DA-720 Series Windows Software User's Manual

Edition 2.0, March 2018

www.moxa.com/product

DA-720 Series Windows Software 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.

Copyright Notice

Trademarks

The MOXA logo is a registered trademark of Moxa Inc.
All other trademarks or registered marks in this manual belong to their respective manufacturers.

Disclaimer

Information in this document is subject to change without notice and does not represent a commitment on the part of Moxa.

Moxa provides this document as is, without warranty of any kind, either expressed or implied, including, but not limited to, its particular purpose. Moxa reserves the right to make improvements and/or changes to this manual, or to the products and/or the programs described in this manual, at any time.

Information provided in this manual is intended to be accurate and reliable. However, Moxa assumes no responsibility for its use, or for any infringements on the rights of third parties that may result from its use.

This product might include unintentional technical or typographical errors. Changes are periodically made to the information herein to correct such errors, and these changes are incorporated into new editions of the publication.

Technical Support Contact Information

www.moxa.com/support

Moxa Americas
Toll-free: 1-888-669-2872
Tel: +1-714-528-6777
Fax: +1-714-528-6778

Moxa Europe
Tel: +49-89-3 70 03 99-0
Fax: +49-89-3 70 03 99-99

Moxa India
Tel: +91-80-4172-9088
Fax: +91-80-4132-1045

Moxa China (Shanghai office)
Toll-free: 800-820-5036
Tel: +86-21-5258-9955
Fax: +86-21-5258-5505

Moxa Asia-Pacific
Tel: +886-2-8919-1230
Fax: +886-2-8919-1231

Table of Contents

1

Introduction

Thank you for buying Moxa's DA-720 panel computer. This model of the DA-720 comes with the Windows 10 Enterprise LTSB 2016 software platform, which provides a simple and familiar development environment for your various industrial applications needs.

The following topics are covered in this chapter:

❑ **Software Components**

Software Components

The Windows 10 Enterprise LTSB preinstalled on the DA-720 computer consists of the following:

Windows 10 Enterprise LTSB (by CTOS)

Core OS:

• 64-bit support

• Remote Client

• Remote Procedure Call

Applications and Services Development:

• .Net Framework 4.6

• Remote Desktop Protocol 10

• COM OLE Application Support

• COM+ Application Support

• MSMQ (message queuing)

• Work Folders Client

Internet Services:

• Internet Explorer 11

• IIS 10

File Systems and Data Storage:

• Windows Data Access Components

• Windows Backup and Restore

Diagnostics:

• Common Diagnostic Tools

• Problem Reports and Solutions

Graphic and Multimedia Tools:

• MPEG DTV-DVD Audio Decoder (MPEG-2, AAC)

• MPEG Layer-3 Audio Codecs (MP3)

• MPEG4 Decoders

• Windows Media Video VC-1 (WMV) Codecs

• DirectX and Windows Device Experience

• Create and play DVDs

• Photo Viewer

• Remote media streaming

• Windows Media Player

• Windows Mail

• Microsoft Print to PDF

• Internet Printing Client

• Windows Fax and Scan

• XPS Viewer

• XPS Services

Management:

• Group Policy Management

• Windows Management Instrument (WMI)

• Windows Update

Networking:
- Extensible Authentication Protocol (EAP)
- Internet Authentication Service
- Telnet Server
- Bluetooth
- Domain Services
- Network Access Protection
- Network and Sharing Center
- Quality of Service
- Remote Access Service (RAS)
- Telephony API Client
- Windows Firewall
- Wireless Networking

Security:
- Credential Roaming Service
- Credentials and Certificate Management
- Windows Authorization Manager (AzMan)
- Windows Security Center
- Active Directory Rights Management
- Security Base
- Encrypted File System (EFS)
- MS Antimalware
- Windows Defender

Embedded Features:
- Unified Write Filter (UWF)
- Message Box Default Reply
- Registry Filter
- The Microsoft Web Services on Devices API (WSDAPI) for .NET

Embedded Self-Health Diagnostics: SNMP-based remote scripting layer for monitoring, reporting, and control

2

System Initialization

This chapter covers the initial system settings on the DA-720 computer after you boot up the computer for the first time.

The following topics are covered in this chapter:

❏ **Overview**

    ➢ Initializing User Settings

Overview

Like most laptop computers, you must first create a user account and configure the user account settings.
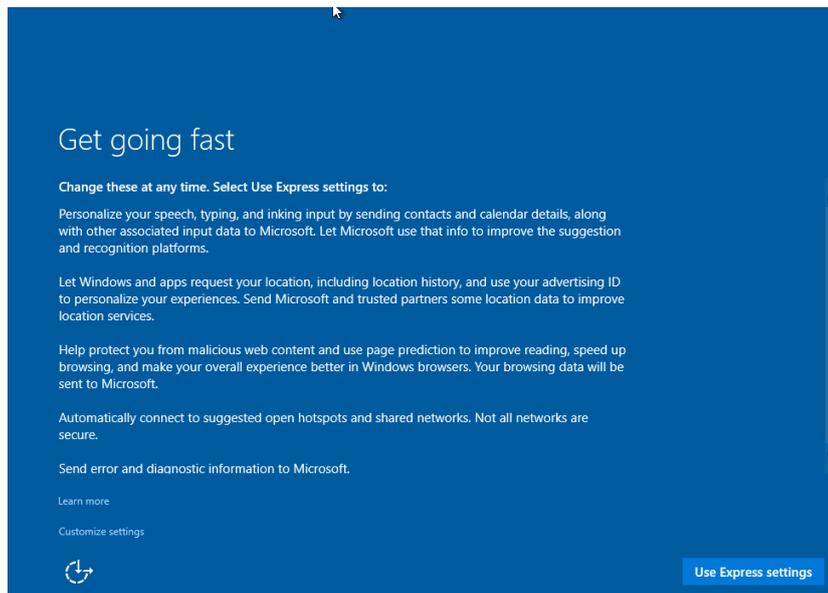
Initializing User Settings

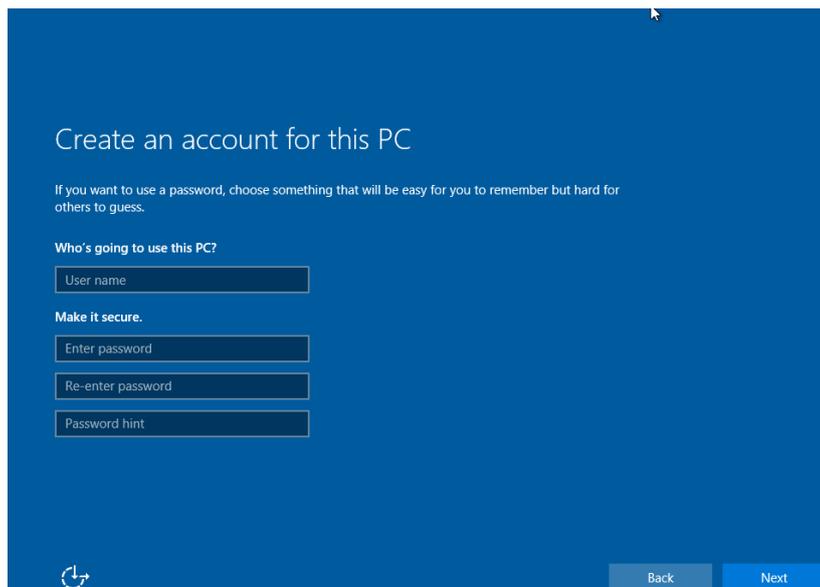1. When you boot the embedded computer for the first time, you will be asked the following four questions:

   (1) What's your home country/region?
   (2) What's your preferred app language?
   (3) What keyboard layout would you like to use?
   (4) What time zone are you in?

   Choose your answers and click **Next**.

2. Click **Use Express settings**.

   ## Get going fast

   Change these at any time. Select Use Express settings to:

   Personalize your speech, typing, and inking input by sending contacts and calendar details, along with other associated input data to Microsoft. Let Microsoft use that info to improve the suggestion and recognition platforms.

   Let Windows and apps request your location, including location history, and use your advertising ID to personalize your experiences. Send Microsoft and trusted partners some location data to improve location services.

   Help protect you from malicious web content and use page prediction to improve reading, speed up browsing, and make your overall experience better in Windows browsers. Your browsing data will be sent to Microsoft.

   Automatically connect to suggested open hotspots and shared networks. Not all networks are secure.

   Send error and diagnostic information to Microsoft.

   Learn more

   Customize settings

   Use Express settings

3. Provide User name, Password, and Password hint, and click **Next** to create a user account on the computer.

   ## Create an account for this PC

   If you want to use a password, choose something that will be easy for you to remember but hard for others to guess.

   Who's going to use this PC?

   User name

   Make it secure.

   Enter password

   Re-enter password

   Password hint

   Back   Next

You can start using the DA-720 embedded computer once the user account is created.

3

Configuring the Serial Interface

This chapter describes how to configure the serial interface on the DA-720 computer.

The following topics are covered in this chapter:

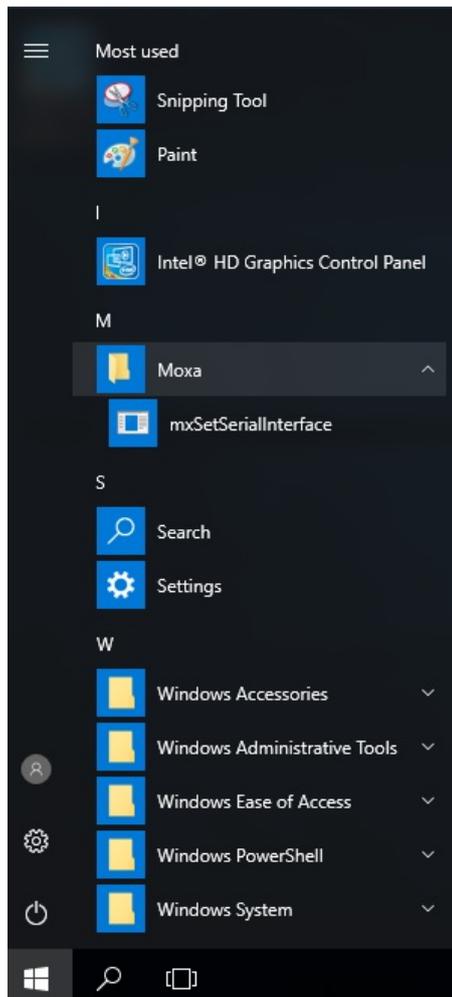□ **Overview**
□ **Configuring Serial Interface Mode**

Overview

The DA-720 supports three serial modes: **RS232**, **RS485-2-wire**, and **RS422/RS485-4-wire**. These modes can be configured either on COM1 or COM2 port.
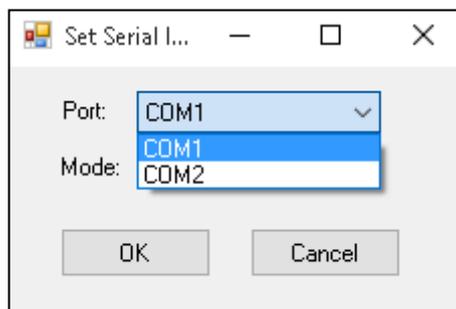
Configuring Serial Interface Mode

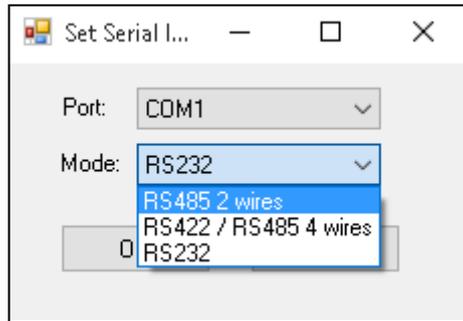To change the serial interface mode settings, do the following:

1. In the Start menu select **All apps → Moxa → mxSetSerialInterface**.

2. Select a **Port** for the serial interface.

3. Select the specific **Mode** for the serial interface.

4. Click **OK**.

4

Enabling Embedded Filters

This chapter describes how to operate the embedded features on the DA-720 computer.

The following topics are covered in this chapter:

❑ **Unified Write Filter**

➢ Overview

➢ Configuring File-Based Write Filter

Unified Write Filter

Overview

What is UWF?

The Unified Write Filter (UWF) is a feature to protect physical storage media from data writes. UWF intercepts all write attempts to a protected volume and redirects those write attempts to a virtual overlay. This improves the reliability and stability of your device and reduces the wear on write-sensitive media, such as flash memory media like solid-state drives.

About UWF Overlay

UWF intercepts all modifications to any sector on a protected volume. A sector is the smallest unit that can be changed on a storage volume. Any time the file system attempts to modify a protected sector, UWF instead copies the sector from the protected volume to the overlay, and then modifies the overlay instead. If an application attempts to read from that sector, UWF returns the data from the overlay instead, so that the system maintains the appearance of having written to the volume, while the volume remains unchanged.
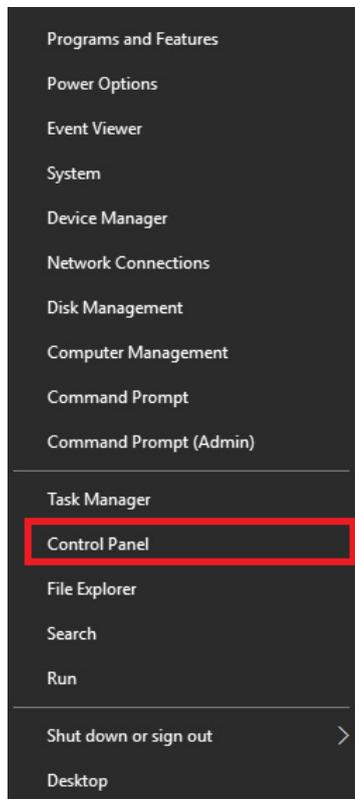
Source: www.microsoft.com

Configuring File-Based Write Filter
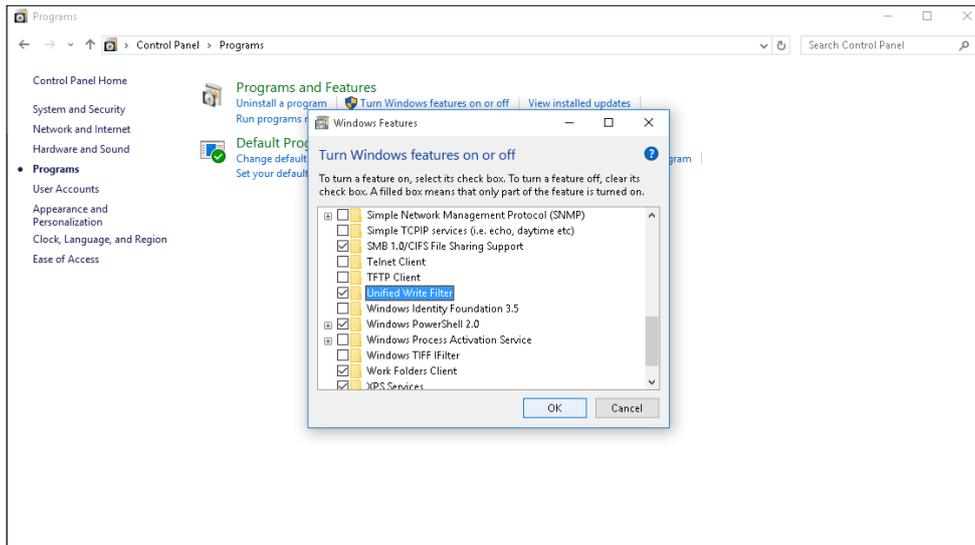
1) Enabling or Disabling the UWF Function

To enable the UWF function, do the following:

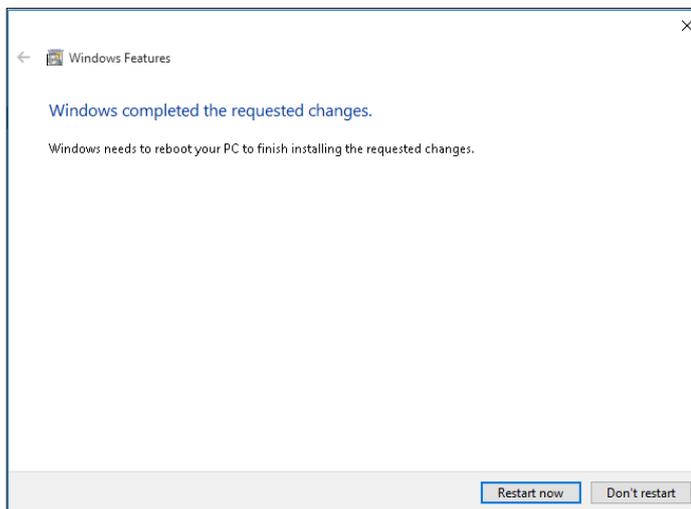1. Select **Control Panel** in the Windows Start menu.

2. In the left panel, click **Programs.**

3. Click on the **Turn Windows features on or off** link under **Programs and Features**, select **Unified Write Filter** from the list, and click **OK**.

