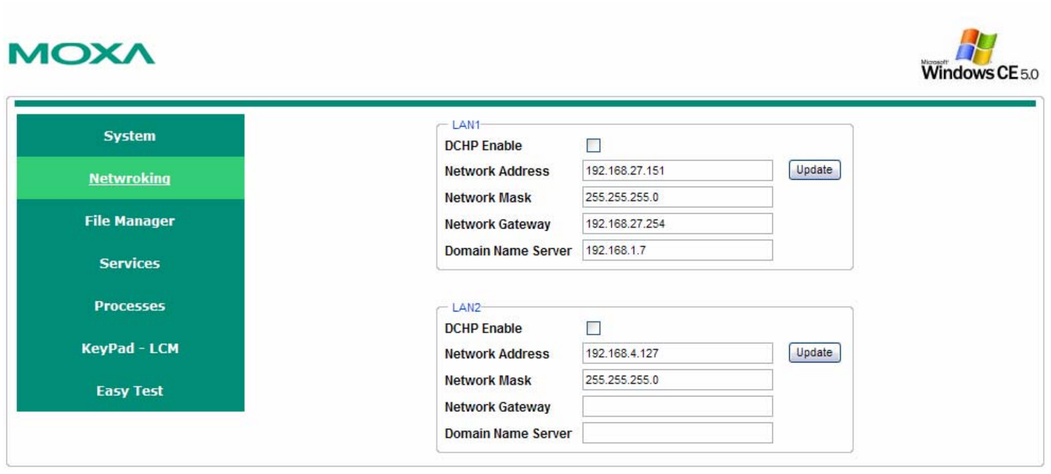
After you logon successfully, the main page displays the system information of the target UC-7400-CE computer, including the firmware version of the computer, the RTC time, the CPU system time, and system resources including main memory and file system usage (RAM and Flash).



Firmware Version	V1.0
CPU Architecture	IXP 422
Real Time Clock	2006-03-14 16:46:26
CPU System Time	3/14/2006 4:45 PM Pacific Standard Time
Free / Total Memory	90592 / 98100 (K) Bytes
Free / Total FS (RAM)	16451 / 16584 (K) Bytes
Free / Total FS (Flash)	15017 / 15544 (K) Bytes

Networking/Server Configuration

The UC-7400-CE computer has two network interfaces. To view or change their settings, click the ***Networking*** item on the menu bar. After the page loads, enter the relevant details on the corresponding text fields and then click "Update" to make the changes effective for the interfaces.



LAN1

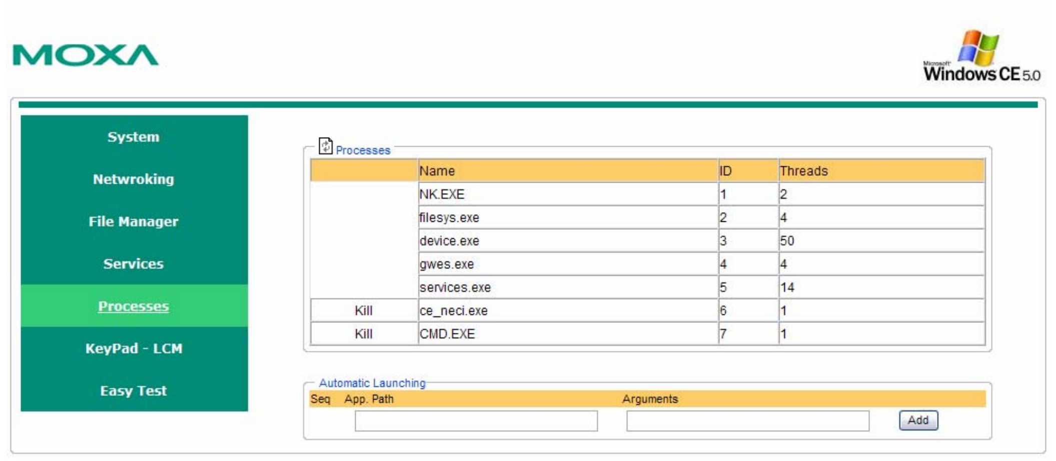
DCHP Enable	☐
Network Address	192.168.27.151
Network Mask	255.255.255.0
Network Gateway	192.168.27.254
Domain Name Server	192.168.1.7

LAN2

DCHP Enable	☐
Network Address	192.168.4.127
Network Mask	255.255.255.0
Network Gateway	
Domain Name Server	

Process (Thread) Monitoring/Control

At runtime, the UC-7400-CE computer manages up to 32 applications. You can use the management system to monitor and control them. To view current processes, please click the ***Processes*** item on the main menu bar. The running processes are then displayed. You can kill a process by clicking the "kill" button next to the process name.



	Name	ID	Threads
	NK.EXE	1	2
	filesys.exe	2	4
	device.exe	3	50
	gwes.exe	4	4
	services.exe	5	14
Kill	ce_neci.exe	6	1
Kill	CMD.EXE	7	1

Seq	App. Path	Arguments

Launching Processes Automatically

To have your application start on boot, do the following:

Step 1: Click the "Processes" item on the main menu bar. At the lower part of the page, there is an area marked as "Automatic Launching".

Step 2: Fill in the full path of the application in the first text field and its arguments in a separate text field if there are any.

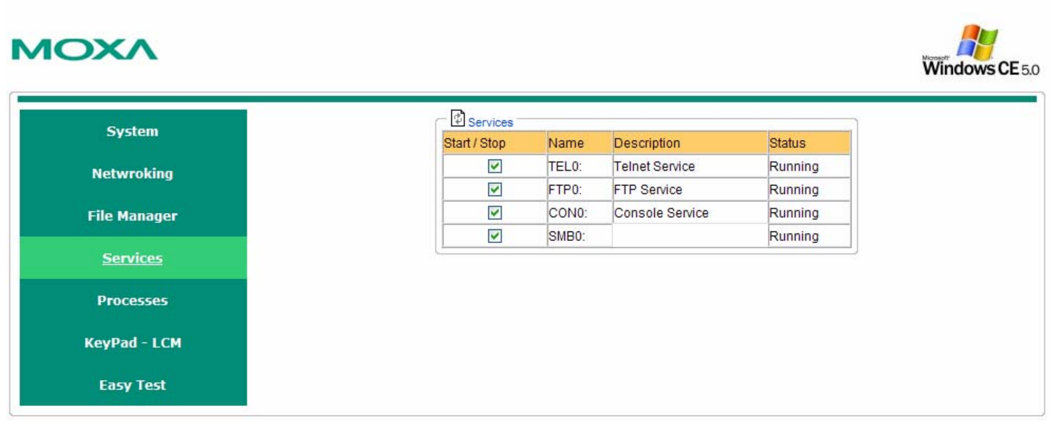
Step 3: Click "Add".

Services Monitoring/Control

Some services run on the background to provide services, such as ftp, telnet and etc, for user requests. To monitor and control these services, do the following:

Step 1: Click the "Services" item on the main menu bar. The running services are displayed.

Step 2: Click on a check box to toggle a start/stop operation for a service.



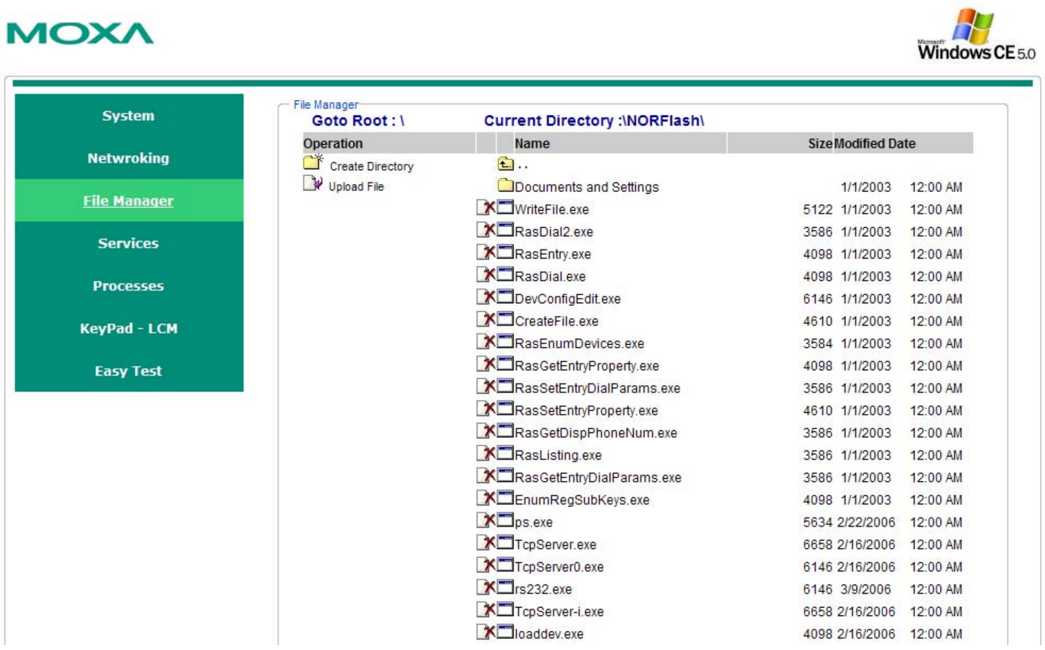
Start / Stop	Name	Description	Status
✓	TEL0:	Telnet Service	Running
✓	FTP0:	FTP Service	Running
✓	CON0:	Console Service	Running
✓	SMB0:		Running

Some listed services cannot be stopped in order to maintain normal operation of the computer. Such services do not have a check box next to them.

Binary/Text File Management

On a PC, it is certainly convenient to have a friendly window-based file manager to browse, delete, and organize files and directories. On the UC-7400-CE computer, such a convenient feature is simulated by the web-based management system. Just click "File Manager" to view the directory tree of your target UC-7400-CE computer. Using the File manager, you can perform the following operations:

- To browse a child directory, click the name of the directory.
- To delete a file, click the icon with an ***X*** in front of the file icon.
- To create a child directory, click ***Create Directory*** and then follow the instructions.
- To refresh the current directory, click ***Current Directory*** at the top of the page.