4. Click **Restart now** to apply the changes.

2) Configuring the UWF Overlay Settings

To configure the UWF overlay settings, do the following:

1. Type **cmd** in the Windows Start menu field and press Enter to open a terminal.

2. Run the following command to protect the computer volume C:

   uwfmgr volume protect C:

3. Run the following command to enable UWF protection:

   uwfmgr filter enable

4. Run the following command to exclude files in the **C:\Program Files** folder from UWF protection:

    `uwfmgr file Add-Exclusion C:\Program Files`

```
Administrator: C:\Windows\System32\cmd.exe                                   —   □   ×

Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Windows\system32>uwfmgr file Add-Exclusion "C:\Program Files"
Unified Write Filter Configuration Utility version 10.0.10240
Copyright (C) Microsoft Corporation. All rights reserved.

The file/folder "C:\Program Files" will be excluded from protection after system restart.

C:\Windows\system32>uwfmgr overlay set-size 2048
Unified Write Filter Configuration Utility version 10.0.10240
Copyright (C) Microsoft Corporation. All rights reserved.
```
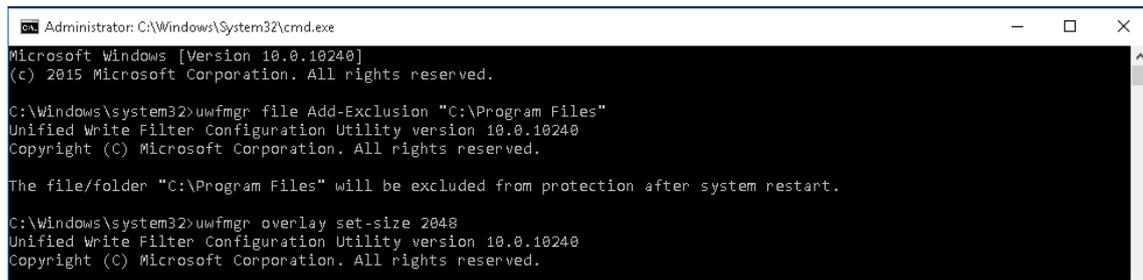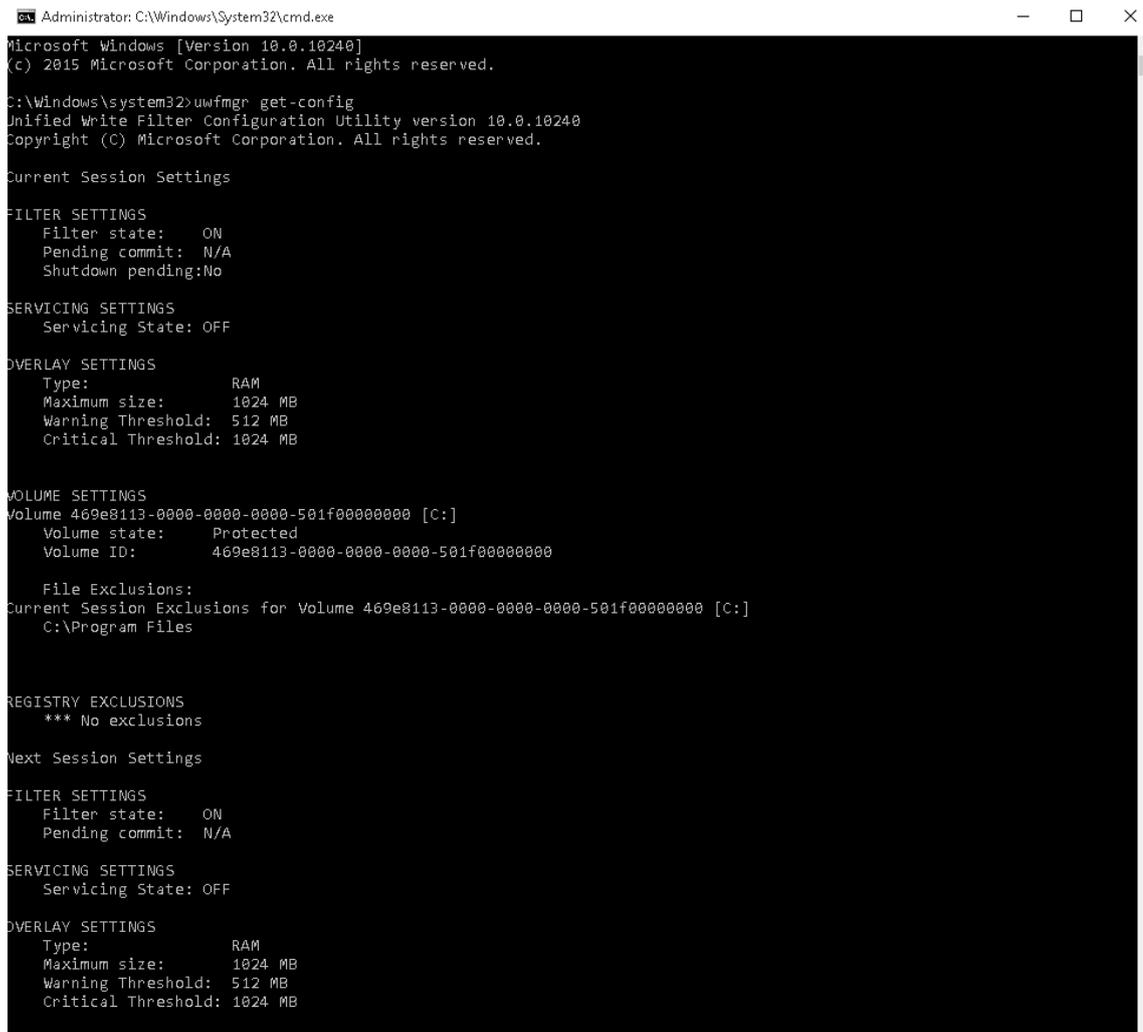
5. Exit from the terminal and restart your computer for the changes to take effect.

```
Administrator: C:\Windows\System32\cmd.exe                                   —   □   ×

Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Windows\system32>uwfmgr get-config
Unified Write Filter Configuration Utility version 10.0.10240
Copyright (C) Microsoft Corporation. All rights reserved.

Current Session Settings

FILTER SETTINGS
    Filter state:    ON
    Pending commit:  N/A
    Shutdown pending:No

SERVICING SETTINGS
    Servicing State: OFF

OVERLAY SETTINGS
    Type:            RAM
    Maximum size:    1024 MB
    Warning Threshold:  512 MB
    Critical Threshold: 1024 MB


VOLUME SETTINGS
Volume 469e8113-0000-0000-0000-501f00000000 [C:]
    Volume state:    Protected
    Volume ID:       469e8113-0000-0000-0000-501f00000000

    File Exclusions:
Current Session Exclusions for Volume 469e8113-0000-0000-0000-501f00000000 [C:]
    C:\Program Files



REGISTRY EXCLUSIONS
    *** No exclusions

Next Session Settings

FILTER SETTINGS
    Filter state:    ON
    Pending commit:  N/A

SERVICING SETTINGS
    Servicing State: OFF

OVERLAY SETTINGS
    Type:            RAM
    Maximum size:    1024 MB
    Warning Threshold:  512 MB
    Critical Threshold: 1024 MB
```

After restarting your computer, you can check the UWF status by running the **uwfmgr get-config** command in a terminal.

To test the UWF protection:

1. After you enable UWF protection, create files both in the C:\Program Files and the C:\ folders.
2. Restart the computer.

    Only the file that you created in the **C:\Program Files** should exist. The file that you created in the **C:** folder is erased.

    To disable the UWF protection, open a terminal and run the `uwfmgr filter disable` command**.**

5

Examples

This chapter describes how to use the different functions of the DA-720 with examples.

The following topics are covered in this chapter:

❒ **Watchdog Function**

➢ Enabling the Watchdog Function

❒ **LED Indicators**

➢ Displaying and Controlling the LED On/Off Status

❒ **Serial Interface**

➢ Displaying and Controlling the UART Mode

❒ **Relay Output**

➢ Displaying the Relay Status and Changing the Status to High or Low

Watchdog Function

An executable file, **watchdog.exe** that enables the watchdog function, is provided in the software DVD that ships with the computer.

Enabling the Watchdog Function

To enable the watchdog function on your computer using the **watchdog.exe** file, do the following:

1. Create **c:\programs\example** folder and copy the following files into the folder:

   mxdwg.dll: <Software DVD>\examples\DA720-W10-example\3.lib\mxwdg

   mxGeneralIo.dll: <Software DVD>\examples\DA720-W10-example\3.lib\MxGeneralIo

   Watchdog.exe: <Software DVD>\examples\DA720-W10-example\Release\x64\

2. Run **Watchdog.exe**.

   You must press **Enter** every 10 seconds to prevent the system from restarting.

   If you want to stop the watchdog function and exit the program, type **q**.
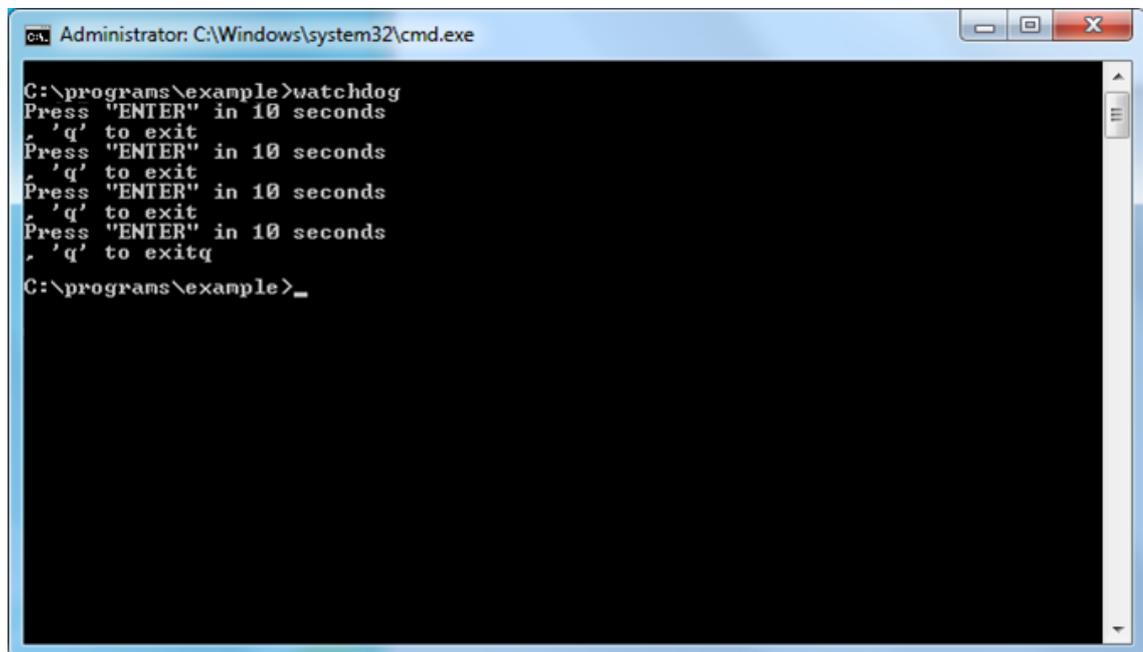
LED Indicators

An executable file, **LED.exe** that displays and controls the status of the LEDs, is provided in the software DVD that ships with the computer.

Displaying and Controlling the LED On/Off Status

To display the status of the LEDs and to switch the LEDs On or Off, do the following:

1. Copy the following files from the product software DVD:

   mxgpio.dll: <Software DVD>\examples\DA720-W10-example\3.lib\mxgpio\x64

   mxGeneralIo.dll: <Software DVD>\examples\DA720-W10-example\3.lib\MxGeneralIo

   LED.exe: <Software DVD>\examples\DA720-W10-example\Release\x64\

2. Run **LED.exe**.

3. Select **1** to get the value of the current LED.

NOTE	The LED port numbers 0 to 5 are used to represent the LEDs 1 to 6 on the computer's front panel.
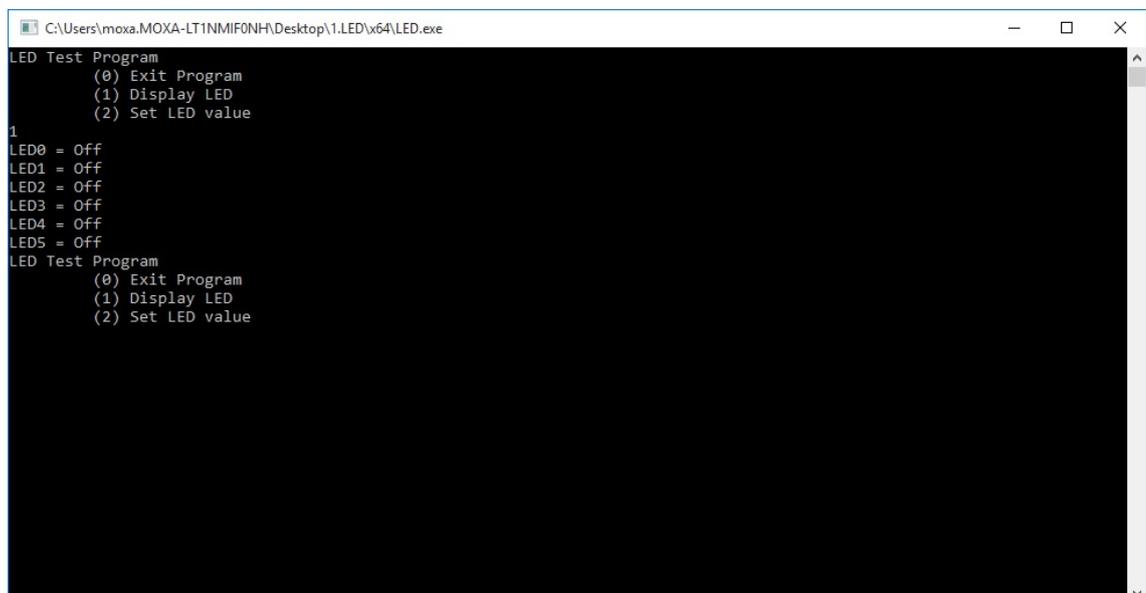
4. Select **2** to change the status (On, Off) of the current LED.
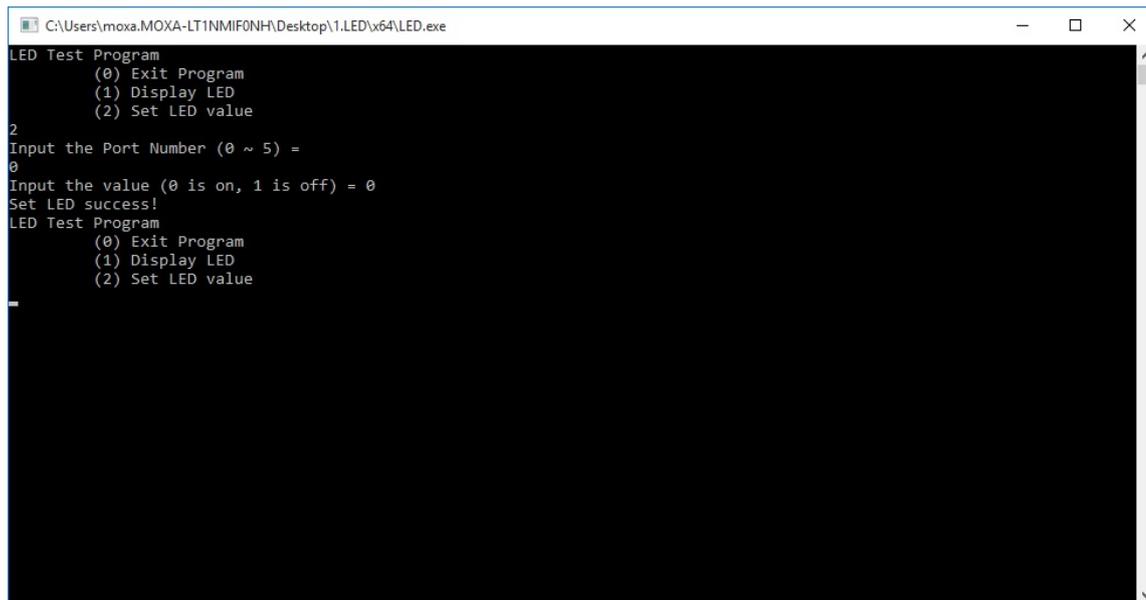
Serial Interface

An executable file, **UartMode.exe** that displays the status and controls the UART mode of the computer is provided in the software DVD that ships with the computer.

Displaying and Controlling the UART Mode

To display the status of the UART interface and to set the UART mode, do the following:

1. Copy the following files from the product software DVD:

   mxsp.dll, SysInfo.dll, SysInfo.sys, SysInfoX64.sys:
   <Software DVD>\examples\DA720-W10-example\3.lib\mxsp\x64
   mxGeneralIo.dll: <Software DVD>\examples\DA720-W10-example\3.lib\MxGeneralIo
   **UartMode.exe:** <Software DVD>\examples\DA720-W10-example\Release\x64\

2. Run **UartMode.exe**.

3. Type **2** to set the serial interface and follow the onscreen instructions.

4. Type **1** to display the current serial interface settings.

Relay Output

An executable file, **Relay.exe** that displays the relay status and helps you change its status is provided in the software DVD that ships with the computer.

Displaying the Relay Status and Changing the Status to High or Low

To display the relay status and to set the status to high or low, do the following:

1. Copy the following files from the product software DVD:

    mxgpio.dll: <Software DVD>\examples\DA720-W10-example\3.lib\mxgpio\x64
    mxGeneralIo.dll: <Software DVD>\examples\DA720-W10-example\3.lib\MxGeneralIo
    Relay.exe: <Software DVD>\examples\DA720-W10-example\Release\x64\

2. Run **Relay.exe**.

3. Type **2** to set the serial interface and follow the onscreen instructions.

4. Type **1** to display the current serial interface settings.

6

System Recovery

This chapter describes the Windows 10 Enterprise LTSB platform recovery process in the event of system instability.

The following topics are covered in this chapter:

❒ **Recovery Environment**
❒ **Recovery Procedure**
❒ **Saving the System Image to the USB Drive**

Recovery Environment

The recovery environment consists of the DA-720 panel computer and a bootable USB disk that contains the recovery programs and system image file.

The hardware used includes a PC, a DA-720 computer, and a USB disk with the recovery programs.

NOTE   The USB disk should have at least 8 GB of free space.

```
                                              ┌──────────────────────┐
                                              │       DA-720         │
                                              ├──────────────────────┤
┌──────────────────────────┐                  │                      │
│ USB Disk                 │──────────────────│      USB Ports       │
│ (Recovery data included) │                  │                      │
└──────────────────────────┘                  └──────────────────────┘
```

Recovery Procedure

Step 1: Prepare your USB drive

1. Run the **tuxboot-windows-23.exe** program from the *<Software DVD>***\recovery** folder, select the **Pre Downloaded** option, and then click on the **...** button as shown below:
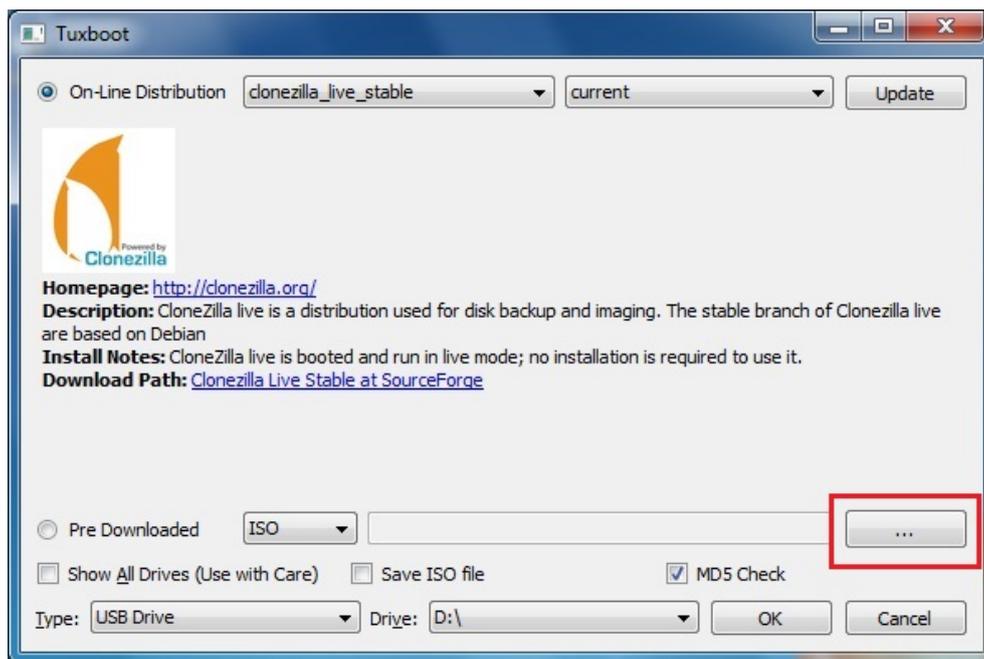
2. Browse to and select the CloneZilla ISO file from the *<Software DVD>***recovery** folder.
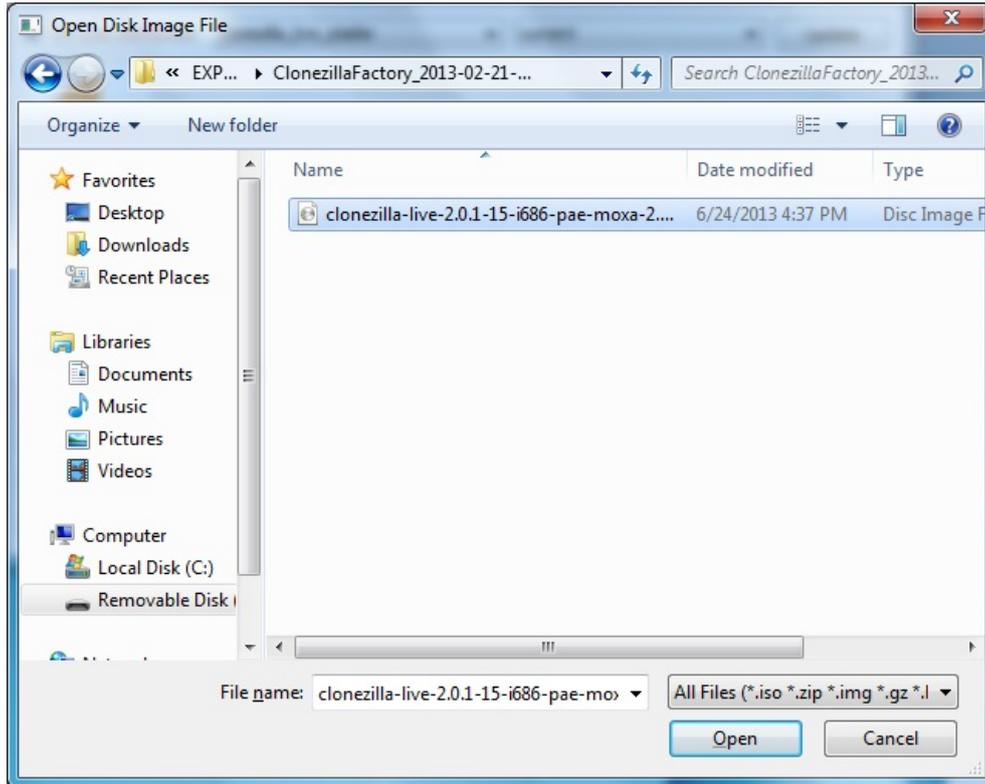
3. Select the **USB Drive** type and the **Drive**, and then click **OK** to continue.
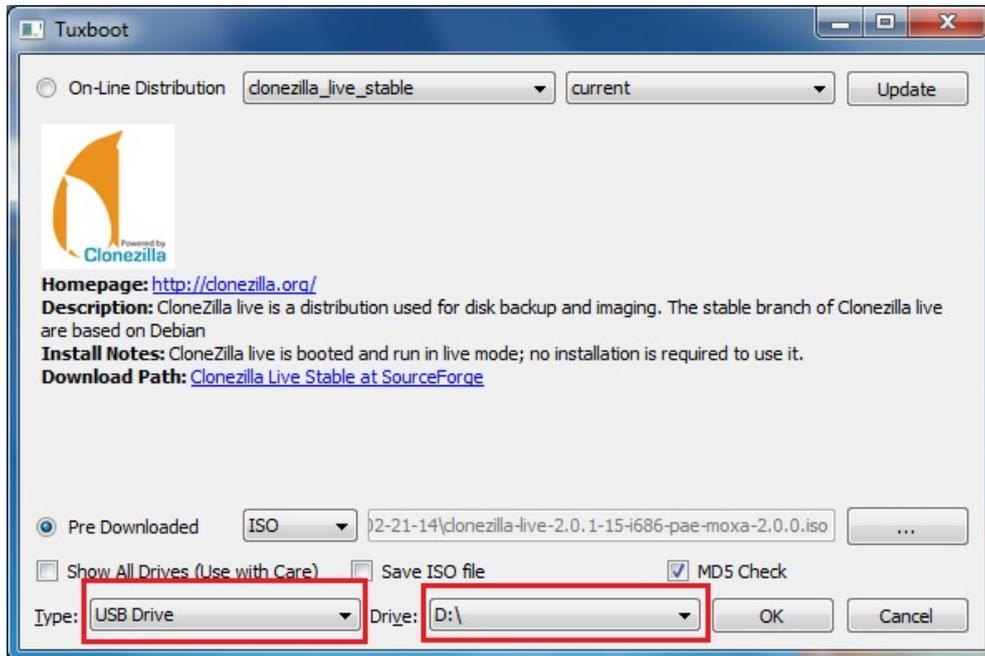
The boot files will be copied to your USB drive.
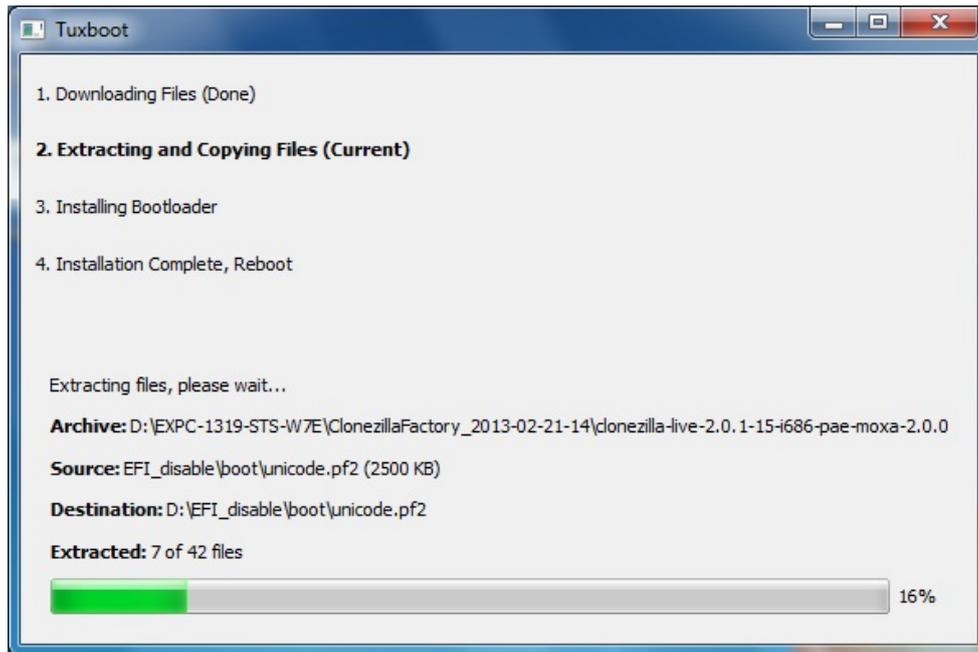
4. Once the boot files are copied, click **Exit** to stop the program.
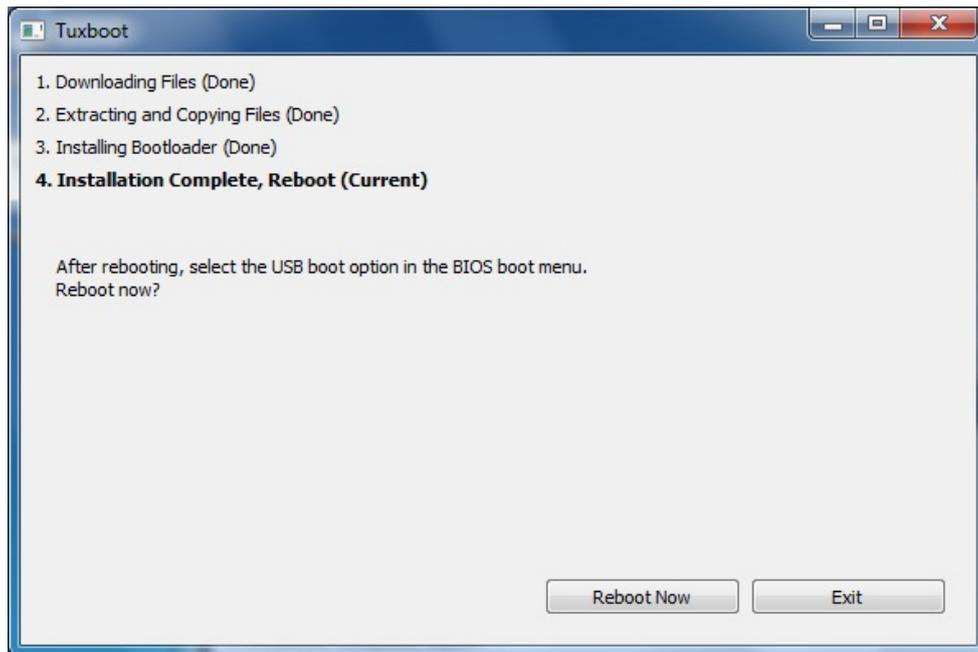
5. Manually copy the **os_image** directory from the *<Software DVD>***recovery** folder to the **\home\partimag** folder on the USB drive.

Step 2: Change the BIOS Settings

You will need to change the BIOS settings of your computer to enable it to boot from the USB disk.

1. Turn on the computer and press **F2** till you hear a beep and the BIOS setup menu is displayed.

2. Select the **Boot** tab and then select **Legacy**. Press **Enter** to continue.

```
          InsydeH20 Setup Utility                    Rev. 3.5
 Main  Advanced  Security  Power  Boot  Exit

                                              Enable/Disable UEFI Boot
                                              Function
 UEFI Boot                <Enabled>
 Quick Boot               <Enabled>
 PXE Boot to LAN          <Disabled>
 USB Boot                 <Enabled>

 ▶EFI
 ▶Legacy









 F1   Help      ↑↓ Select Item   F5/F6 Change Values      F9  Setup Defaults
 Esc  Exit      ↔  Select Menu    Enter Select ▶ SubMenu   F10 Save and Exit
```

3. Select **Boot Type Order**.

```
          InsydeH20 Setup Utility                    Rev. 3.5
                          Boot

 Boot Device Priority                         Select Normal Boot
                                              Option Priority or
 Normal Boot Menu         <Normal>            Advance Boot Option
                                              Priority
 ▶Boot Type Order
 ▶Hard Disk Drive
 ▶USB









 F1   Help      ↑↓ Select Item   F5/F6 Change Values      F9  Setup Defaults
 Esc  Exit      ↔  Select Menu    Enter Select ▶ SubMenu   F10 Save and Exit
```

4. Select the USB disk and then press "+" to move it to the first boot device position.

> ⚠️ **WARNING**
>
> **An incorrect boot priority will lead to recovery failure.**

```
        InsydeH2O Setup Utility              Rev. 3.5
                    Boot

Boot Type Order

USB
CD/DVD-ROM Drive
Hard Disk Drive
Others
```

```
F1   Help       ↑↓ Select Item   F5/F6 Change Values    F9  Setup Defaults
Esc  Exit       ↔ Select Menu    Enter Select ▶ SubMenu F10 Save and Exit
```
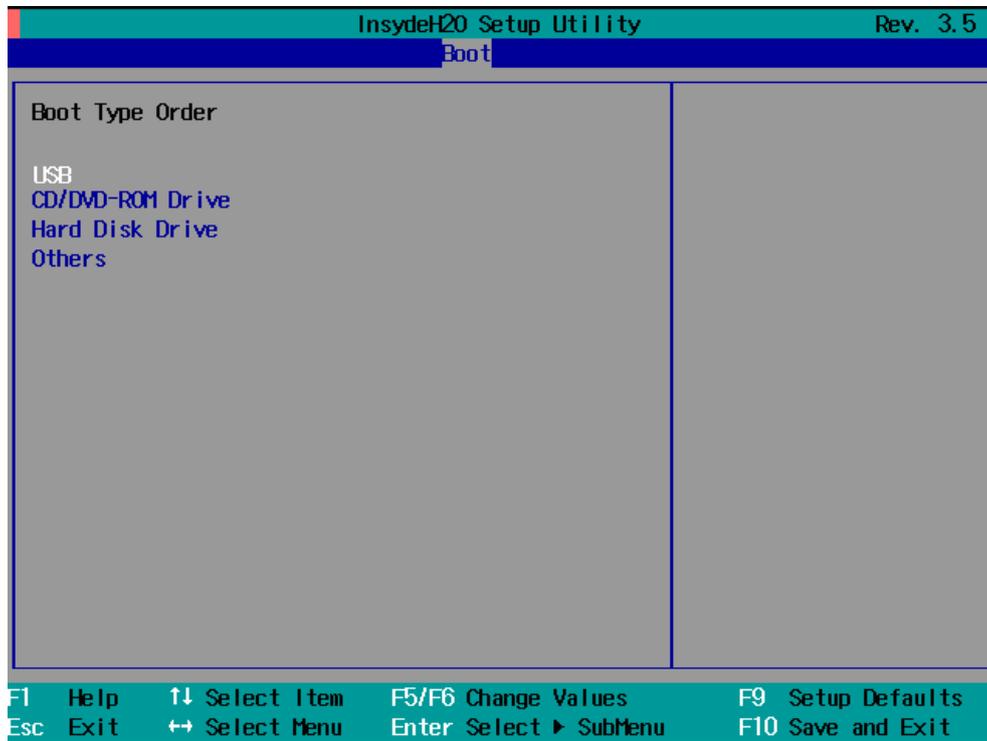
5. Press **F10** and then press **Enter** to save and exit the BIOS setup.

Step 3: Restore the system from USB drive

Connect the USB disk to any of the DA-720's USB ports and then reboot the computer. The system will boot from the USB disk and the **System Save & Restore** utility is displayed.