File Manager

Goto Root : \ Current Directory :\NORFlash\

Operation	Name	Size	Modified Date	
Create Directory	..			
Upload File	Documents and Settings		1/1/2003	12:00 AM
	WriteFile.exe	5122	1/1/2003	12:00 AM
	RasDial2.exe	3586	1/1/2003	12:00 AM
	RasEntry.exe	4098	1/1/2003	12:00 AM
	RasDial.exe	4098	1/1/2003	12:00 AM
	DevConfigEdit.exe	6146	1/1/2003	12:00 AM
	CreateFile.exe	4610	1/1/2003	12:00 AM
	RasEnumDevices.exe	3584	1/1/2003	12:00 AM
	RasGetEntryProperty.exe	4098	1/1/2003	12:00 AM
	RasSetEntryDialParams.exe	3586	1/1/2003	12:00 AM
	RasSetEntryProperty.exe	4610	1/1/2003	12:00 AM
	RasGetDispPhoneNum.exe	3586	1/1/2003	12:00 AM
	RasListing.exe	3586	1/1/2003	12:00 AM
	RasGetEntryDialParams.exe	3586	1/1/2003	12:00 AM
	EnumRegSubKeys.exe	4098	1/1/2003	12:00 AM
	ps.exe	5634	2/22/2006	12:00 AM
	TcpServer.exe	6658	2/16/2006	12:00 AM
	TcpServer0.exe	6146	2/16/2006	12:00 AM
	rs232.exe	6146	3/9/2006	12:00 AM
	TcpServer-i.exe	6658	2/16/2006	12:00 AM
	loaddev.exe	4098	2/16/2006	12:00 AM

In addition, the management system offers a mechanism for file upload. This mechanism helps you transfer files from your workstation to the target computer in an easy way. For instance, after you have built an application on the development workstation, you can use this mechanism to upload the application to the current directory of the target computer.

Step 1: Click "Upload File". A browser window pops up.

Step 2: From the pop-up browser window, click "Browse" to bring up a local file manager.

Step 3: Browse to and select the file that you want to upload and click "open".

Step 4: Navigate back to the "File Upload" browser window, and click "OK". The file uploading starts.

Step 3: After the file is uploaded completely, refresh the page.

4
Application Development

Application development on the UC-7400-CE computer is catered to by a number of well-known tools that are provided by the Windows® environment in programmers' workstations. These tools are trouble-free to use for Windows® programmers. This chapter presents the development tools for Windows® Embedded Application Development.

The following topics are covered in this chapter:

- **Installation of Development Tools (IDEs)**
  - VB.NET/C# Applications: Use Visual Studio 2005
  - VB.NET/C# Applications: Use Visual Studio .NET 2003
  - C/C++ Applications: Use eMbedded Visual C++ (eVC) 4.0
- **Steps to Install MOXA Windows® CE 5.0 C/C++ SDKs**
- **Steps to Develp Applications with IDEs**
  - To develop an eMbedded Visual C++ 4.0 application with an imported SDK
  - To develop a VB.NET / C# application with .NET Compact Framework

Installation of Development Tools (IDEs)

Building applications for the UC-7400-CE computer can be done with a number of friendly Integrated Development Environments (IDE) tools. Choose the tools based on the application language you plan to use and install them onto your development workstation.

VB.NET/C# Applications: Use Visual Studio 2005

Microsoft® Visual Studio 2005 is a complete set of development tools for building ASP.NET Web applications, XML Web services, desktop applications, and mobile applications. Visual Basic, Visual C++, Visual C#, and Visual J# all use the same IDE, which allows them to share tools and facilitates the creation of mixed-language solutions.

VB.NET/C# Applications: Use Visual Studio .NET 2003

Visual Studio .NET 2003 is another IDE you can use to develop your applications. Depending on the language you use, install the proper tools in the following order:

- Install Visual Studio .NET 2003
- Install Windows® CE Utilities for Visual Studio .NET 2003 Add-on Pack (550KB) for VB .NET
- Import Compact .Net Framework SDK

In addition, there are more development tools available from Microsoft® Web site. Find one for your suitable use.

C/C++ Applications: Use eMbedded Visual C++ (eVC) 4.0

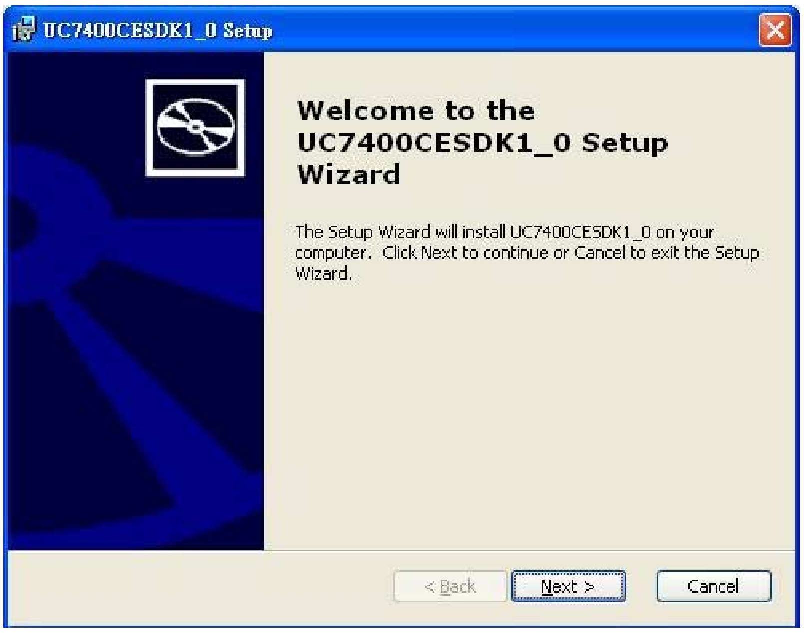
The eVC 4.0 tools are free. You can download them from MSDN's eMbedded Visual Tools Download Page. Install the eVC 4.0 tools, and then install the service pack in the following order:

- Install eMbedded Visual C++ 4.0 (230 MB)
- Install Service Pack 4 for eVC 4.0 (68 MB)
- Install MOXA Windows® CE 5.0 C/C++ SDKs

Steps to Install MOXA Windows® CE 5.0 C/C++ SDKs

After installing eVC 4.0 on the development workstation, you need to install a SDK from MOXA. The SDK file, e.g., ***UC7400CESDK1_0.msi,*** is available on the package CD.

❑ Double click on the msi file. A Setup Wizard pops up to start the installation process. Choose **Next.**



Welcome to the UC7400CESDK1_0 Setup Wizard

The Setup Wizard will install UC7400CESDK1_0 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

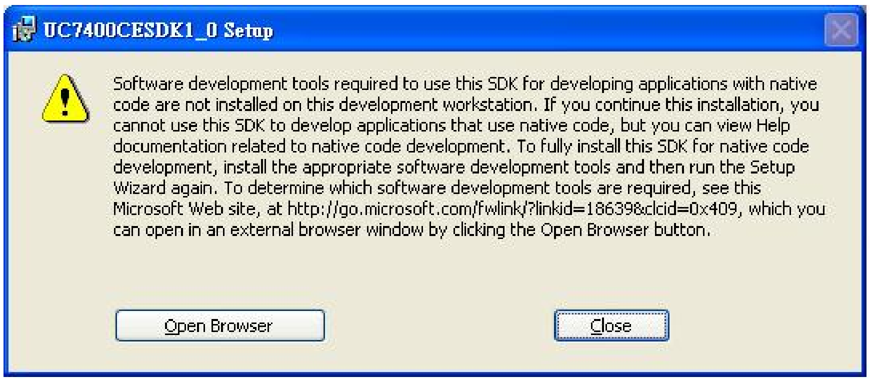
❑ If an error message appears, the appropriate versions of the application development tools required by the SDK are not installed. Choose Close to continue the installation process, even if you do not need these tools. Otherwise, cancel the process and download appropriate files from the Microsoft® site. For example, if a dialog like the following pops up, download two files from:

Microsoft® .NET Framework Version 2.0 Redistributable Package (x86) (25 MB)

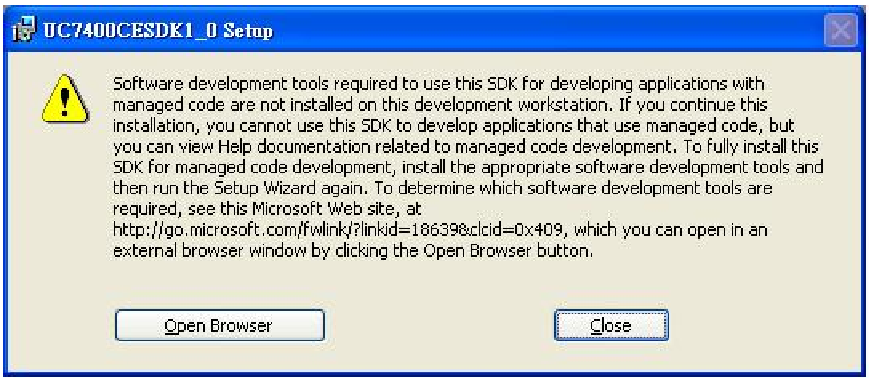
http://www.microsoft.com/downloads/details.aspx?FamilyID=0856EACB-4362-4B0D-8EDD-AAB15C5E04F5&displaylang=en

NET Framework 2.0 Software Development Kit (SDK) (x86) (354 MB)

http://www.microsoft.com/downloads/details.aspx?FamilyID=FE6F2099-B7B4-4F47-A244-C96D69C35DEC&displaylang=en



Software development tools required to use this SDK for developing applications with native code are not installed on this development workstation. If you continue this installation, you cannot use this SDK to develop applications that use native code, but you can view Help documentation related to native code development. To fully install this SDK for native code development, install the appropriate software development tools and then run the Setup Wizard again. To determine which software development tools are required, see this Microsoft Web site, at http://go.microsoft.com/fwlink/?linkid=18639&clcid=0x409, which you can open in an external browser window by clicking the Open Browser button.