1. In the utility window, select the **clonezilla live restore disk** option.

```
        Moxa System Save & Restore Utility (V2.4.0-0)

clonezilla live restore disk
clonezilla live save disk




                    Press [Tab] to edit options
```
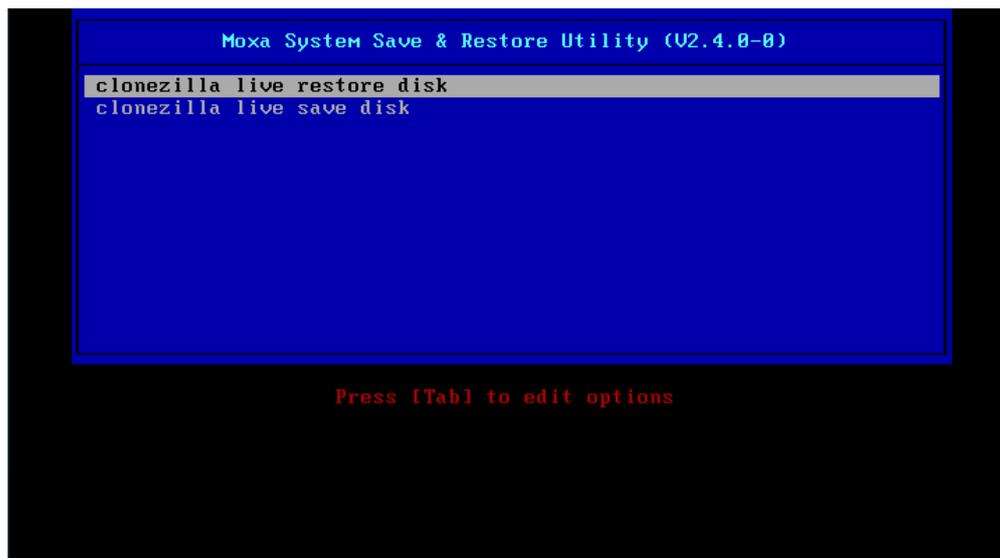
2. Wait for the USB drive boot process to finish.

```
Command (m for help): The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

Warning: Unable to open /dev/sr0 read-write (Read-only file system).  /dev/sr0 has been opened read-
only.
Warning: Unable to open /dev/sr0 read-write (Read-only file system).  /dev/sr0 has been opened read-
only.
Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x469e8113

Device     Boot   Start      End  Sectors  Size Id Type
/dev/sda1          2048  1026047  1024000  500M  7 HPFS/NTFS/exFAT
/dev/sda2       1026048 41943039 40916992 19.5G  7 HPFS/NTFS/exFAT


Disk /dev/sdb: 14.8 GiB, 15846080512 bytes, 30949376 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00000000

Device     Boot Start      End  Sectors  Size Id Type
/dev/sdb1  *     2048 30949375 30947328 14.8G  c W95 FAT32 (LBA)


Disk /dev/loop0: 208.9 MiB, 218980352 bytes, 427696 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

3. Enter **y** to continue the restore process.

```
Do NOT create partition table on the client harddisk!
/usr/share/drbl/sbin/ocs-functions: line 10757: warning: setlocale: LC_ALL: cannot change locale (en
)
/usr/share/drbl/sbin/ocs-functions: line 10739: warning: setlocale: LC_ALL: cannot change locale (en
)
/usr/share/drbl/sbin/ocs-functions: line 10739: warning: setlocale: LC_ALL: cannot change locale (en
)
/usr/share/drbl/sbin/ocs-functions: line 10739: warning: setlocale: LC_ALL: cannot change locale (en
)
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
        LANGUAGE = (unset),
        LC_ALL = "en",
        LANG = "en_US.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to a fallback locale ("en_US.UTF-8").
Activating the partition info in /proc... done!
Getting /dev/sda1 info...
/usr/share/drbl/sbin/ocs-functions: line 3632: warning: setlocale: LC_ALL: cannot change locale (en)
/usr/share/drbl/sbin/ocs-functions: line 3645: warning: setlocale: LC_ALL: cannot change locale (en)
Getting /dev/sda2 info...
/usr/share/drbl/sbin/ocs-functions: line 3632: warning: setlocale: LC_ALL: cannot change locale (en)
/usr/share/drbl/sbin/ocs-functions: line 3645: warning: setlocale: LC_ALL: cannot change locale (en)
***********************************************.
The following step is to restore an image to the hard disk/partition(s) on this machine: "/home/part
imag/os_image" -> "sda sda1 sda2"
The image was created at: 2016-0907-1744
WARNING!!! WARNING!!! WARNING!!!
WARNING. THE EXISTING DATA IN THIS HARDDISK/PARTITION(S) WILL BE OVERWRITTEN! ALL EXISTING DATA WILL
 BE LOST:
***********************************************.
Machine: VMware Virtual Platform
sda (21.5GB_VMware_Virtual_S_No_disk_serial_no)
sda1 (500M(In_VMware_Virtual_S)_No_disk_serial_no)
sda2 (19.5G(In_VMware_Virtual_S)_No_disk_serial_no)
***********************************************.
Are you sure you want to continue? (y/n)
```

4. Enter **y** to confirm again.

```
Getting /dev/sda1 info...
/usr/share/drbl/sbin/ocs-functions: line 3632: warning: setlocale: LC_ALL: cannot change locale (en)
/usr/share/drbl/sbin/ocs-functions: line 3645: warning: setlocale: LC_ALL: cannot change locale (en)
Getting /dev/sda2 info...
/usr/share/drbl/sbin/ocs-functions: line 3632: warning: setlocale: LC_ALL: cannot change locale (en)
/usr/share/drbl/sbin/ocs-functions: line 3645: warning: setlocale: LC_ALL: cannot change locale (en)
*****************************************************.
The following step is to restore an image to the hard disk/partition(s) on this machine: "/home/part
imag/os_image" -> "sda sda1 sda2"
The image was created at: 2016-0907-1744
WARNING!!! WARNING!!! WARNING!!!
WARNING. THE EXISTING DATA IN THIS HARDDISK/PARTITION(S) WILL BE OVERWRITTEN! ALL EXISTING DATA WILL
 BE LOST:
*****************************************************.
Machine: VMware Virtual Platform
sda (21.5GB_VMware_Virtual_S_No_disk_serial_no)
sda1 (500M(In_VMware_Virtual_S)_No_disk_serial_no)
sda2 (19.5G(In_VMware_Virtual_S)_No_disk_serial_no)
*****************************************************.
Are you sure you want to continue? (y/n) y
OK, let's do it!!
This program is not started by clonezilla server.
*****************************************************.
Let me ask you again.
The following step is to restore an image to the hard disk/partition(s) on this machine: "/home/part
imag/os_image" -> "sda sda1 sda2"
The image was created at: 2016-0907-1744
WARNING!!! WARNING!!! WARNING!!!
WARNING. THE EXISTING DATA IN THIS HARDDISK/PARTITION(S) WILL BE OVERWRITTEN! ALL EXISTING DATA WILL
 BE LOST:
*****************************************************.
Machine: VMware Virtual Platform
sda (21.5GB_VMware_Virtual_S_No_disk_serial_no)
sda1 (500M(In_VMware_Virtual_S)_No_disk_serial_no)
sda2 (19.5G(In_VMware_Virtual_S)_No_disk_serial_no)
*****************************************************.
Are you sure you want to continue? (y/n) _
```

5. Wait for the process to finish.

```
                        ┌─────── Partclone ───────┐
                         Partclone v0.2.78 http://partclone.org
                         Starting to restore image (-) to device (/dev/sda1)
                         Calculating bitmap... Please wait... done!
                         File system:  NTFS
                         Device size:  524.3 MB = 127999 Blocks
                         Space in use: 335.3 MB = 81864 Blocks
                         Free Space:   189.0 MB = 46135 Blocks
                         Block size:   4096 Byte



                         Elapsed: 00:00:06 Remaining: 00:00:08   Rate:   1.36GB/min
                         Current Block: 79394  Total Block: 127999

                         Data Block Process:
                                                                       40.65%

                         Total Block Process:
                                                                       62.03%
                        └─────────────────────────┘
```

6. Select **(0) Poweroff** to power off the computer.
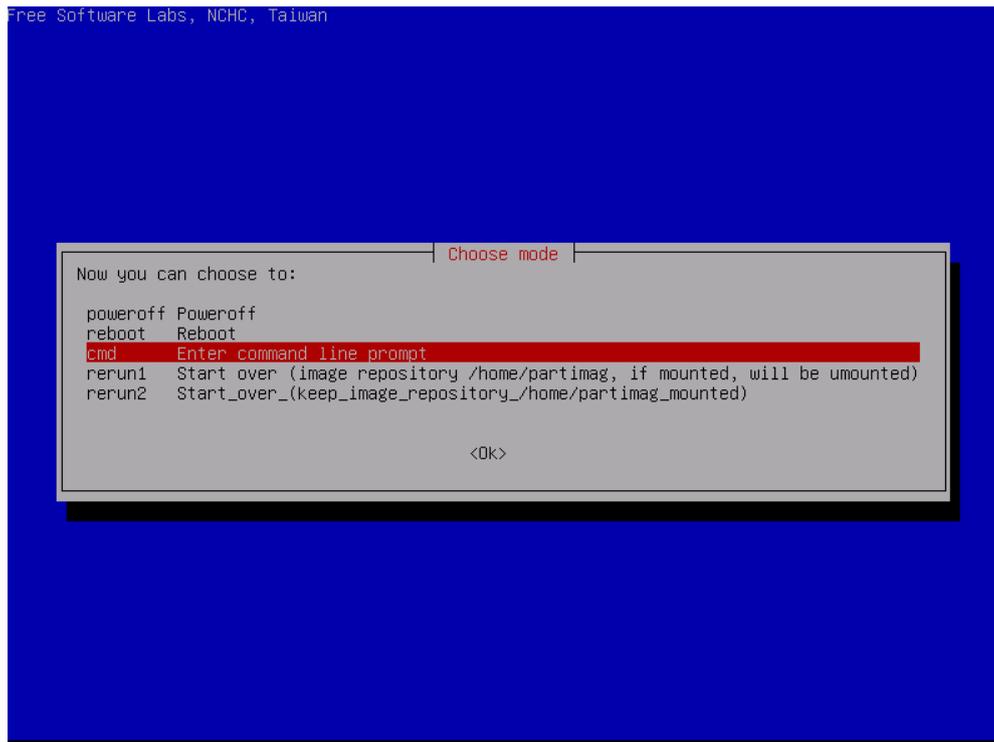
7. Remove the USB drive after the computer has been powered off.

Step 4: Change the BIOS Settings to Boot from the Original Disk

Now you will need to change the boot priority so that the computer can boot from the original disk.

As the system reboots, press **F2** to enter the BIOS setup menu.

1. Select **Hard Disk Boot Priority** and then press + to move to the first boot device position, and then press **Enter**. Make sure the hard disk has first boot priority.
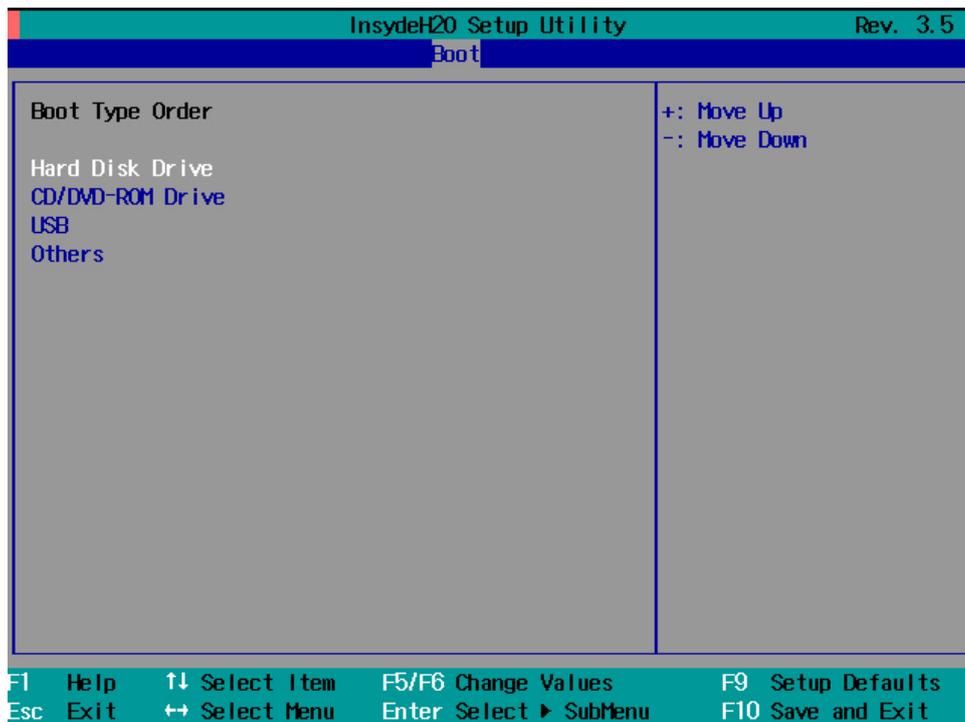
2. Press **F10** and then press **Enter** to save and exit BIOS settings.

Step 5: Reboot the Computer

You need to wait about 10 to 15 minutes for the system to restart two times automatically, since the system configuration files will 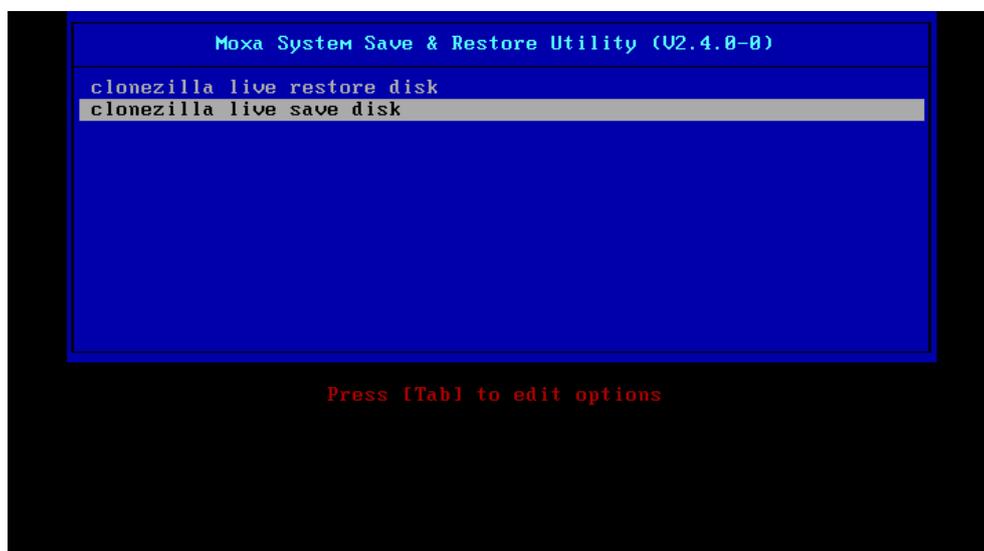be initiated while booting up for the first time. **Do not turn off the computer or shut down the computer** while the system is restarting; otherwise, the IIS service will be terminated. When the operating system has successfully launched, you will need to restart your computer so that the new settings can be activated.

Saving the System Image to the USB Drive

You may also save the current system image to the USB drive for system recovery in case the system crashes. Before saving the system image to the USB drive, we suggest you remove all files under **\home\partimag** on the USB drive. In addition, change the BIOS settings to make the USB drive the first boot priority.

When the system boots up, do the following:

1. Select **clonezilla live save disk**.

```
          Moxa System Save & Restore Utility (V2.4.0-0)

 clonezilla live restore disk
 clonezilla live save disk




                  Press [Tab] to edit options
```

2. Wait for the USB drive boot process to finish.

```
[    5.141941] sd 0:0:1:0: [sdb] Attached SCSI disk
[    5.257277] sd 0:0:0:0: Attached scsi generic sg0 type 0
[    5.269691] sd 0:0:1:0: Attached scsi generic sg1 type 0
[    5.280668] sr 1:0:0:0: Attached scsi generic sg2 type 5
Begin: Loading essential drivers ... [    5.772551] Atheros(R) L2 Ethernet Driver - version 2.2.3
[    5.774561] Copyright (c) 2007 Atheros Corporation.
[    5.863196] Broadcom NetXtreme II 5771x 10Gigabit Ethernet Driver bnx2x 1.62.00-6 (2011/01/30)
[    6.005932] Btrfs loaded
[    6.054095] device-mapper: uevent: version 1.0.3
[    6.059737] device-mapper: ioctl: 4.19.1-ioctl (2011-01-07) initialised: dm-devel@redhat.com
done.
Begin: Running /scripts/init-premount ... done.
Begin: Mounting root file system ... [    6.289382] Uniform Multi-Platform E-IDE driver
[    6.301889] ide_generic: please use "probe_mask=0x3f" module parameter for probing all legacy ISA
 IDE ports
[    6.801141] NTFS driver 2.1.30 [Flags: R/W MODULE].
[    6.914295] NTFS volume version 3.1.
Begin: Running /scripts/live-premount ... done.
[    7.331989] FAT: utf8 is not a recommended IO charset for FAT filesystems, filesystem will be cas
e sensitive!
[    7.453369] aufs: module is from the staging directory, the quality is unknown, you have been war
ned.
[    7.479098] aufs 2.1-standalone.tree-38-rcN-20110228
[    7.610228] loop: module loaded
[    7.905144] squashfs: version 4.0 (2009/01/31) Phillip Lougher
Begin: Running /scripts/live-realpremount ... done.
Begin: Mounting "/live/image/live/filesystem.squashfs" on "//filesystem.squashfs" via "/dev/loop0" .
.. done.
done.
Begin: Running /scripts/live-bottom
 ... Begin: Configuring fstab ... done.
Begin: Preconfiguring networking ... done.
Begin: Loading preseed file ... done.
Begin: Running /scripts/init-bottom ... done.
INIT: version 2.88 booting
Using makefile-style concurrent boot in runlevel S.
_
```

3. Enter **y** to continue.