UC7400CESDK1_0 Setup

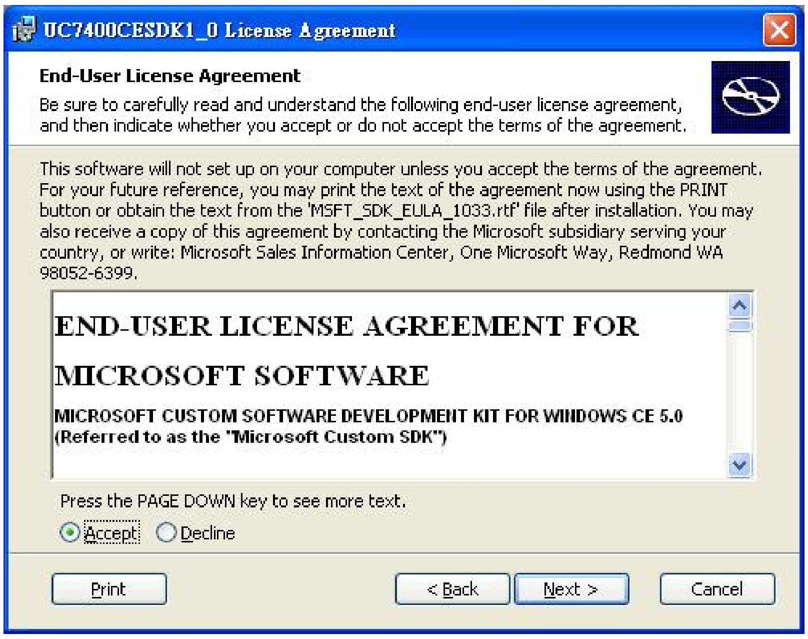
Software development tools required to use this SDK for developing applications with managed code are not installed on this development workstation. If you continue this installation, you cannot use this SDK to develop applications that use managed code, but you can view Help documentation related to managed code development. To fully install this SDK for managed code development, install the appropriate software development tools and then run the Setup Wizard again. To determine which software development tools are required, see this Microsoft Web site, at http://go.microsoft.com/fwlink/?linkid=18639&clcid=0x409, which you can open in an external browser window by clicking the Open Browser button.

Assuming that you have saved these two files as ***dotnetfx*** and ***setup***, do the following to install the tools:

- Execute the program "dotnetfx".
- Install .Net Framework by executing the program "setup".

Then, reinstall the MOXA SDK.

❑ Read the "License Agreement" dialog, and if you accept the terms of the end-user license agreement (EULA), choose **Accept**, and then click **Next**.


UC7400CESDK1_0 License Agreement

End-User License Agreement

Be sure to carefully read and understand the following end-user license agreement, and then indicate whether you accept or do not accept the terms of the agreement.

This software will not set up on your computer unless you accept the terms of the agreement. For your future reference, you may print the text of the agreement now using the PRINT button or obtain the text from the 'MSFT_SDK_EULA_1033.rtf' file after installation. You may also receive a copy of this agreement by contacting the Microsoft subsidiary serving your country, or write: Microsoft Sales Information Center, One Microsoft Way, Redmond WA 98052-6399.

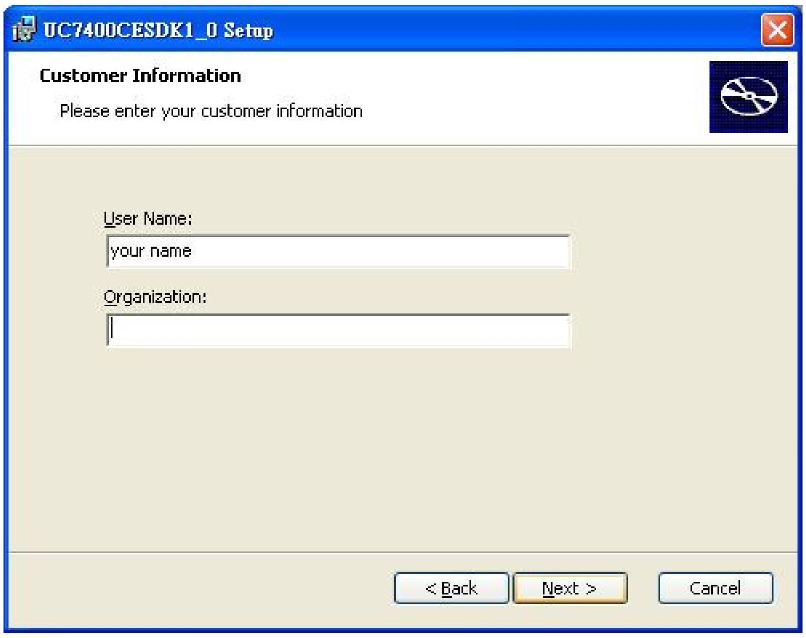
END-USER LICENSE AGREEMENT FOR MICROSOFT SOFTWARE

MICROSOFT CUSTOM SOFTWARE DEVELOPMENT KIT FOR WINDOWS CE 5.0 (Referred to as the "Microsoft Custom SDK")

Press the PAGE DOWN key to see more text.

[✓ Accept] Decline

- ❑ Type your name for **User Name**, and the name of your company for **Organization**. After that, click **Next**.



UC7400CESDK1_0 Setup

Customer Information

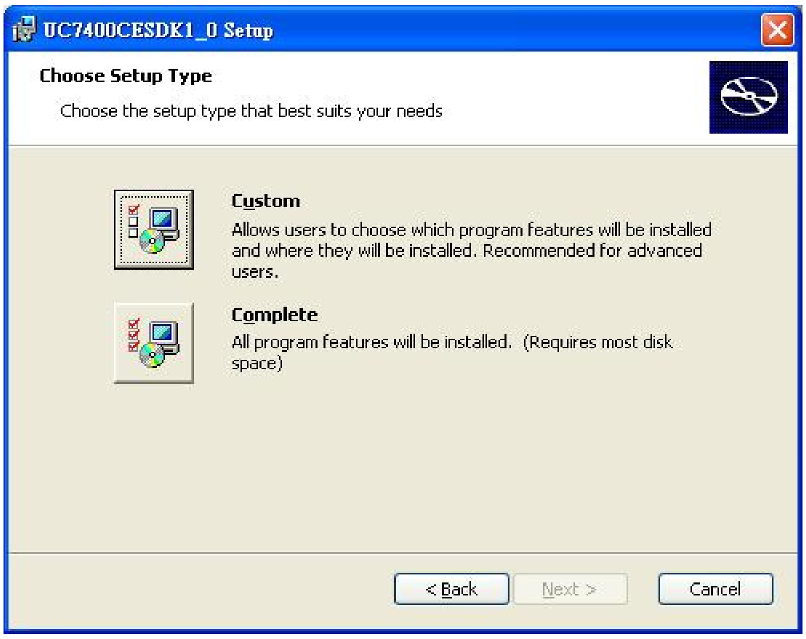
Please enter your customer information

User Name:

your name

Organization:

- ❑ To install all functionality provided with the SDK, click **Complete** and then click **Next**.



UC7400CESDK1_0 Setup

Choose Setup Type

Choose the setup type that best suits your needs

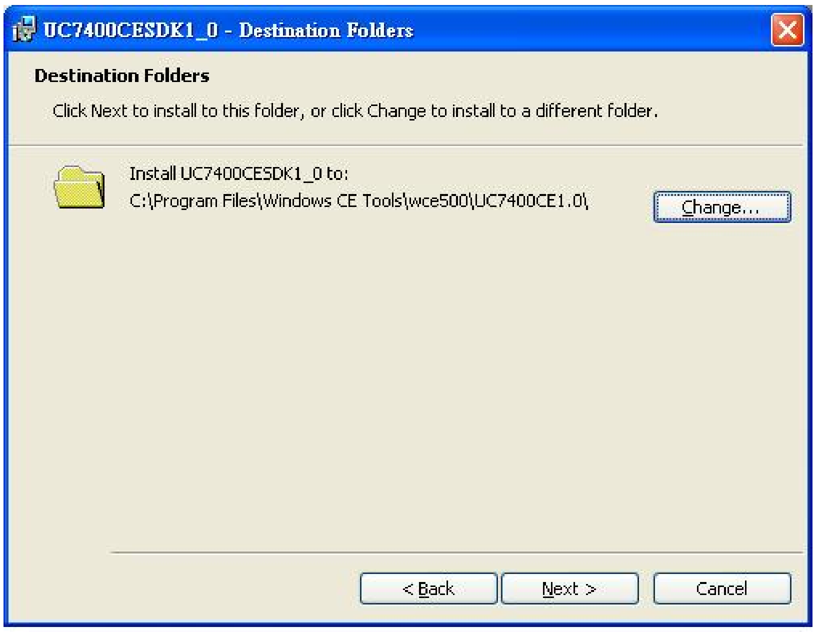
Custom

Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.

Complete

All program features will be installed. (Requires most disk space)

- ❑ If you have a folder preference where you want the SDK to reside, click **Change** and browse to the folder. Otherwise, place it in the default folder. After that, click **Next**.