```
Setting the TERM as linux
*********************************************************.
Clonezilla image dir: /home/partimag
*********************************************************.
Shutting down the Logical Volume Manager
  No volume groups found
  No volume groups found
Finished Shutting down the Logical Volume Manager
Selected device [sda] found!
The selected devices: sda
*********************************************************.
Activating the partition info in /proc... done!
Selected device [sda] found!
The selected devices: sda
Searching for data partition(s)...
Excluding busy partition or disk...
Unmouted partitions (including extended or swap): sda1
Collecting info.. done!
Searching for swap partition(s)...
Excluding busy partition or disk...
Unmouted partitions (including extended or swap): sda1
Collecting info.. done!
The data partition to be saved:    sda1
The swap partition to be saved:
Activating the partition info in /proc... done!
Selected device [sda1] found!
The selected devices: sda1
Getting /dev/sda1 info...
*********************************************************.
The following step is to save the hard disk/partition(s) on this machine as an image:
*********************************************************.
Machine: VirtualBox
sda (2103MB_VBOX_HARDDISK__ata-VBOX_HARDDISK_VB1c64a0a3-c9f7523d)
sda1 (2065MB_ntfs(In_VBOX_HARDDISK_)_ata-VBOX_HARDDISK_VB1c64a0a3-c9f7523d)
*********************************************************.
-> "/home/partimag/xpe_savedisk".
Are you sure you want to continue? ? (y/n) y
```
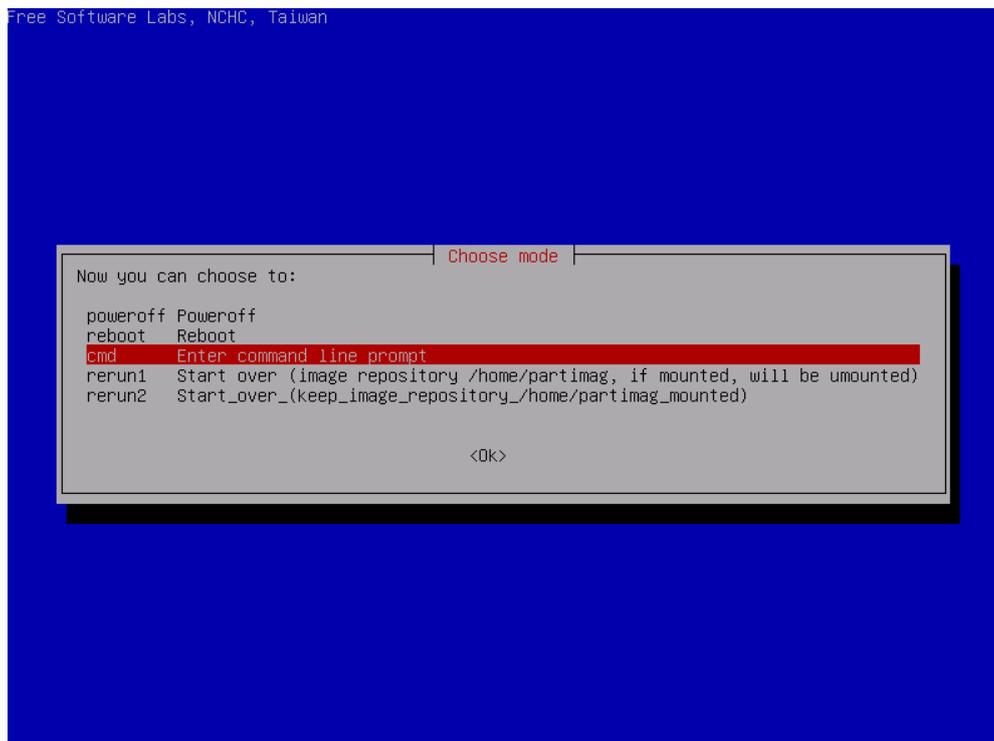
4. Wait for the process to finish.

```
 /dev/sdb1: read failed after 0 of 2048 at 0: Input/output error
  No volume groups found
  No volume groups found
Finished Shutting down the Logical Volume Manager
Checking the integrity of partition table in the disk /dev/sda...
Reading the partition table for /dev/sda...RETVAL=0
*******************************************************.
*******************************************************.
done!
Saving the MBR data for sda...
1+0 records in
1+0 records out
512 bytes (512 B) copied, 0.00347646 s, 147 kB/s
*******************************************************.
*******************************************************.
Starting saving /dev/sda1 as /home/partimag/xpe_savedisk/sda1.XXX...
/dev/sda1 filesystem: ntfs.
*******************************************************.
Checking NTFS integrity in /dev/sda1... done!
Checking the disk space...
Use ntfsclone with gzip to save the image.
Image file will be split with size limit 1000000 MB.
*******************************************************.
If this action fails or hangs, check:
* Is the disk full ?
*******************************************************.
ntfsclone v2.0.0 (libntfs 10:0:0)
NTFS volume version: 3.1
Cluster size       : 2048 bytes
Current volume size: 2064510976 bytes (2065 MB)
Current device size: 2064513024 bytes (2065 MB)
Scanning volume ...
100.00 percent completed
Accounting clusters ...
Space in use       : 1770 MB (85.7%)
Saving NTFS to image ...
_ 0.64 percent completed
```

5. Select **(0) Poweroff** so that the computer will power off when the process is finished.

```
Free Software Labs, NCHC, Taiwan




                            ┌ Choose mode ┐
       Now you can choose to:

       poweroff  Poweroff
       reboot    Reboot
       cmd       Enter command line prompt
       rerun1    Start over (image repository /home/partimag, if mounted, will be umounted)
       rerun2    Start_over_(keep_image_repository_/home/partimag_mounted)