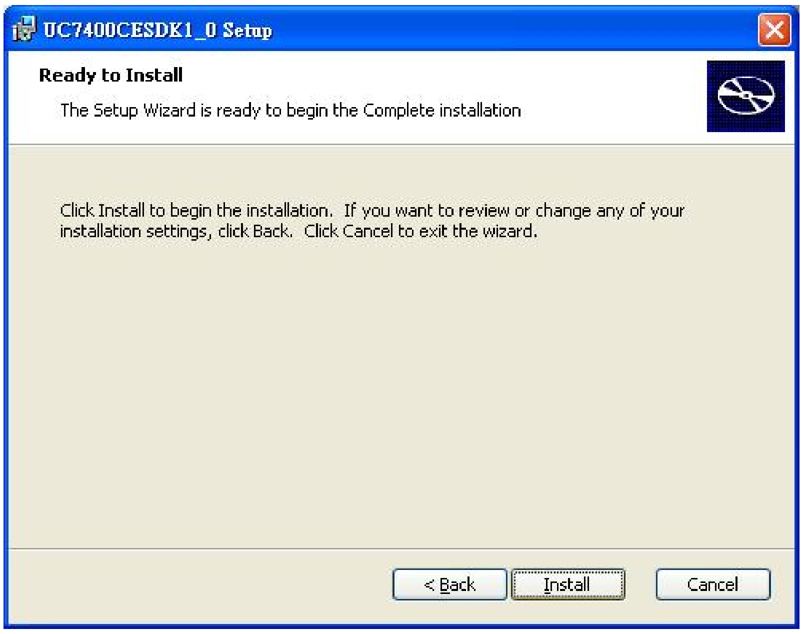
UC7400CESDK1_0 - Destination Folders

Destination Folders

Click Next to install to this folder, or click Change to install to a different folder.

Install UC7400CESDK1_0 to:
C:\Program Files\Windows CE Tools\wce500\UC7400CE1.0\

- ❑ To install the SDK, click **Install**.



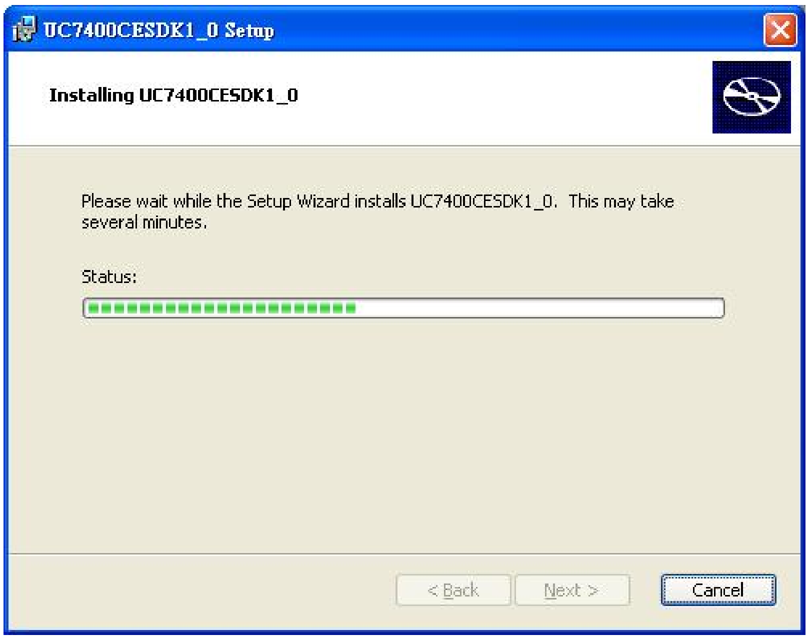
UC7400CESDK1_0 Setup

Ready to Install

The Setup Wizard is ready to begin the Complete installation

Click Install to begin the installation. If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

- ❑ Wait for the installation to be complete.

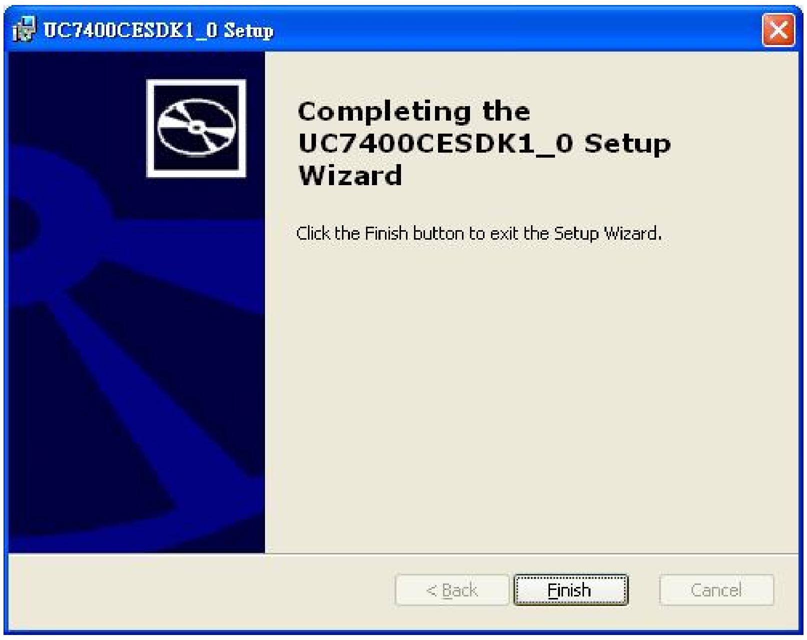

UC7400CESDK1_0 Setup

Installing UC7400CESDK1_0

Please wait while the Setup Wizard installs UC7400CESDK1_0. This may take several minutes.

Status:

- ❑ Once the installation is complete, click **Finish** to close the installer.


UC7400CESDK1_0 Setup

Completing the UC7400CESDK1_0 Setup Wizard

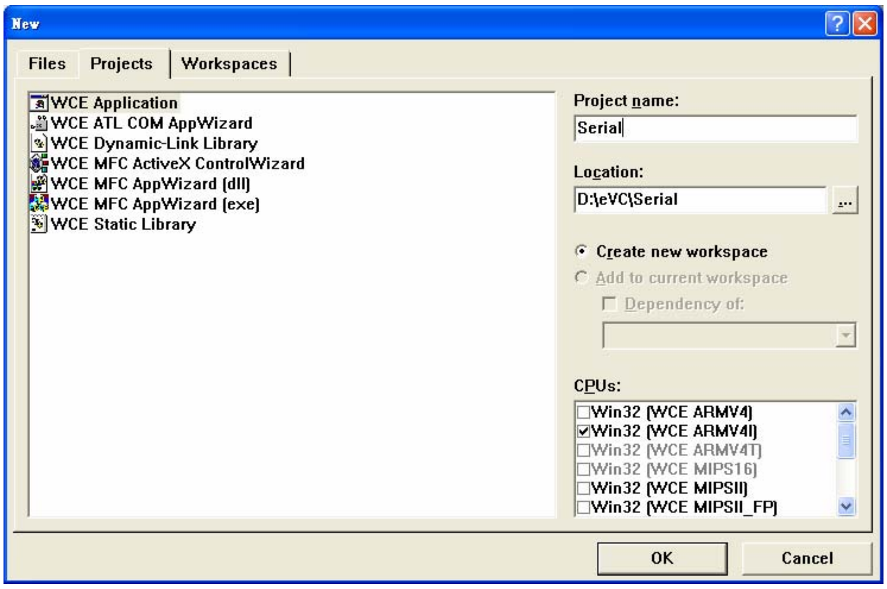
Click the Finish button to exit the Setup Wizard.

Steps to Develop Applications with IDEs

Windows® CE 5.0 with the MOXA SDK you have just installed offers application developers the familiar Windows® IDE and application programming interface (API) with the versatility of scripting languages. We use eVC 4.0 and Visual Studio .Net 2003 as examples to demonstrate the easy process of application development.

To develop an eMbedded Visual C++ 4.0 application with an imported SDK

- Open Microsoft® eMbedded Visual C++ 4.0.
- From the **File** menu, choose **New**.
- Choose the **Projects** tab and then select the type of application that you want to build.

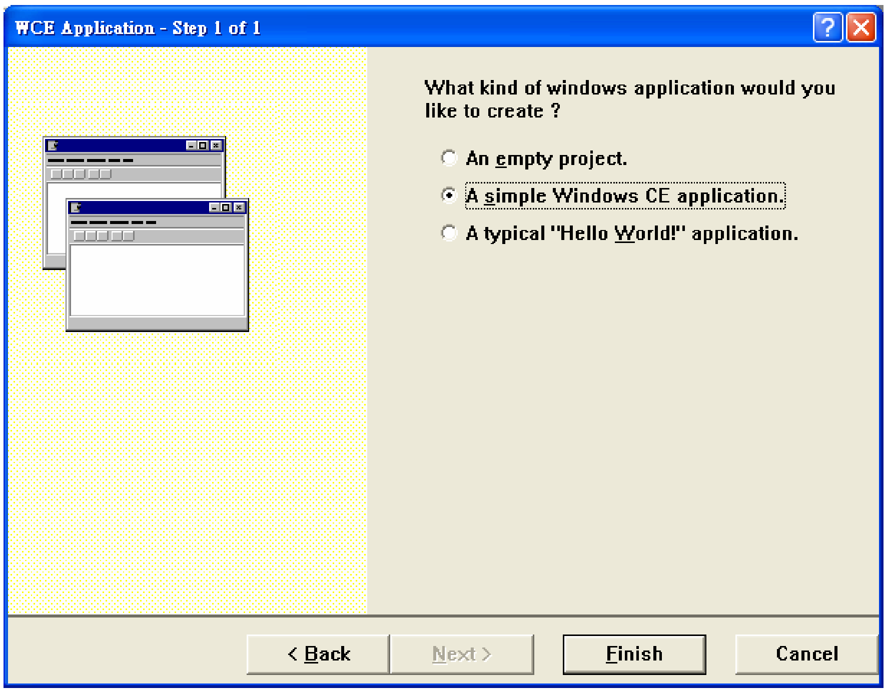


Project name: Serial

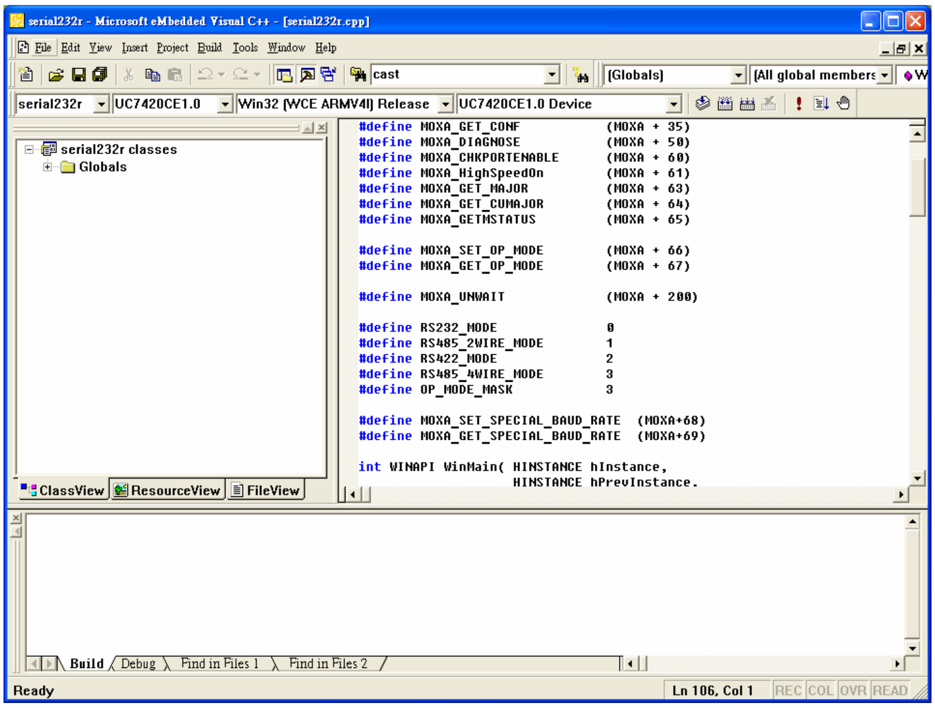
Location: D:\eVC\Serial

CPUs:
- ☐ Win32 (WCE ARMV4)
- ☑ Win32 (WCE ARMV4I)
- ☐ Win32 (WCE ARMV4T)
- ☐ Win32 (WCE MIPS16)
- ☐ Win32 (WCE MIPSII)
- ☐ Win32 (WCE MIPSII_FP)

- Fill in the project name and click **OK**.
- Choose the type of application that you want to create and then click **Finish**.

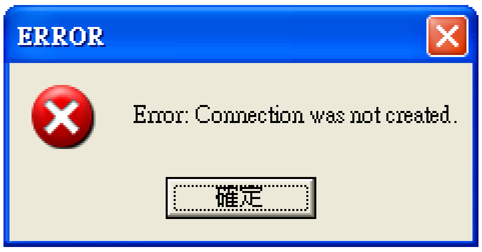

WCE Application - Step 1 of 1
What kind of windows application would you like to create ?
An empty project.
A simple Windows CE application.
A typical "Hello World!" application.
< Back
Next >
Finish
Cancel

- From the **Build** toolbar, choose the SDK (UC7400CE1.0 in this example), the type of run-time image (Release or Debug), and the target device (UC7400CE1.0 Device in the example).
- Write your code.



```
#define MOXA_GET_CONF           (MOXA + 35)
#define MOXA_DIAGNOSE           (MOXA + 50)
#define MOXA_CHKPORTENABLE      (MOXA + 60)
#define MOXA_HighSpeedOn        (MOXA + 61)
#define MOXA_GET_MAJOR          (MOXA + 63)
#define MOXA_GET_CUMAJOR        (MOXA + 64)
#define MOXA_GETMSTATUS         (MOXA + 65)

#define MOXA_SET_OP_MODE        (MOXA + 66)
#define MOXA_GET_OP_MODE        (MOXA + 67)

#define MOXA_UNWAIT             (MOXA + 200)

#define RS232_MODE              0
#define RS485_2WIRE_MODE        1
#define RS422_MODE              2
#define RS485_4WIRE_MODE        3
#define OP_MODE_MASK            3

#define MOXA_SET_SPECIAL_BAUD_RATE  (MOXA+68)
#define MOXA_GET_SPECIAL_BAUD_RATE  (MOXA+69)

int WINAPI WinMain( HINSTANCE hInstance,
                    HINSTANCE hPrevInstance.
```

- From the **Build** menu, choose **Rebuild All** to compile your application.
- Ignore the following dialog if it pops up. This is generated by ActiveSync while developing a mobile application. The UC-7400-CE computer is not a mobile device.



Error: Connection was not created.

- When you complete your application, use the web-based management system to upload it to the target computer.
- Logon the target computer and execute the program at the console prompt.

To develop a VB.NET / C# application with .NET Compact Framework

- Open Microsoft® Visual Studio .Net 2003.
- From the **File** menu, choose **New → Project.**
- Choose the **Project Type** and then select the **Smart Device Application** as the type of project.
- Fill in the project name and click **OK**.
- Choose **Windows CE** as the target platform.
- Select the desired project type and click **OK**.

- Write your application code.
- From the **Device** toolbar, choose **Windows CE.Net Device**.
- From the **Build** menu, choose **Build Project or Rebuild Project**.
- When you complete your application, use the web-based management system to upload it to the target computer.
- Logon the target computer. At the console prompt, execute it directly if it is a C# one. Otherwise, if it is a VB script or java script, execute the via the program **wcescript.exe**.

5
Programming Examples

The following examples are designed to ease application development on the UC-7400-CE computer. Find and copy useful code segments to your C/C++ programs. In addition, to simplify application development with the eVC4.0 tool, we provide you with a device library (***mxdev.lib***). This library covers the APIs for the buzzer, LCM, function key, and digital I/O devices. To link the library with your eVC4.0 compilation environment, perform the following steps from your eVC4.0 environment:

- From the main tool bar, choose **Project → Settings**.
- From **Project Settings**, select the **Link** tab.
- Append "*mxdev.lib*" to the text field "Object/Library Modules:".
- Click **OK**

Before you compile your application, please make sure that the header file and the library file in the SDK directories are as shown. If any of them is not in the specified directory, find it on the package CD and copy it to the specified directory.

"C:\Program Files\Windows CE Tools\wce500\UC7420CE1.0\Include\Armv4i\moxa\devices.h"

"C:\Program Files\Windows CE Tools\wce500\UC7420CE1.0\lib\Armv4i\mxdev.lib"

There are 5 code samples in this chapter and cover:

- ❑ **Example #1—MOXA UART Supporting RS-232/422/485**
- ❑ **Example #2—Buzzer**
- ❑ **Example #3—Digital I/O**
- ❑ **Example #4—LCM Display**
- ❑ **Example #5—Function Keys**

Example #1—MOXA UART Supporting RS-232/422/485

The following C/C++ code shows a sample application transmitting data from port "COM3" to port "COM4" using the RS-232 operation mode. After these ports have been opened, the application generates a thread to receive data from port "COM3" and then the application itself executes as a main thread to transmit data to port "COM4".

```
#include "stdafx.h"
#include <stdio.h>
#include <moxa/devices.h>

#define MAX_DATA_LEN 128

//========================================================================
static HANDLE createComHandle(WCHAR  *comPort, unsigned int baudrate)
{
  HANDLE hCom;
  DCB dcb;
  COMMTIMEOUTS         to;

  hCom = CreateFile(comPort,  GENERIC_READ | GENERIC_WRITE,
           0, // exclusive access, until the handle is closed
           NULL, // no security
           OPEN_EXISTING,
           0, // no overlapped I/O
           NULL); // null template
  if (hCom==INVALID_HANDLE_VALUE )
           return NULL;

  // set serial setting
  GetCommState(hCom, &dcb);
  dcb.BaudRate = baudrate;
  dcb.ByteSize = 8;
  dcb.StopBits = ONESTOPBIT;
  dcb.Parity = NOPARITY;
  dcb.fRtsControl = RTS_CONTROL_HANDSHAKE;
  dcb.fOutxCtsFlow = TRUE;
  SetCommState(hCom, &dcb);

  // set timeout parameter
  to.ReadIntervalTimeout = 0;
  to.ReadTotalTimeoutMultiplier = 0;
  to.ReadTotalTimeoutConstant = 0;
  to.WriteTotalTimeoutMultiplier = 0;
  to.WriteTotalTimeoutConstant = 0;
  if (!SetCommTimeouts(hCom,&to))
  {
           printf("SetCommTimeouts error!\n");
```

```
          CloseHandle(hCom);
          return NULL;
  }
  return hCom;
}

//=================================================================
static DWORD comReadThread(LPVOID param)
{
  HANDLE hCom = (HANDLE)param ;
  DWORD rtn;
  unsigned char buffer[MAX_DATA_LEN+1];
  while(1)
  {
          if (ReadFile(hCom, buffer, MAX_DATA_LEN, &rtn, NULL)==0)
          {
                  printf("read data fail\n");
                  return 0;
          }
          buffer[rtn] = '\0';
          printf("Data = %s\n", buffer);
  }
  CloseHandle(hCom);
}

int
comPair(WCHAR *wComPort, WCHAR *rComPort, unsigned int baudrate)
{
  HANDLE wCom, rCom;
  WCHAR sPort[64];
  DWORD rtn, i, loop=0;
  unsigned char buffer[MAX_DATA_LEN];
  HANDLE waitH = CreateEvent(NULL, FALSE, FALSE, NULL);

  if (waitH==NULL)
          return 99;

  /* create a handle to port "COM3" for transmitting data */
         wsprintf(sPort, L"$device\\%s\0", wComPort);
  wCom = createComHandle(sPort, baudrate);
  if (wCom==NULL)
  {
          printf("Fail to create write port\n" );
          return 1;
  }

         /* create a handle to port "COM3" for receiving data */
         wsprintf(sPort, L"$device\\%s\0", rComPort);
  rCom = createComHandle(sPort, baudrate);
  if (rCom == NULL)
  {
           printf("Fail to create read port\n");
           CloseHandle(wCom);
           return 2;
  }

  /* for a thread to handle receiving */
  if (CreateThread( NULL, 0, comReadThread, (LPDWORD) rCom, NULL, 0 )==NULL)
  {
           printf("Fail to create a receiving thread\n");
           CloseHandle(wCom);
           CloseHandle(rCom);
           return 3;
  }

  for (i=0; i< MAX_DATA_LEN;i++)
           buffer[i] = (unsigned char) ('a'+i%26);

  PurgeComm(wCom,PURGE_TXCLEAR | PURGE_TXABORT);
  while(loop++ < 100)
  {
       if (WriteFile(wCom, buffer, MAX_DATA_LEN, &rtn, NULL)==0)
           {
                    printf("Fail to write\n");
                    break;
           }
           WaitForSingleObject(waitH, 100);
  }
  CloseHandle(wCom);
  CloseHandle(rCom);
  CloseHandle(waitH);

  return 0;
}

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
LPTSTR lpCmdLine, int nCmdShow)
{
  comPair(L"COM3", L"COM4", 38400);
  return 0;
}
```

Windows® API function CreateFile(…) opens a named file corresponding to a serial port. For any serial port from "COM10" or after, the file name must be prefixed by "$device\\". For "COM1" to "COM9", the name can be either prefixed by "$device\\" or not. API function GetCommState(..) is the standard function to get the current parameters of a serial port, including its baudrate, data bits (bytesize),parity and stop bits. In contrast to GetCommState(), API function SetCommState(...) controls the serial communication by setting these parameters. Use other parameters including flow control, DTR signal, and RTS signal to fine tune the serial port. Be sure to set the flow control correctly or else you will lose data.