                              <Ok>
```

7

DE-PRP-HSR-EF Expansion Module

This chapter describes how to operate the DE-PRP-HSR-EF card on a DA-720 computer, and includes information on configuring the DE-PRP-HSR-EF function using the configuration utility.

The following topics are covered in this chapter:

❒ **Software Installation**
- ➢ Installing the DE-PRP-HSR-EF Utility
- ➢ Checking the PRP/HSR Related Service and Program

❒ **Configuring the DE-PRP-HSR-EF**
- ➢ Setting the Operating Mode
- ➢ Getting the DE-PRP-HSR-EF Status
- ➢ Getting the FPGA Version
- ➢ Getting the Status of the Supervision Frame
- ➢ Configuring Alert Notifications

❒ **API Reference**
- ➢ Overview

❒ **API Functions**
- ➢ Set Operating Mode
- ➢ Get RX/TX Counters

Software Installation

## Installing the DE-PRP-HSR-EF Utility

1. Extract the DE-PRP-HSR_V1.0_Utility.zip file and run WinPcap-4-1-3.exe to install the **WinPcap** setup package.
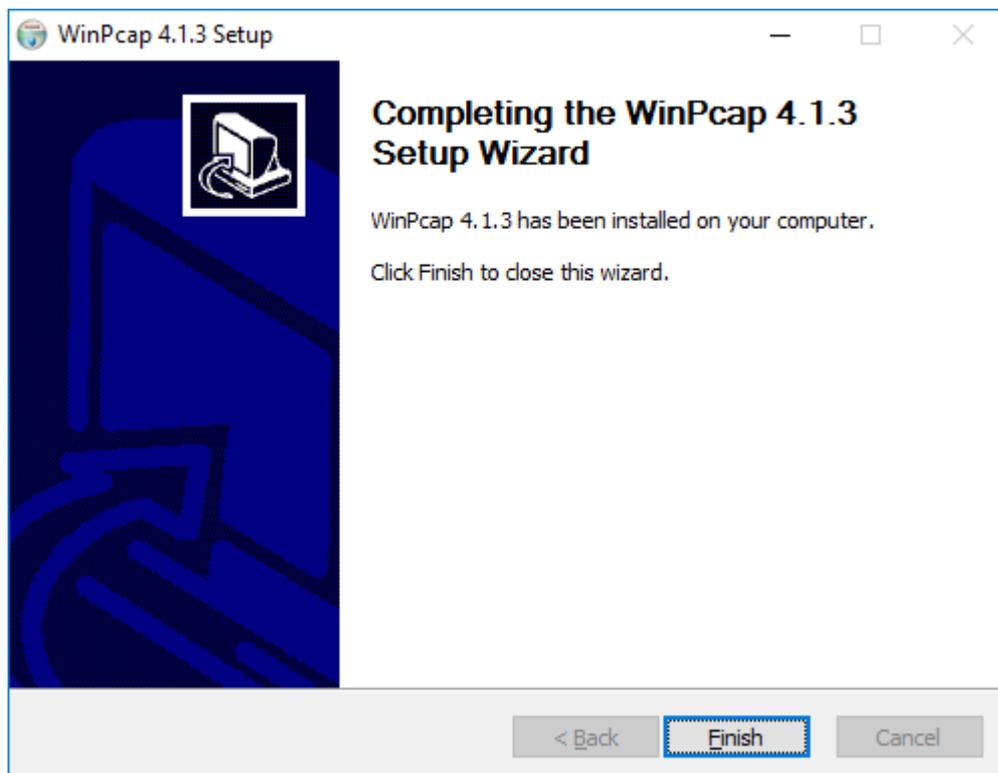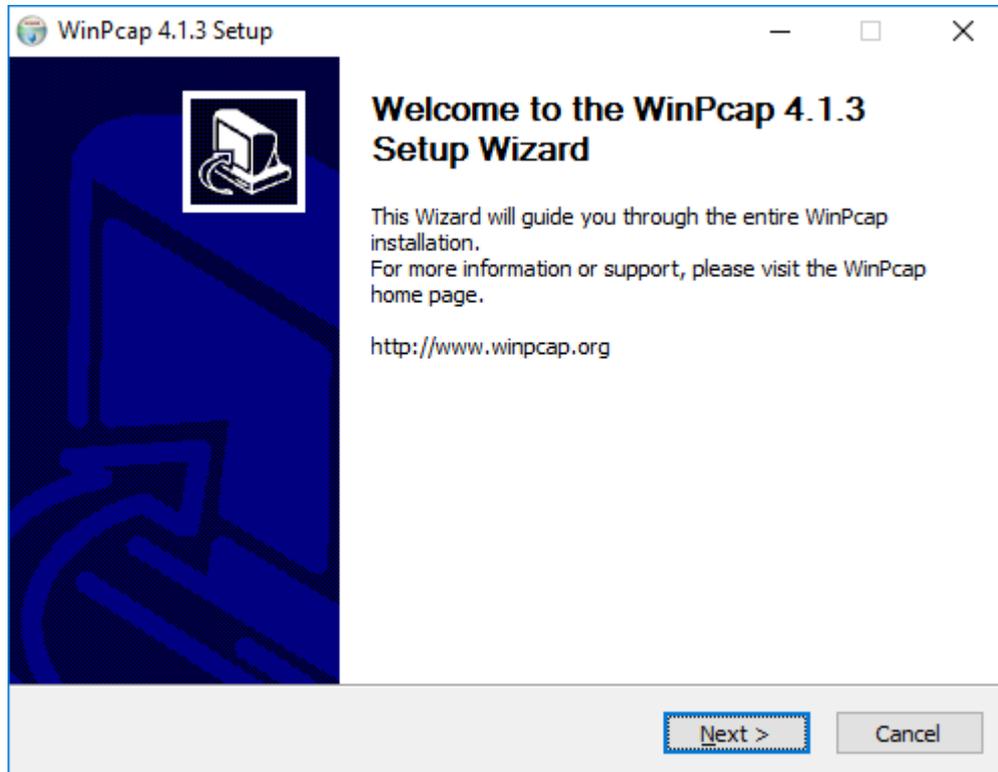
2. Obtain the MxPrpSetup.msi file on the DE-PRP-HSR-EF Software CD/DVD or download the file from Moxa's support website at http://www.moxa.com/support/download_center.asp. Then, save the file on the DA-720 computer.

3. Double-click on the MxPrpSetup.msi file to start the installation process.
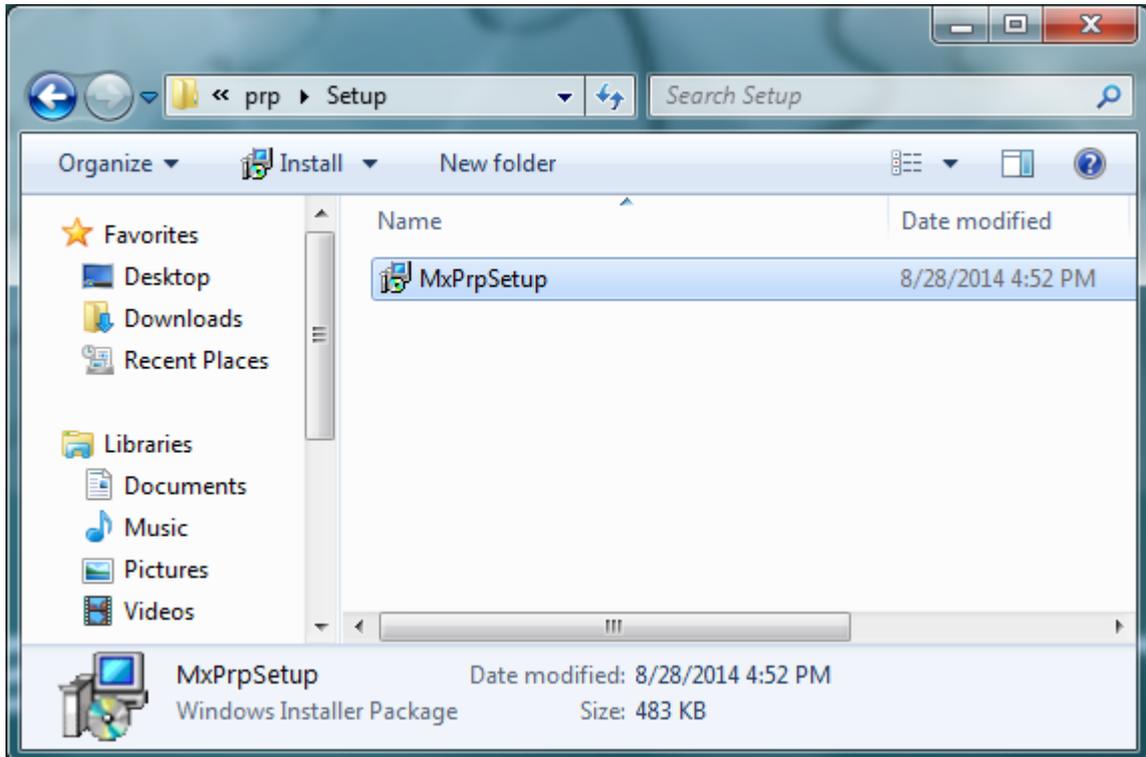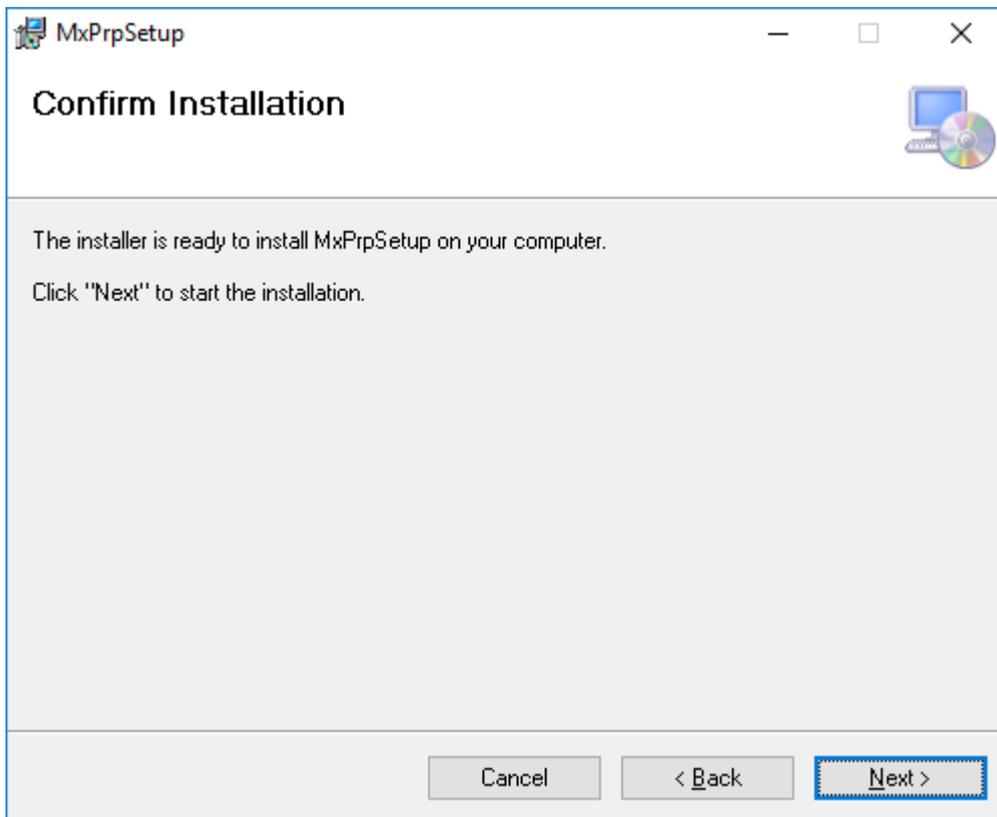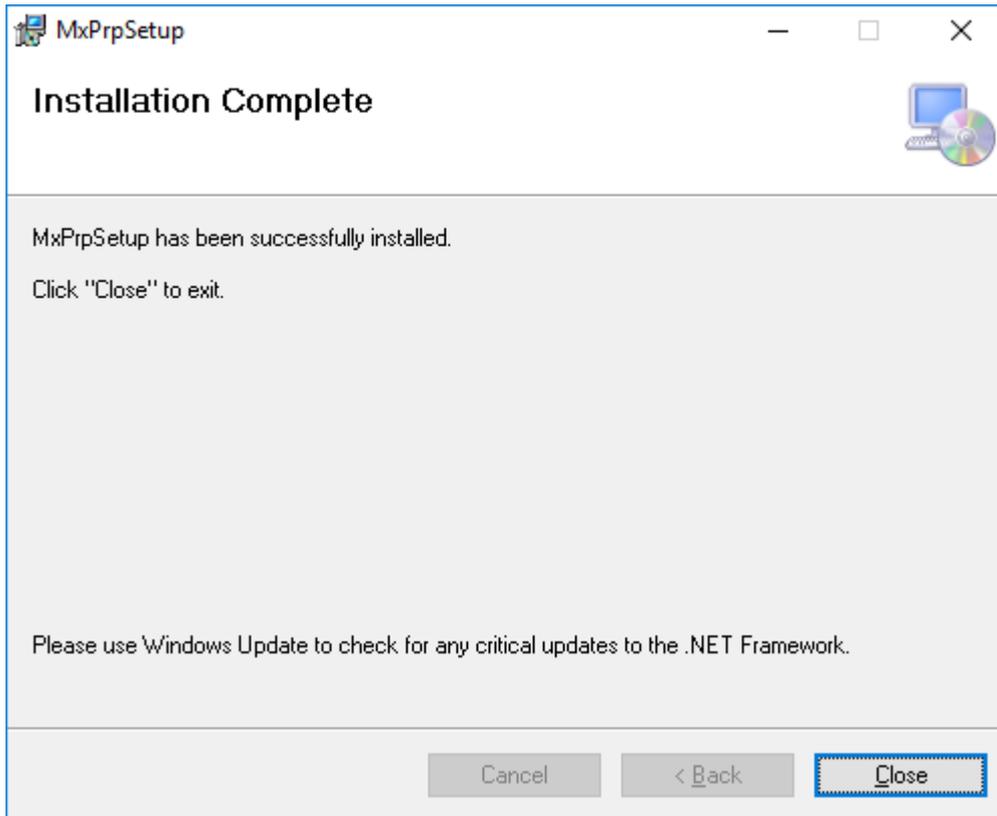

4. Click the **Next** button on the welcome screen.
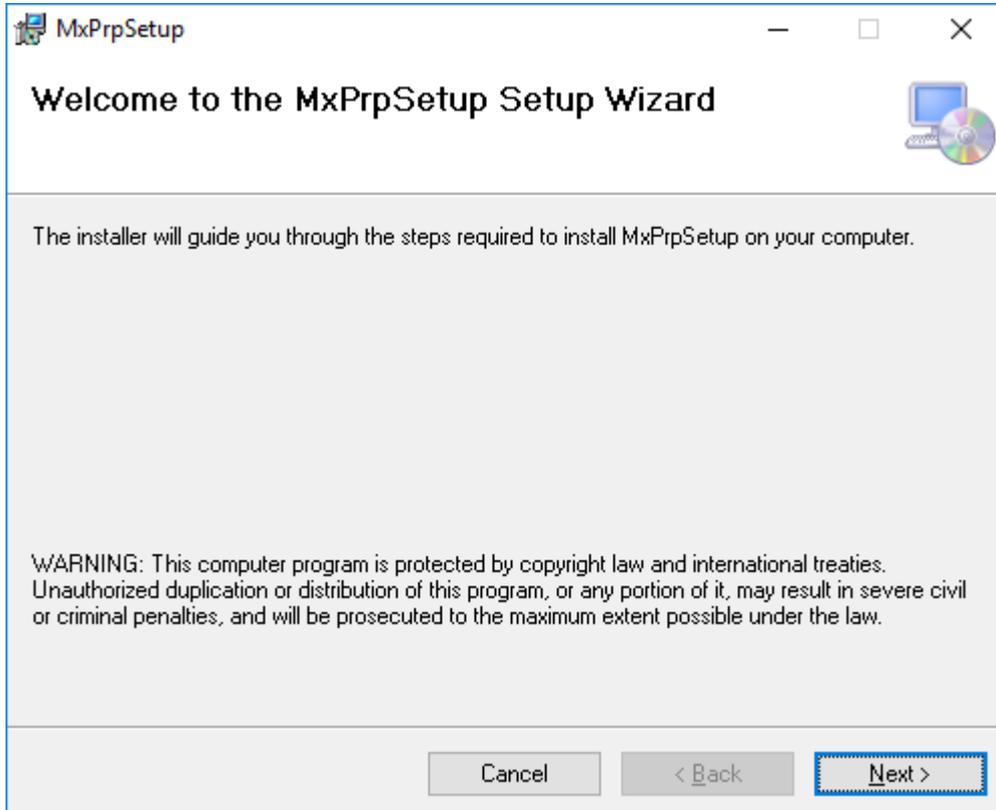
5. Accept the default installation directory or click **Browse** to select one and Click **Next**.
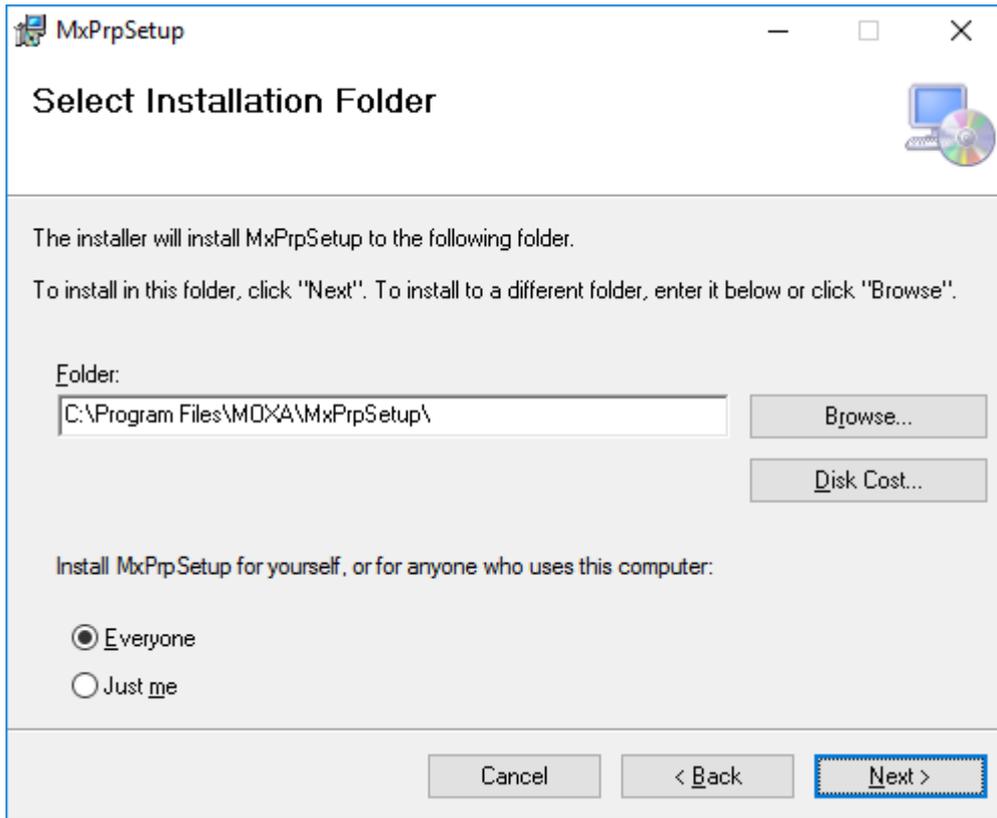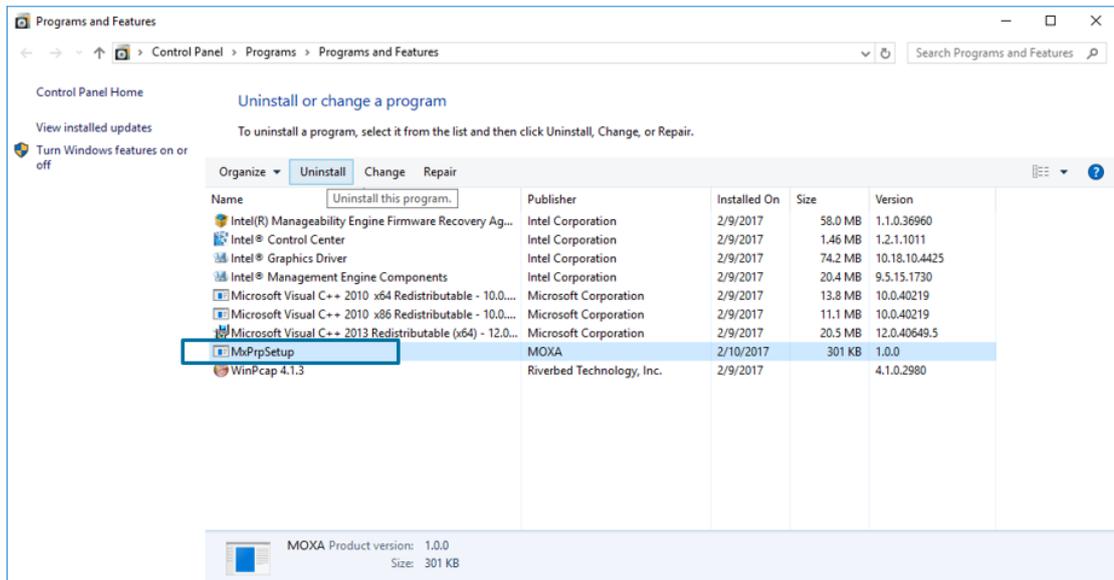

6. Click **Next** to continue.

7. Click **Close** to complete the installation.
    The Moxa PRP Service will be installed on the computer.
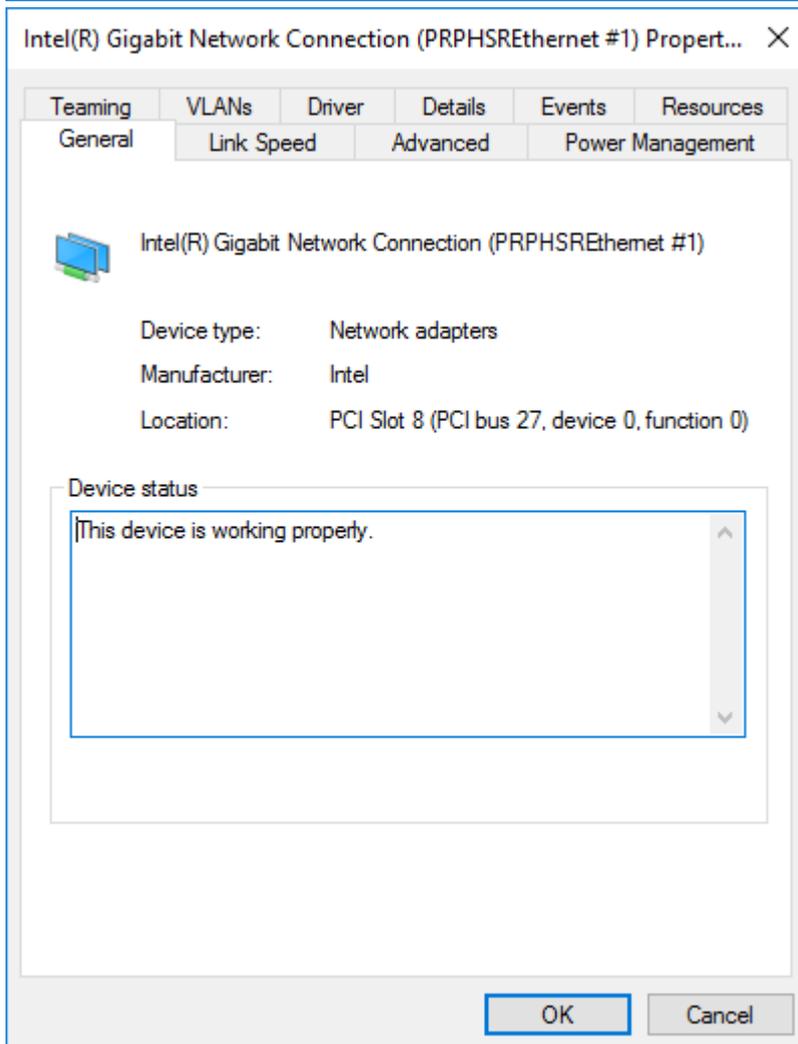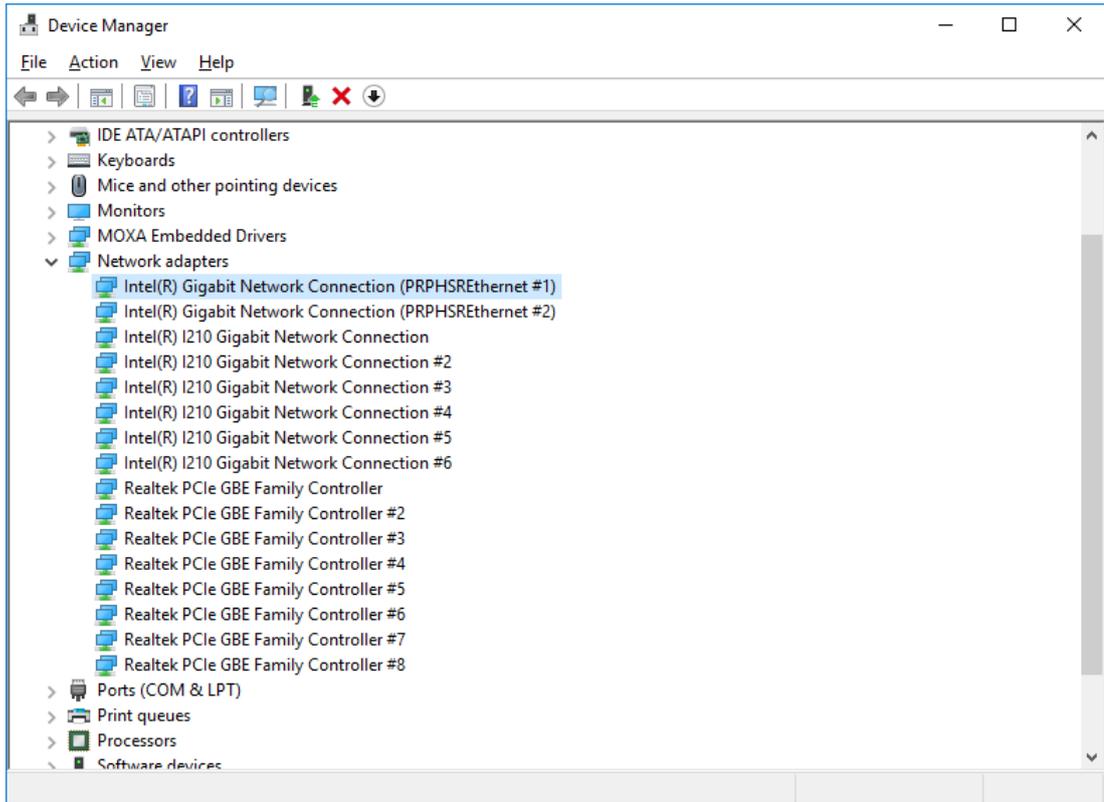

Checking the PRP/HSR Related Service and Program

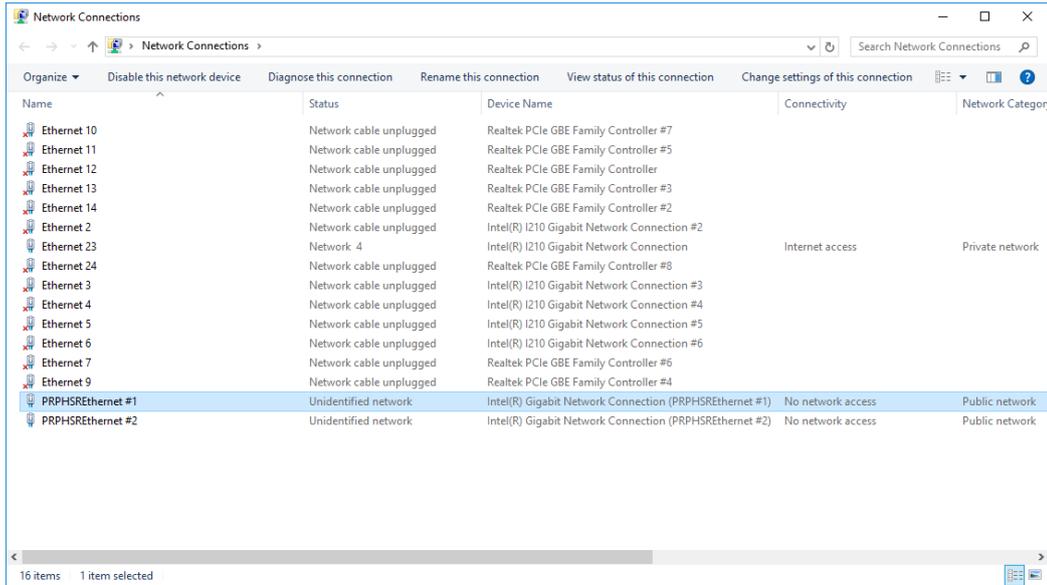1. In the Windows Programs and Features window, check the software version.
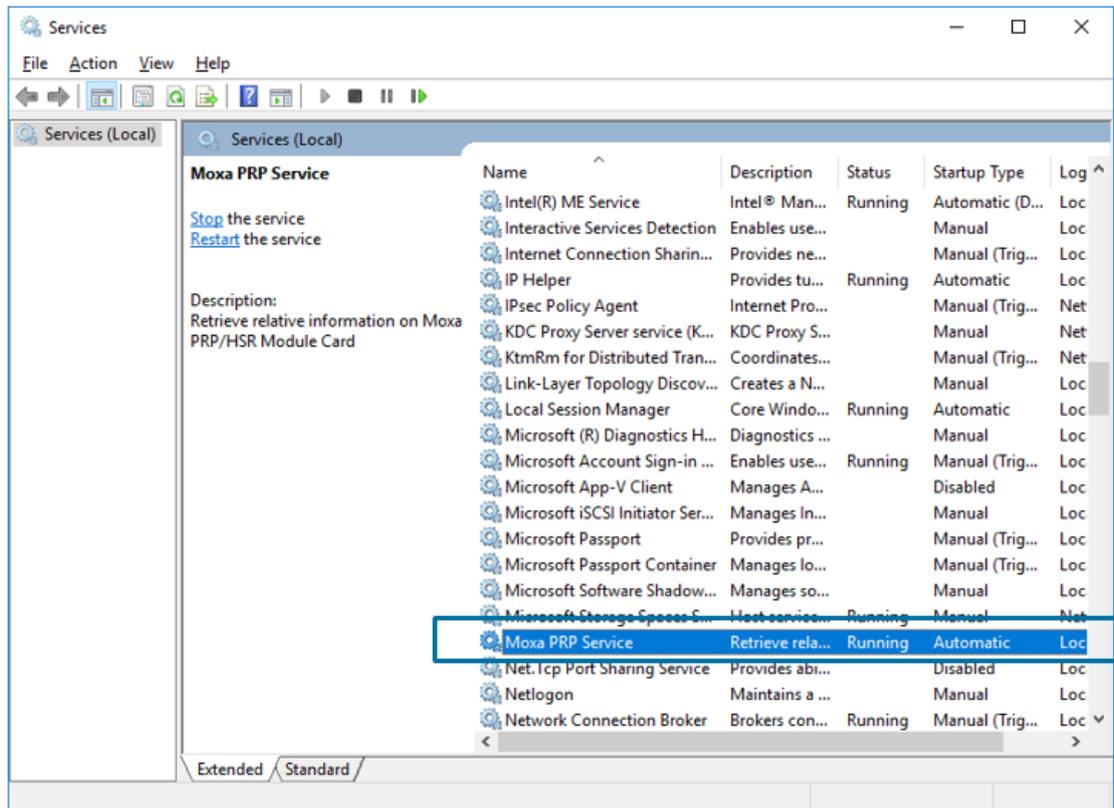
2. Check if the network adapter is renamed correctly.

The network adapter names should have the prefix `PRPHSREthernet` as shown below:

3. Check if the network connections are renamed correctly based on the network adapter name.

4. Wait for 1 minute to confirm that the PRP/HSR service is running continuously without stopping.

5. Check if the **PRPSuperVisionFrame.exe** is running correctly.
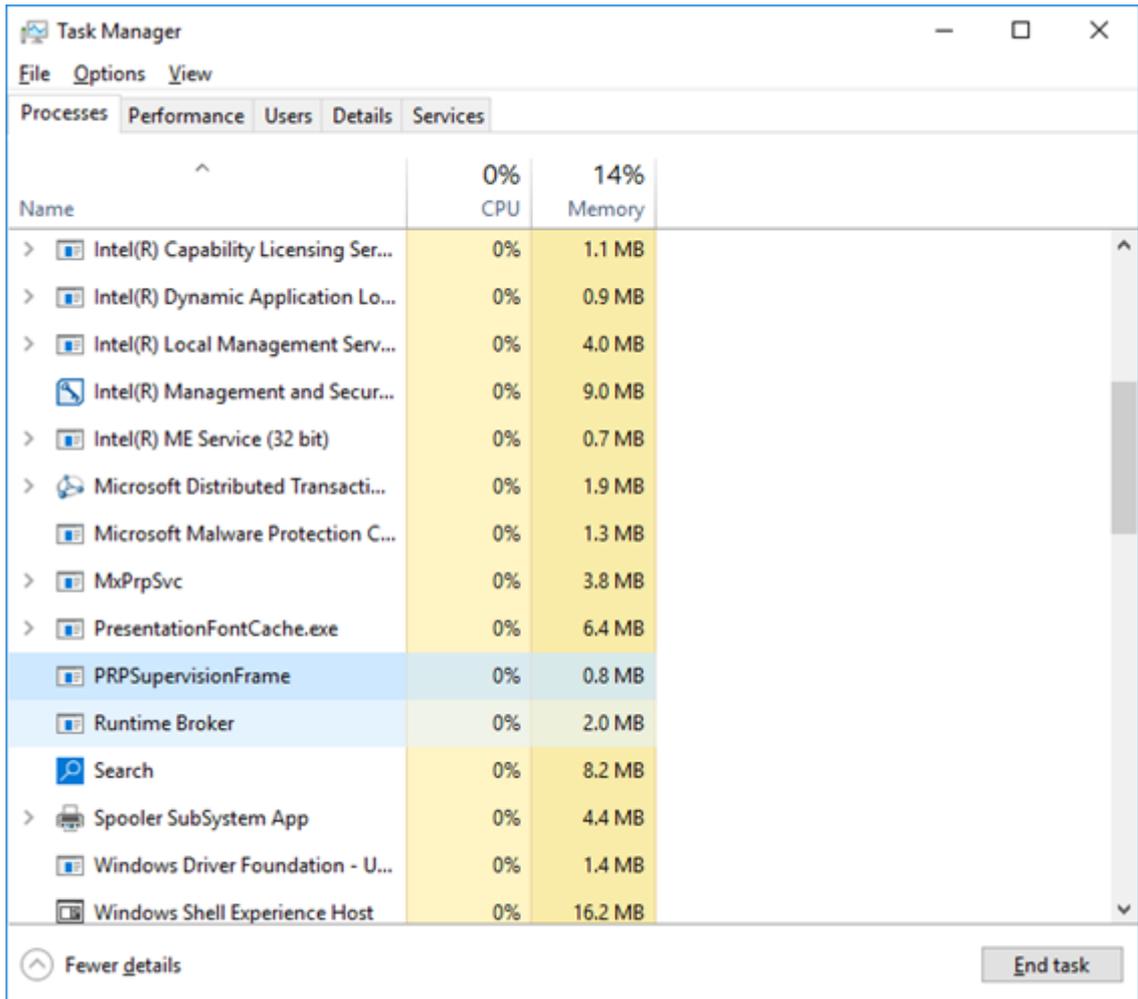
Configuring the DE-PRP-HSR-EF

You can use the Moxa PRP Settings utility to set the operating mode on the DE-PRP-HSR-EF.

Setting the Operating Mode

1. Open the Moxa PRP Settings utility from the Start menu.