Depending on the device a serial port connects to, the UC-7400-CE computer supports serial communication in four operation modes: RS-232, RS-422, 2-wire RS-485, and 4-wire RS-485; RS-232 is the default. To change the operation mode, your program should include the following macro definitions; also insert the following DeviceIoControl function on the open handle before the program performs read/write operations.

```
/* --------- extracted from <moxa/devices> ------------------
#define MOXA_SET_OP_MODE     (0x400 + 66)
#define MOXA_GET_OP_MODE     (0x400+ 67)
#define RS232_MODE      0
#define RS485_2WIRE_MODE     1
#define RS422_MODE      2
#define RS485_4WIRE_MODE     3
*/

/* for example, set the mode to 4-wire RS485 */
BYTE mode = RS485_4WIRE_MODE;
DeviceIoControl(hCom,MOXA_SET_OP_MODE, &mode, sizeof(mode), NULL,0,NULL,NULL);
/* for example, get the current mode */
BYTE mode, size;
DeviceIoControl(hCom,MOXA_GET_OP_MODE, NULL, 0, (LPVOID)&mode,
sizeof(mode),&size,NULL);
```

Example #2—Buzzer

The UC-7400-CE computer supports buzzer hardware for your applications to generate alarms on critical errors. You can set the frequency and the duration of the buzzer at the application level by using the APIs as follows: (within "mxdev.lib"). For example, the codes shown trigger the buzzer for 2000 milliseconds at frequency 500Hzs.

```
/* execute a beeper at a specified frequency for lasting a duration in miniseconds */
```

int mxbeep(unsigned int nHz, unsigned int nMiniSec);

```
#include <moxa/devices.h>

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
           LPTSTR   lpCmdLine, int      nCmdShow)
{
    mxbeep(500,2000);
return 0;
}
```

Example #3—Digital I/O

The UC-7408-CE computer supports 8 programmable digital input channels and 8 programmable digital output channels for your applications. These IO channels can be accessed at run-time via the following APIs.

```
/* this function initialize a connection to the devices. It must be called before you
call other functions
  On success, it returns a handle. Otherwise, it return NULL
*/
```

HANDLE mxdio_init(void);

```
/* this function requests the system to detect a low-to-high or high-to-low event on
a specified input port every tick_slot milliseconds. On success, it returns 0.
    <hndl> handle created by mxdio_init
    <port> port number 0~7
    <handler> callback function with prototype (unsigned int, unsigned int, unsigned
    int)
    <low_high> macro DIN_EVENT_LOW_TO_HIGH or DIN_EVENT_HIGH _TO _LOW
    <tick_slot> 10 milliseconds minimal
*/
```

```
int mxdio_set_input_event( HANDLE hndl,
                           unsigned int port,
                           mxdio_input_cb handler,
                           unsigned int low_high,
                           unsigned int tick_slot);
```

```
/* this function starts a dispatcher to handle events.
    <hndl> handle created by mxdio_init
    Return non-zero to abort.
*/
```

int   mxdio_dispatch(HANDLE hndl);

```
/* this function set a digit level for a specified output port
    <hndl> handle created by mxdio_init
    <port> port number 0~7
    <data> 0 or 1
*/
```

int mxdio_set_dout(HANDLE hndl, unsigned int port, unsigned int data);

```
/* this function gets the current digit level of a specified port
    <hndl> handle created by mxdio_init
    <port> port number 0~7
    Return DIN_EVENT_LOW_TO_HIGH or DIN_EVENT_HIGH _TO _LOW  (0 or 1)
*/
```

int mxdio_get_dout(HANDLE hndl, unsigned int port);

To use the library ***mxdev.lib***, include file ***moxa/devices.h*** in your program. The following example shows a simple mechanism to use the library.

```
#include <moxa/devices.h>

DIOHANDLER(port0_process)
{
printf("port_0_process = %d\t%d\t%ld\n", port,type,syst_tick);
}

DIOHANDLER(port1_process)
{
printf("port_1_process = %d\t%d\t%ld\n", port,type,syst_tick);
}

int WINAPI WinMain( HINSTANCE  hInstance,
                    HINSTANCE  hPrevInstance,
                    LPTSTR  lpCmdLine,
                    int  nCmdShow)
{
    HANDLE h = mxdio_init();
    if(h==NULL)
    {
              printf("fail to init\n");
              return 1;
    }

    mxdio_set_input_event(h, 0, port0_process, DIN_EVENT_LOW_TO_HIGH, 30);
    mxdio_set_input_event(h, 1, port1_process, DIN_EVENT_HIGH_TO_LOW, 20);
    mxdio_dispatch(h);
    return 0;
}
```

All the callback functions have the same format ***(unsigned int port, unsigned int type, unsigned sys_tick)***. Name each of the functions and fill out the logic of your application inside it.

In the example shown above, based on the duration periods (20 and 30 milliseconds) provided by the callers, the dispatcher checks digit level changes at 20, 30, 40, 60, 80, …milliseconds from the moment that the dispatcher function ***mxdio_dispatch()*** is called. If there is any change, the associated callback function is initiated.

Example #4—LCM Display

Every UC-7420-CE and UC-7410-CE computer is equipped with LCM hardware with a display area of 8x16. This device offers a simple eye-catching interface for your applications. An internal service is pre-installed to manage a set of English text fonts for your applications. As an example, the following program transmits text to the service.

```
#include <windows.h>
#include <moxa/devices.h>

int WINAPI WinMain(  HINSTANCE hInstance,
                     HINSTANCE hPrevInstance,
                     LPTSTR    lpCmdLine,
                     int       nCmdShow)
{
    HANDLE hndl;
    char *buf="Hi, hello World!!\nabcdefg\n\n1234567890";
    unsigned int w, h, x, y;

    hndl = mxlcm_open(&w, &h);
    if (!hndl)
       return 1;

    x = 2; y=1;
mxlcm_clear(hndl);
    mxlcm_write(hndl, buf, strlen(buf), &x, &y);
    mxlcm_close(hndl);
    return 0;
}
```

Upon detecting the character '\n' or '\r' in the received text, the internal service forces a down shift of a line on the display screen. In addition, the service scrolls up a lengthy portion of text by dropping its leading part in order to fit the area of the LCM display.

```
/* open a handle to the LCM devie
       <lcm_w> the function outputs the width of the LCM display to this pointer
       <lcm_h> the function outputs the height of the LCM display to this pointer
       Return nonzero value in success.
*/
```

HANDLE mxlcm_open(unsigned int *lcm_w, unsigned int *lcm_h);

```
/* write text to the LCM device at a specify coordinate
       <hndl> the open handle
       <buffer> text
       <len> the length of the text
       <lcm_x> and <lcm,_y> pointers to the coordinate of the starting position of the
       text,
       the function would output the ending coordinate of the text to these pointers.
       This function returns the number of written bytes
*/
```

int mxlcm_write(HANDLE hndl, char *buffer, unsigned int len, unsigned int *lcm_x, unsigned int *lcm_y);

```
/* close the handle */
```

void mxlcm_close(HANDLE hndl);

```
/* clear the LCM screen */
```

void mxlcm_clear(HANDLE hndl);

Example #5—Function Keys

There are five programmable function keys available on the UC-7420-CE or UC-7410-CE computer for your applications. You can press a specified function key to trigger another application or switch your application among different operation modes.

An internal service is pre-installed and mimics key presses for your application. Your application needs to register a key press with the service. Once the key press is detected, the service notifies your application via a software interrupt. Your application can then operate on your logic. In the application level, library ***mxdev.lib*** offers the following APIs.