2. If more than one DE-PRP-HSR-EF is installed on the computer, select the index (1 to 3) of the DE-PRP-HSR-EF that you want to configure from the Module Index drop-down list.
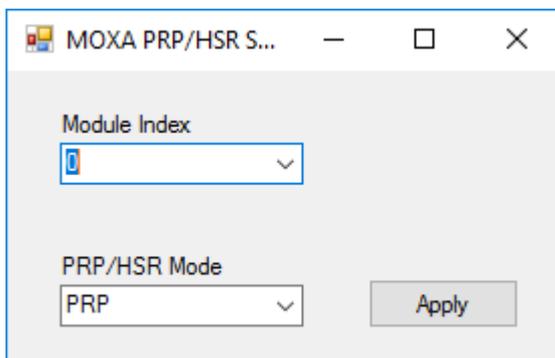
3. From the PRP/HSR Mode drop-down list, select an option and click **Apply**.
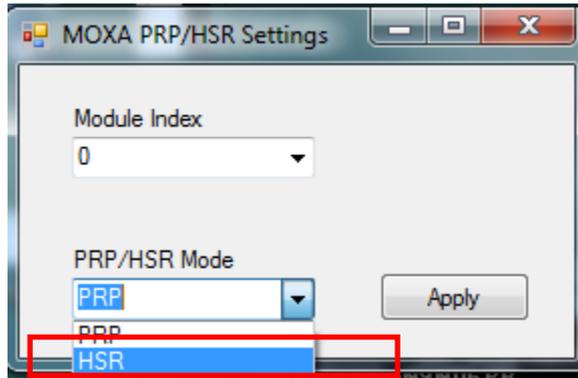
4. Wait for the service to make the changes.

Getting the DE-PRP-HSR-EF Status

To get the counter value, do the following:

1. Run the **GetRXCounters.exe** program from the DE-PRPHSR_V1.0_example.zip file.

2. Get RX count values from the internal ports, port A and port B

3. Get the TX count value.
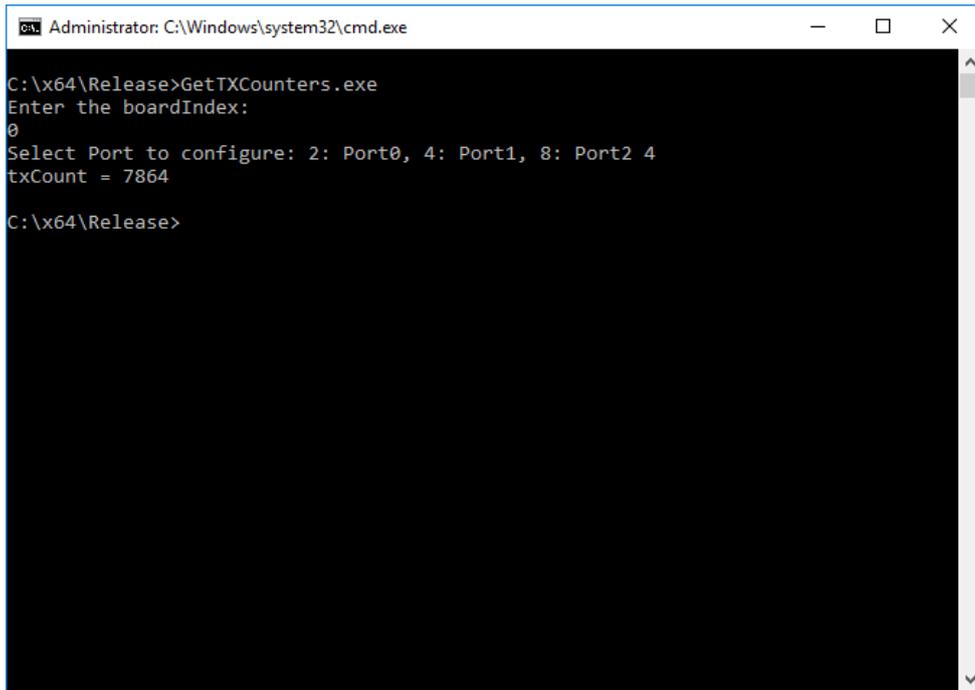
Getting the FPGA Version

1. Execute the GetFPGAVersion.exe program

2. Check if the read status and the firmware version are correct.

Getting the Status of the Supervision Frame

Open **Wireshark** using the `hsr_prp_supervision` filter and check if the Supervision frame is sent correctly.
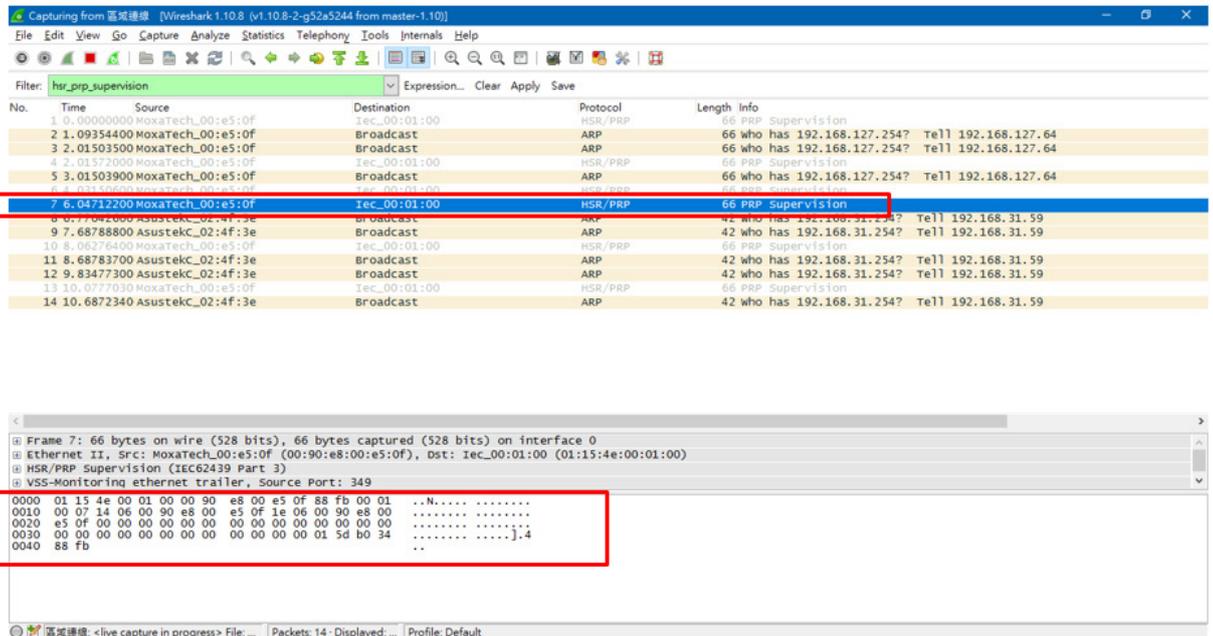
Configuring Alert Notifications

Setting Custom Actions

1. Connect the PRP/HSR card with a RedBox.
2. Stop the Moxa PRP/HSR service.
3. Check if the relay output is set to high.

4. Add the relay output program to **`AlertNotification.cmd`** located in the **AlertNotification** folder.
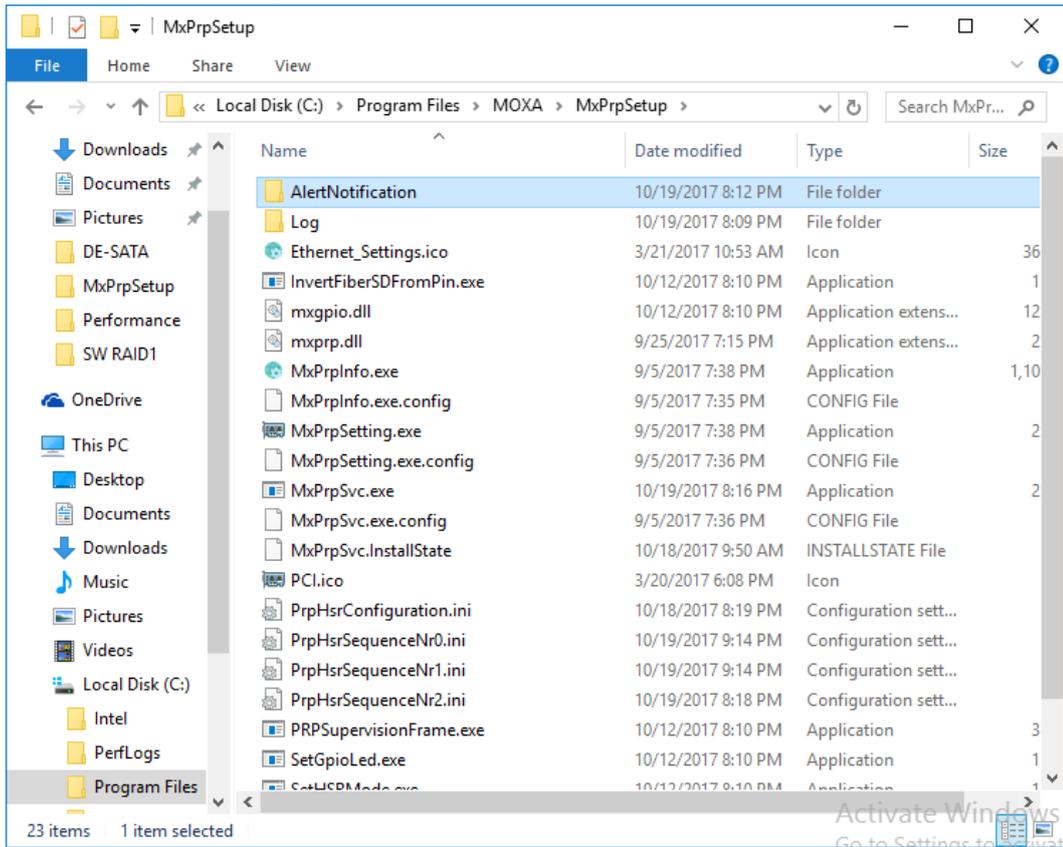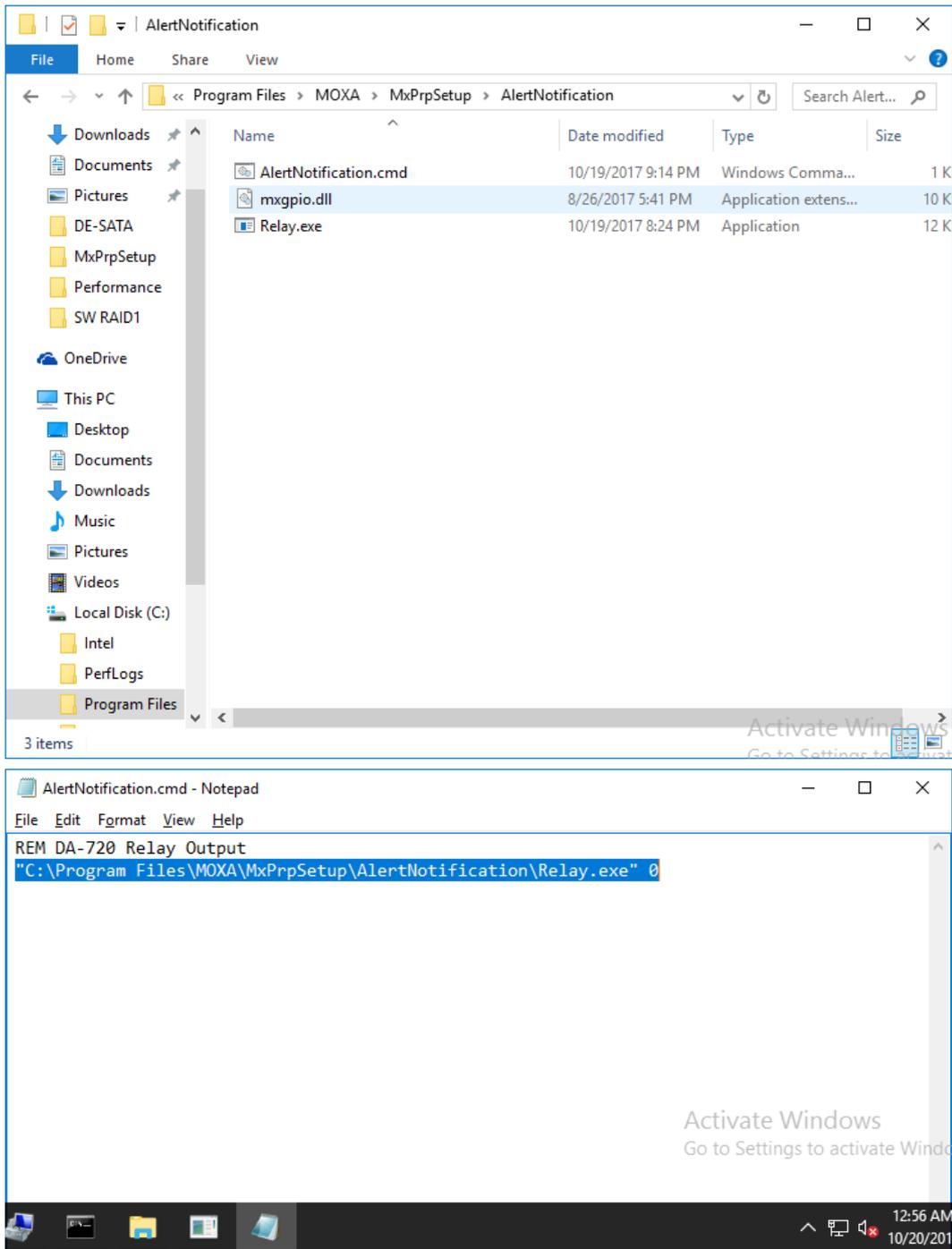
5. Check the current status of the relay using the **RelaySample.exe** test program.

   (The RelaySample.exe program is available in the DA-720-Win10_V1.0_Example.zip under the examples folder)

   ```
   RELAY Test Program
           (0) Exit Program
           (1) Display RELAY pin status
           (2) Set RELAY pin value
   1
   Status = HIGH
   RELAY Test Program
           (0) Exit Program
           (1) Display RELAY pin status
           (2) Set RELAY pin value
   ```

6. Start the Moxa PRP/HSR service.

7. Disconnect all connections on port A and port B

8. Check if the relay output is set to low
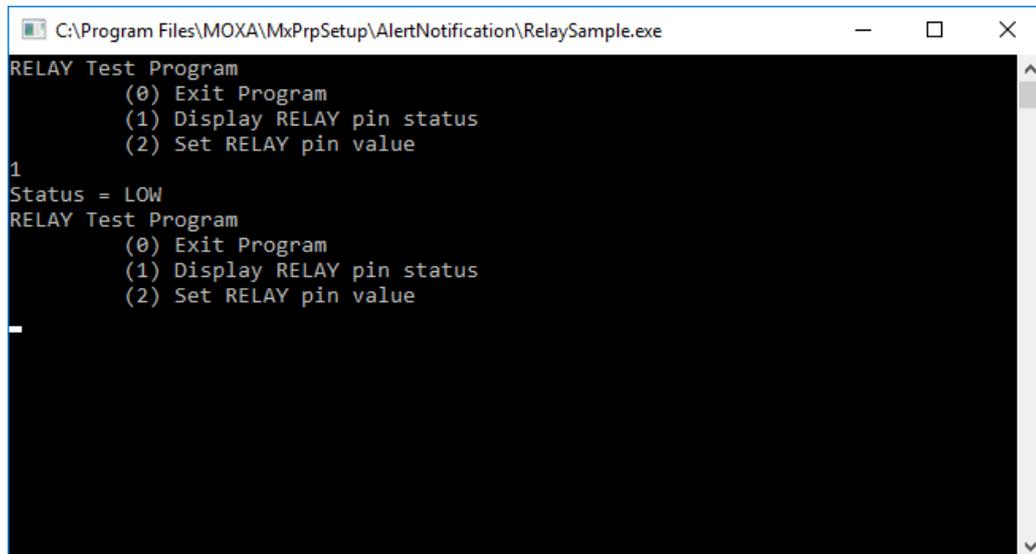
   ```
   C:\Program Files\MOXA\MxPrpSetup\AlertNotification\RelaySample.exe        —   □   ×
   RELAY Test Program
           (0) Exit Program
           (1) Display RELAY pin status
           (2) Set RELAY pin value
   1
   Status = LOW
   RELAY Test Program
           (0) Exit Program
           (1) Display RELAY pin status
           (2) Set RELAY pin value
   ```

API Reference

Overview

The Moxa PRP/HSR API provides a set of C functions for communicating with hardware devices.
The Moxa PRP/HSR API supports the following tasks:

- Setting the operating mode (PRP/HSR).
- Getting RX/TX counters

API Functions

Set Operating Mode

SetPrpMode	Set the operating mode to PRP mode.
SetHHsrMode	Set the operating mode to HSR mode.

SetPRPMode