#define FKEYHNDL(f) int f(void* param)

```
/* initialization */
```

int mxfkey_init(void);

```
/* stop */
```

void mxfkey_stop(void);

```
/* register an operation to a keypad
    <key> a pointer to a string "F1"~"F5"
    <cb> a callback function to be trigger when the associated keypad is pressed
    <param> parameters that would be carried to the callback function
    Return zero in success.
*/
```

int mxkey_register(char *key, FKEYHNDL ((*cb)), void *param);

All you need to do is register your callback function to an associated keypad with the function in the format ***(unsigned int key, void* param)***. The argument ***param*** is used to transfer your private data to the callback function.

```
#include <windows.h>
#include <moxa/devices.h>

   FKEYHNDL(keyF2)
   {
      printf("keyF2 is pressed\n");
      return 0;
   }

   FKEYHNDL(keyF3)
   {
      printf("keyF3 is pressed\n");
      return 0;
   }

   int WINAPI WinMain( HINSTANCE hInstance,
                       HINSTANCE hPrevInstance,
                       LPTSTR    lpCmdLine,
                       int       nCmdShow)
   {
      HANDLE hEvent;
      DWORD rtn;

      if (mxfkey_init())
         return 1;
      rtn = mxfkey_register("F2", keyF2, 0);
      if (rtn)
         return 1;
      rtn = mxfkey_register("F3", keyF3, 0);
      if (rtn)
      {
         mxfkey_stop();
         return 1;
      }
      hEvent = CreateEvent( NULL, FALSE, FALSE, NULL );
      WaitForSingleObject( hEvent,20000 );
      mxfkey_stop();
      return 0;
```

A

Operating Programmable Function Keys and LCM

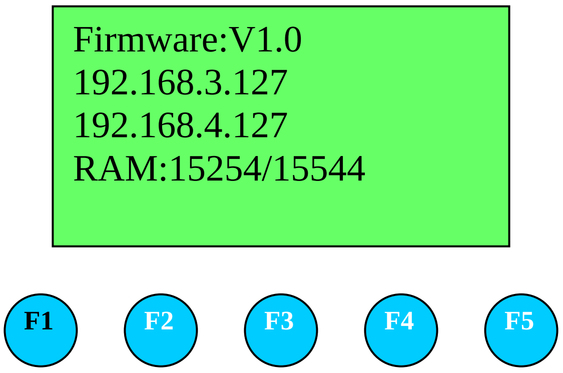
Notice: Only UC-7420 and UC-7410 support LCM and Keypad functions

In this appendix, we show you an internal service that is built around function keys. This mechanism is designed to aid your applications to access these I/Os without being aware of the complexity of I/O controls. Though this service gains controls on these programmable I/Os, it does not prevent you from developing on them. Instead, we provide this mechanism for you to easily satisfy your development needs.

This mechanism consists of a function key agent and a configuration utility. The agent operates based on five lists of pre-defined action items. Each list corresponding to a function key links a cyclic series of action items (10 at most) to be executed. The header item of this list maps directly to the first hit of the function key and then the next items follow. At the end of the list, the next hit circles the action back to the first item. A list without items comes with no action to be executed.

Each action item can be in plain text or an application that you develop. The former is normally used for reminder message such as a manual for operating the function keys. The latter can be an LCM application. Nevertheless, this application is required to use the LCM APIs in our library ***mxdev.lib*** to transmit the variable text to the LCM.

We have used this mechanism to implement an application on function key F1. When the key is pressed, the agent triggers this application to gather information and then displays the information on the LCM. The information includes the firmware version, network address settings, and current memory usage of the computer. These are particularly useful when you forget network information and have difficulties connecting to the computer. In addition, it provides an alarm system indicating the availability of your memory resource in the computer by preventing you from overusing your memory or losing data.

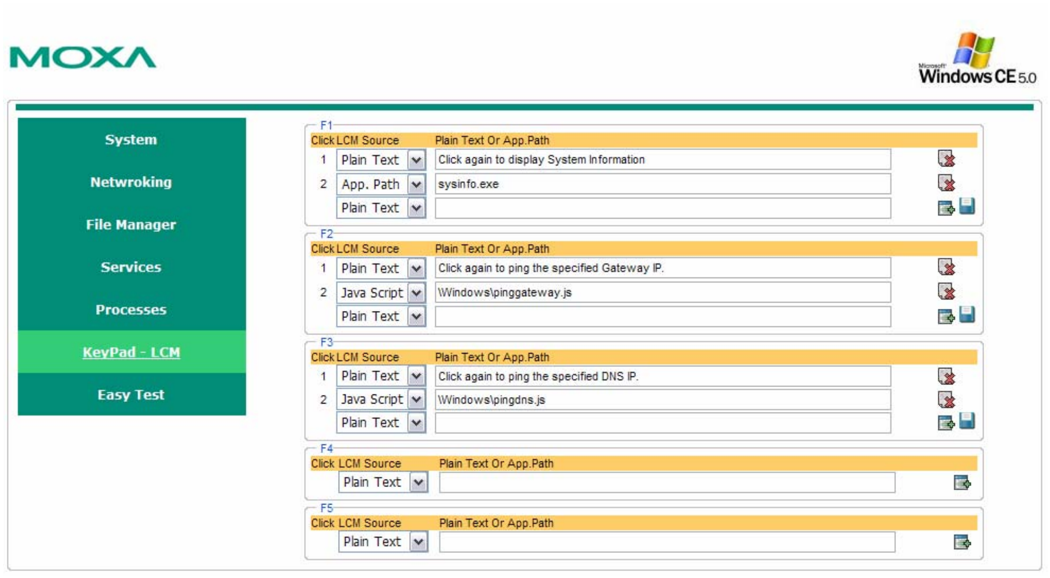


To assist you in using this function key service for I/Os, the web-based management system also implements a utility to configure action items. To configure a plain text item, please follow the steps:

Step 1: Logon the web-based management system and select the "keypad-LCM" in the function area. The browser opens a page which shows five current sets of action items.

Step 2: From the 5 sets of action items, choose one associated with a function key. Select the "Plain Text" option and then type the plain text into the text area.

Step 3: Save the item by clicking the disk icon.



F1

Click	LCM Source	Plain Text Or App.Path
1	Plain Text	Click again to display System Information
2	App. Path	sysinfo.exe
	Plain Text	

F2

Click	LCM Source	Plain Text Or App.Path
1	Plain Text	Click again to ping the specified Gateway IP.
2	Java Script	\Windows\pinggateway.js
	Plain Text	

F3

Click	LCM Source	Plain Text Or App.Path
1	Plain Text	Click again to ping the specified DNS IP.
2	Java Script	\Windows\pingdns.js
	Plain Text	

F4

Click	LCM Source	Plain Text Or App.Path
	Plain Text	

F5

Click	LCM Source	Plain Text Or App.Path
	Plain Text	

To configure an application for a function key, follow the steps:

Step 1: On your development PC, develop the application and make sure that it can output a multi-line text to the LCM display.

Step 2: Use the web-based management system to upload the application to the target computer.

Step 3: Select the "keypad-LCM" on the menu bar. The browser opens a page showing five current sets of function key action items.

Step 4: Choose one set associated with a function key. Select the "App-Path" option and type the full path of the application into the text area.

Step 5: Save the action item by clicking the disk icon.

To disable an action item, simply click the icon with the **X** next to the item you choose. Be sure to heed the length of a text message. The leading text of a lengthy one is ignored.

B

Firmware Upgrade Procedure

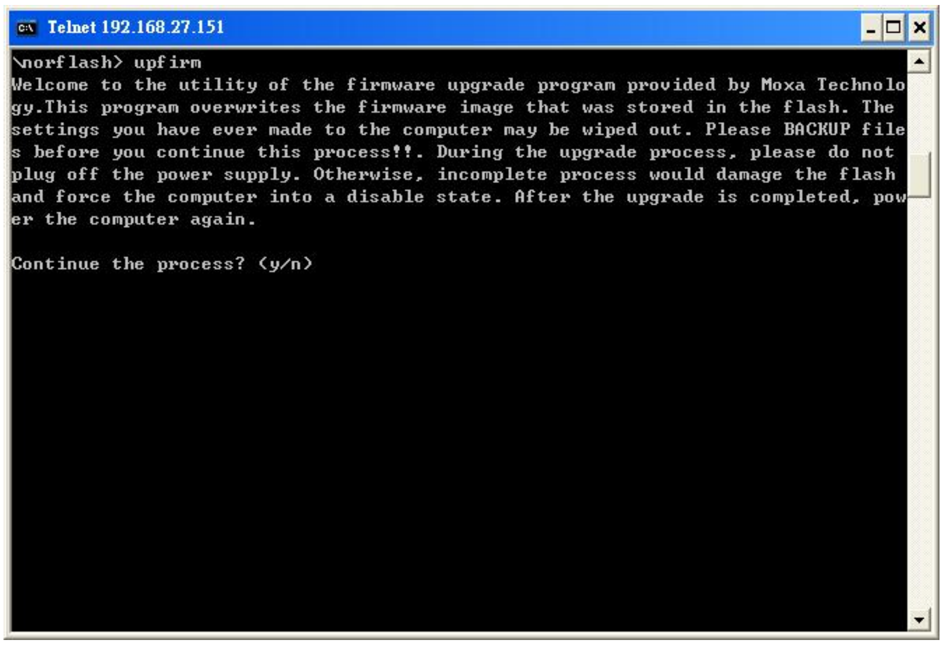
To upgrade the UC-7400-CE firmware, you can use the ***upfirm*** utility. This utility on the software CD; you can also download it from the MOXA Download Center, URL is listed below.

MOXA Download Center

http://web4.moxa.com/support/download_center.asp

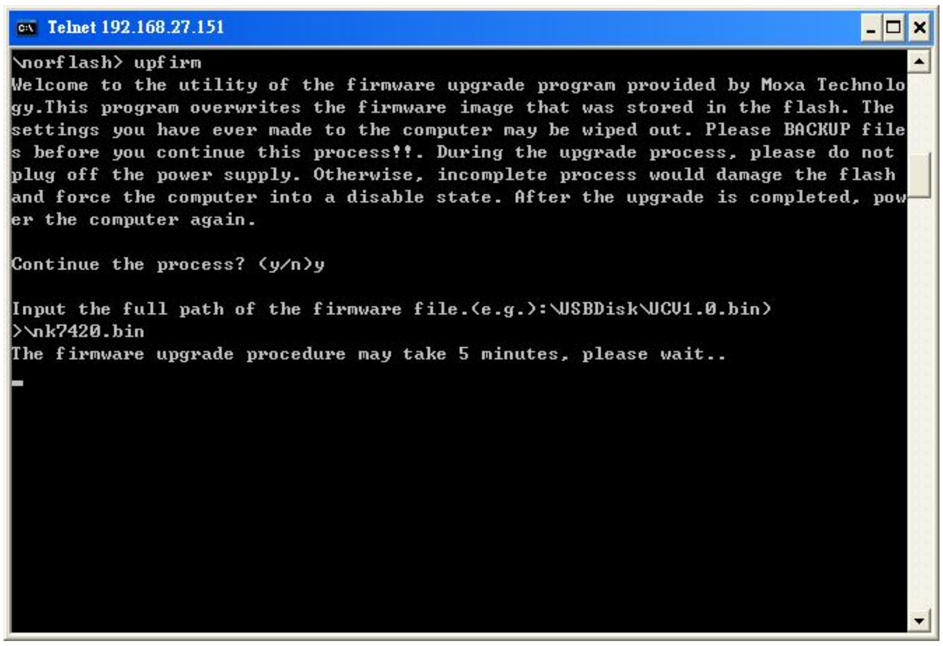
For firmware upgrade procedure, please refer to the following steps.

1. You can use FTP function or File Manager function of Web-based Management System described in Chapter 3 to download this upfirm utility onto the onboard Flash of UC-7400-CE
2. Enter the console of UC-7400-CE via “telnet” or “serial console”.
3. Execute the “upfirm” utility. Press “y” to continue the process.



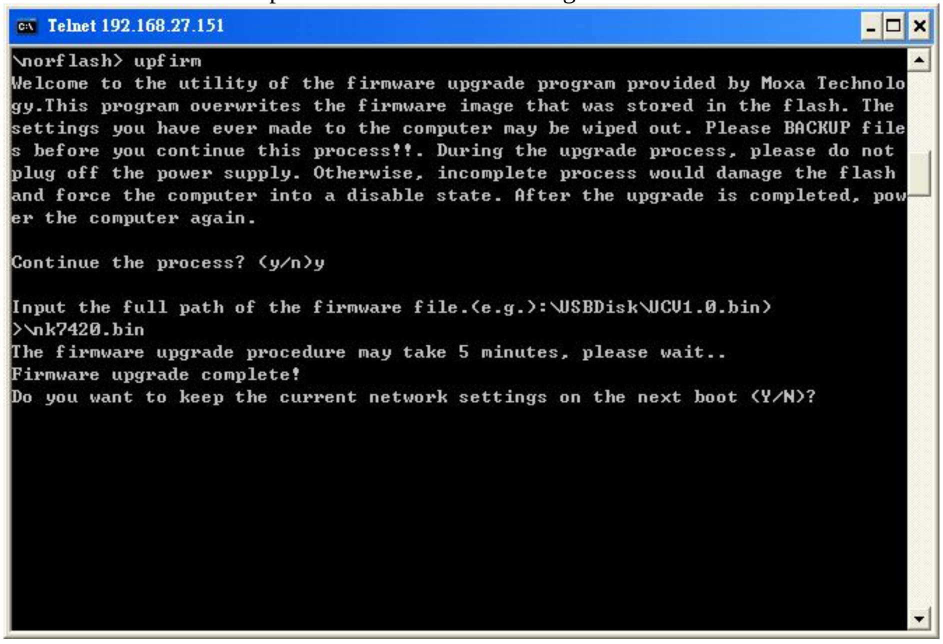
```
\norflash> upfirm
Welcome to the utility of the firmware upgrade program provided by Moxa Technolo
gy.This program overwrites the firmware image that was stored in the flash. The
settings you have ever made to the computer may be wiped out. Please BACKUP file
s before you continue this process!!. During the upgrade process, please do not
plug off the power supply. Otherwise, incomplete process would damage the flash
and force the computer into a disable state. After the upgrade is completed, pow
er the computer again.

Continue the process? (y/n)
```

4. Navigate to the path of the firmware file. The default "path\filename" is "\USBDisk\UCV1.0.bin". Make sure you select the correct target path and actual file name. For example, the new firmware file "nk7420.bin" is put under root path"\", enter the entire path\filename and press "ENTER" to begin upgrade the firmware.



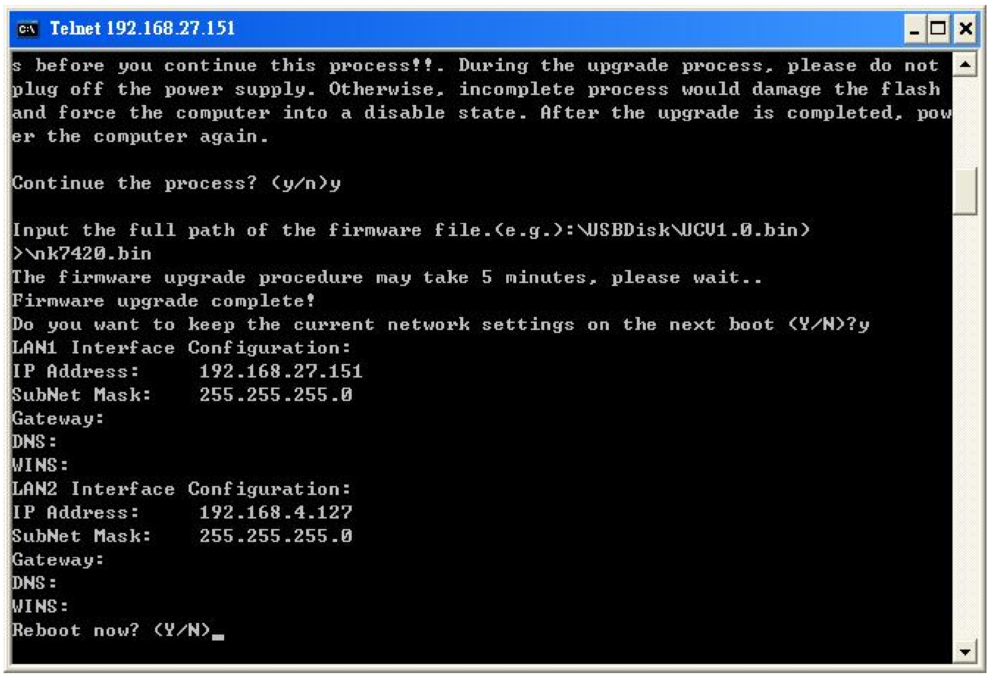
```
Telnet 192.168.27.151
\norflash> upfirm
Welcome to the utility of the firmware upgrade program provided by Moxa Technolo
gy.This program overwrites the firmware image that was stored in the flash. The
settings you have ever made to the computer may be wiped out. Please BACKUP file
s before you continue this process!!. During the upgrade process, please do not
plug off the power supply. Otherwise, incomplete process would damage the flash
and force the computer into a disable state. After the upgrade is completed, pow
er the computer again.

Continue the process? (y/n)y

Input the full path of the firmware file.(e.g.):\USBDisk\UCV1.0.bin)
>\nk7420.bin
The firmware upgrade procedure may take 5 minutes, please wait..
```

5. Wait for the upgrade to complete; it will take about 5 minutes to finish it. After upgrading, select if you would like to keep the current network settings or restore the factory default values. Press "Y" to keep the current network settings.



```
Telnet 192.168.27.151
\norflash> upfirm
Welcome to the utility of the firmware upgrade program provided by Moxa Technolo
gy.This program overwrites the firmware image that was stored in the flash. The
settings you have ever made to the computer may be wiped out. Please BACKUP file
s before you continue this process!!. During the upgrade process, please do not
plug off the power supply. Otherwise, incomplete process would damage the flash
and force the computer into a disable state. After the upgrade is completed, pow
er the computer again.

Continue the process? (y/n)y

Input the full path of the firmware file.(e.g.):\USBDisk\UCV1.0.bin)
>\nk7420.bin
The firmware upgrade procedure may take 5 minutes, please wait..
Firmware upgrade complete!
Do you want to keep the current network settings on the next boot (Y/N)?
```

6. Next, press "Y" to reboot the target UC-7400-CE.



```
s before you continue this process!!. During the upgrade process, please do not
plug off the power supply. Otherwise, incomplete process would damage the flash
and force the computer into a disable state. After the upgrade is completed, pow
er the computer again.

Continue the process? (y/n)y

Input the full path of the firmware file.(e.g.):\USBDisk\UCV1.0.bin)
>\nk7420.bin
The firmware upgrade procedure may take 5 minutes, please wait..
Firmware upgrade complete!
Do you want to keep the current network settings on the next boot (Y/N)?y
LAN1 Interface Configuration:
IP Address:     192.168.27.151
SubNet Mask:    255.255.255.0
Gateway:
DNS:
WINS:
LAN2 Interface Configuration:
IP Address:     192.168.4.127
SubNet Mask:    255.255.255.0
Gateway:
DNS:
WINS:
Reboot now? (Y/N)_
```

7. After system restart, you will find the new firmware ready and running.

C
Service Information

This appendix shows you how to contact MOXA for information about this and other products, and how to report problems.

The following topics are covered in this appendix.

MOXA Internet Services

Customer satisfaction is our primary concern. To ensure that customers receive the full benefit of our products, MOXA Internet Services has been set up to provide technical support, driver updates, product information, and user's manual updates.

The following services are provided

E-mail for technical support...…..............................support@moxa.com.tw

World Wide Web (WWW) Site for product information:

....http://www.moxa.com

Problem Report Form

MOXA UC-7400-CE Series

Customer name:	
Company:	
Tel:	**Fax:**
Email:	**Date:**

1. **Moxa Product:** ☐ UC-7420-CE ☐ UC-7410-CE ☐ UC-7408-CE
2. **Serial Number:** _________________

Problem Description: Please describe the symptoms of the problem as clearly as possible, including any error messages you see. A clearly written description of the problem will allow us to reproduce the symptoms, and expedite the repair of your product.

Product Return Procedure

For product repair, exchange, or refund, the customer must:

- ♦ Provide evidence of original purchase.
- ♦ Obtain a Product Return Agreement (PRA) from the sales representative or dealer.
- ♦ Fill out the Problem Report Form (PRF). Include as much detail as possible for a shorter product repair time.
- ♦ Carefully pack the product in an anti-static package, and send it, pre-paid, to the dealer. The PRA should be visible on the outside of the package, and include a description of the problem, along with the return address and telephone number of a technical contact.