```
int SetPrpMode(UINT8 slaveAddr);
```

Parameters

slaveAddr : The system management bus (smbus) slave address.

Return Value

Return value(ret) is zero on success, a negative error code on failure.

ret = 0 :   Success

ret = -1:   Error

Description

The SetPRPMode function is used to set the operating mode to PRP mode.

Example

```
#include "stdafx.h"
#include <windows.h>
#include "mxprp.h"

bool CheckMxPrpService();

#define LED_INDEX_PRP 0
```

```
/***************************************************************/
int _tmain(int argc, _TCHAR* argv[])
{
    int boardIndex = 0;
    int ledIndex = LED_INDEX_PRP;
    UINT8 slaveAddr = 0x0;
    bool isServiceExist = false;

    isServiceExist = CheckMxPrpService();
    if(isServiceExist==true)
    {
        printf("mxPrpService is running, to run this program, please stop the
mxPrpService\n");
        printf("Exit the program\n");
        return 0;
    }

    if(argc < 2)
    {
        /* Get user selection for smbus slave address */
        printf("Enter the boardIndex:\n");
        scanf("%d", &boardIndex);
    }
    else
    {
        boardIndex = _wtol(argv[1]);
    }

    slaveAddr = SMBUS_SLAVE_ADDRESS
    slaveAddr += (UINT8)boardIndex;

    /* Set the PRP mode */
    SetPrpMode(slaveAddr);

    /* Set the link mode */
    SetLinkSpeedMode(slaveAddr);

    /* Set the module led*/
    ConfigureGpioMode(slaveAddr, ledIndex);
    SetLedValue(slaveAddr, ledIndex);

    return 0;
}

bool CheckMxPrpService()
{
    bool isServiceExists = false;
    HANDLE hMutex;

    hMutex = CreateMutex(
        NULL, // default security descriptor
        FALSE, // mutex not owned
        TEXT("Global\\MxPrpSvcMutex")); // object name

    if (hMutex == NULL)
```

```
    {
      printf("CreateMutex error: %d\n", GetLastError() );
    }
   else
    {
      if ( GetLastError() == ERROR_ALREADY_EXISTS )
        {
            isServiceExists = true;
        }
      else
        {
            isServiceExists = false;

            /* No service is running, now we can close the mutex and run our program
*/
            CloseHandle(hMutex);
        }
    }


    return isServiceExists;
}
```

SetHSRMode

```
int SetHsrMode(UINT8 slaveAddr);
```

Parameters

slaveAddr : The smbus slave address.

Return Value

Return value (ret) is zero on success, a negative error code on failure.

ret = 0 :   Success
ret = -1:   Error

Description

The SetHSRMode function is used to set the operating mode to HSR mode.

Example

```
#include "stdafx.h"
#include <windows.h>
#include "mxprp.h"

bool CheckMxPrpService();

#define LED_INDEX_HSR 1
```

```c
/*************************************************************/
int _tmain(int argc, _TCHAR* argv[])
{
    int boardIndex = 0;
    int ledIndex = LED_INDEX_HSR;
    UINT8 slaveAddr = 0x0;
    bool isServiceExist = false;

    isServiceExist = CheckMxPrpService();
    if(isServiceExist==true)
    {
        printf("mxPrpService is running, to run this program, please stop the
mxPrpService\n");
        printf("Exit the program\n");
        return 0;
    }

    if(argc < 2)
    {
        /* Get user selection for smbus slave address */
        printf("Enter the boardIndex:\n");
        scanf("%d", &boardIndex);
    }
    else
    {
        boardIndex = _wtol(argv[1]);
    }

    slaveAddr = SMBUS_SLAVE_ADDRESS
    slaveAddr += (UINT8)boardIndex;

    /* Set the HSR mode */
    SetHsrMode(slaveAddr);

    /* Set the link mode */
    SetLinkSpeedMode(slaveAddr);

    /* Set the module led*/
    ConfigureGpioMode(slaveAddr, ledIndex);
    SetLedValue(slaveAddr, ledIndex);

    return 0;
}

bool CheckMxPrpService()
{
    bool isServiceExists = false;
    HANDLE hMutex;

    hMutex = CreateMutex(
        NULL, // default security descriptor
        FALSE, // mutex not owned
        TEXT("Global\\MxPrpSvcMutex")); // object name
```

```
    if (hMutex == NULL)
    {
       printf("CreateMutex error: %d\n", GetLastError() );
    }
    else
    {
      if ( GetLastError() == ERROR_ALREADY_EXISTS )
        {
            isServiceExists = true;
        }
       else
        {
            isServiceExists = false;

            /* No service is running, now we can close the mutex and run our program
*/
            CloseHandle(hMutex);
        }
    }

    return isServiceExists;
}
```

Get RX/TX Counters

GetRxCounter	Get the RX counters from hardware device
GetTxCounter	Get the TX counters from hardware device

GetRxCounter

**int GetRxCounter(UINT8 slaveAddr, UINT8 portSel, unsigned short counterOffset,
unsigned short *rxCounter);**

Parameters

slaveAddr :   The smbus slave address.

portSel: Port select, interlink, port A or port B.

counterOffset: Counter address offset.

rxCounter: The RX counter value.

Return Value

Return value(ret) is zero on success, a negative error code on failure.

ret = 0 :   Success

Description

The GetRXCounter function is used to obtain the RX counter of the interlink ports, port A and port B.

Example

```c
#include "stdafx.h"
#include <windows.h>
#include "mxprp.h"

bool CheckMxPrpService();
unsigned short ReverseBytes(unsigned short value);
unsigned int GetRxGoodOctets(UINT8 slaveAddr, UINT8 portSel);
unsigned int GetRxErrOctets(UINT8 slaveAddr, UINT8 portSel);
unsigned int GetRxWrongLan(UINT8 slaveAddr, UINT8 portSel);
unsigned int GetRxDuplicate(UINT8 slaveAddr, UINT8 portSel);

/****************************************************************/
int _tmain(int argc, _TCHAR* argv[])
{
    UINT8 portSel = 0;
    UINT8 funcSel = 0;
    unsigned short counterOffset = 0;
    unsigned int rxCounter = 0x0;
    int boardIndex = 0;
    UINT8 slaveAddr = 0x0;
    bool isServiceExist = false;

    isServiceExist = CheckMxPrpService();
    if(isServiceExist==true)
    {
        printf("mxPrpService is running, to run this program, please stop the
mxPrpService\n");
        printf("Exit the program\n");
        return 0;
    }

    /* Get user selection for smbus slave address */
    printf("Enter the boardIndex:\n");
    scanf("%d", &boardIndex);
    slaveAddr = SMBUS_SLAVE_ADDRESS;
    slaveAddr += boardIndex;

    /* Configure Counter Control bit to 1 to update the values in counter registers
*/
    printf("Select Port to configure: 2: Port0, 4: Port1, 8: Port2\n");
    scanf("%d", &portSel);
    printf("portSel = %d\n", portSel);

    /* Get the counter value */
    printf("Select counter value to get:\n \
                                    0: RX GOOD OCTETS\n \
                                    1: RX ERR\n \
                                    2: RX WRONGLAN\n \
                                    3: RX DUPLICATE\n \
                                    \n");
    scanf("%d", &funcSel);
```

```c
    switch(funcSel)
    {
    case 0:
        rxCounter = GetRxGoodOctets(slaveAddr, portSel);
        break;
    case 1:
        rxCounter = GetRxErrOctets(slaveAddr, portSel);
        break;
    case 2:
        rxCounter = GetRxWrongLan(slaveAddr, portSel);
        break;
    case 3:
        rxCounter = GetRxDuplicate(slaveAddr, portSel);
        break;

    default:
        break;

    }

    printf("rxCounter = 0x%x\n", rxCounter);

    return 0;
}

unsigned int GetRxGoodOctets(UINT8 slaveAddr, UINT8 portSel)
{
    unsigned int rxCount = 0;
    unsigned short rxCountLow = 0;
    unsigned short rxCountHigh = 0;
    unsigned short recvData = 0x0;

     /* Get low part of RX counter */
    GetRxCounter(slaveAddr, portSel, (short)CNT_CFG_REG_OFFSET_RX_GOOD_OCTETS_L,
&recvData);
    rxCountLow = ReverseBytes(recvData);

    /* Get high part of RX counter */
    GetRxCounter(slaveAddr, portSel, (short)CNT_CFG_REG_OFFSET_RX_GOOD_OCTETS_H,
&recvData);
    rxCountHigh = ReverseBytes(recvData);

    /* Combine low part and high part data */
    rxCount = (unsigned int)(rxCountHigh << 16) + rxCountLow;

    return rxCount;
}

unsigned int GetRxErrOctets(UINT8 slaveAddr, UINT8 portSel)
{
    unsigned int rxCount = 0;
    unsigned short rxCountLow = 0;
    unsigned short rxCountHigh = 0;
    unsigned short recvData = 0x0;
```

```
    /* Get low part of RX counter */
   GetRxCounter(slaveAddr, portSel, (short)CNT_CFG_REG_OFFSET_RX_ERR_L,
&recvData);
   rxCountLow = ReverseBytes(recvData);

   /* Get high part of RX counter */
   GetRxCounter(slaveAddr, portSel, (short)CNT_CFG_REG_OFFSET_RX_ERR_H,
&recvData);
   rxCountHigh = ReverseBytes(recvData);

   /* Combine low part and high part data */
   rxCount = (unsigned int)(rxCountHigh << 16) + rxCountLow;

    return rxCount;
}

unsigned int GetRxWrongLan(UINT8 slaveAddr, UINT8 portSel)
{
    unsigned int rxCount = 0;
    unsigned short rxCountLow = 0;
    unsigned short rxCountHigh = 0;
    unsigned short recvData = 0x0;

     /* Get low part of RX counter */
   GetRxCounter(slaveAddr, portSel, (short)CNT_CFG_REG_OFFSET_RX_WRONGLAN_L,
&recvData);
   rxCountLow = ReverseBytes(recvData);

   /* Get high part of RX counter */
   GetRxCounter(slaveAddr, portSel, (short)CNT_CFG_REG_OFFSET_RX_WRONGLAN_H,
&recvData);
   rxCountHigh = ReverseBytes(recvData);

   /* Combine low part and high part data */
   rxCount = (unsigned int)(rxCountHigh << 16) + rxCountLow;

    return rxCount;
}

unsigned int GetRxDuplicate(UINT8 slaveAddr, UINT8 portSel)
{
    unsigned int rxCount = 0;
    unsigned short rxCountLow = 0;
    unsigned short rxCountHigh = 0;
    unsigned short recvData = 0x0;

     /* Get low part of RX counter */
   GetRxCounter(slaveAddr, portSel, (short)CNT_CFG_REG_OFFSET_RX_DUPLICATE_L,
&recvData);
   rxCountLow = ReverseBytes(recvData);

   /* Get high part of RX counter */
   GetRxCounter(slaveAddr, portSel, (short)CNT_CFG_REG_OFFSET_RX_DUPLICATE_H,
&recvData);
   rxCountHigh = ReverseBytes(recvData);
```

```
    /* Combine low part and high part data */
    rxCount = (unsigned int)(rxCountHigh << 16) + rxCountLow;

    return rxCount;
}


unsigned short ReverseBytes(unsigned short value)
{
    return (unsigned short)((value & 0xFFU) << 8 | (value & 0xFF00U) >> 8);
}


bool CheckMxPrpService()
{
    bool isServiceExists = false;
    HANDLE hMutex;

    hMutex = CreateMutex(
        NULL, // default security descriptor
        FALSE, // mutex not owned
        TEXT("Global\\MxPrpSvcMutex")); // object name

    if (hMutex == NULL)
     {
        printf("CreateMutex error: %d\n", GetLastError() );
     }
    else
     {
       if ( GetLastError() == ERROR_ALREADY_EXISTS )
         {
            isServiceExists = true;
         }
       else
         {
            isServiceExists = false;

            /* No service is running, now we can close the mutex and run our program
*/
            CloseHandle(hMutex);
         }
     }

    return isServiceExists;
}
```

GetTxCounter

```
int GetTxCounter(UINT8 slaveAddr, UINT8 portSel, unsigned short counterOffset,
unsigned short *rxCounter);
```

Parameters

slaveAddr :   The smbus slave address.

portSel: Port select, interlink, port A or port B.

counterOffset: Counter address offset.

rxCounter: The RX counter value.

Return Value

Return value(ret) is zero on success, a negative error code on failure.

ret = 0 :   Success

Description

The GetTXCounter function is used to obtain the TX counter of the interlink port, port A and port B.

Example

```
Example#include "stdafx.h"
#include <windows.h>
#include "mxprp.h"

bool CheckMxPrpService();
unsigned short ReverseBytes(unsigned short value);
unsigned int GetTxOctets(UINT8 slaveAddr, UINT8 portSel);

/***************************************************************/
int _tmain(int argc, _TCHAR* argv[])
{
    UINT8 portSel = 0;
    UINT8 funcSel = 0;
    unsigned short counterOffset = 0;
    unsigned int txCount;
    int boardIndex = 0;
    UINT8 slaveAddr = 0x0;
    bool isServiceExist = false;

    isServiceExist = CheckMxPrpService();
    if(isServiceExist==true)
    {
        printf("mxPrpService is running, to run this program, please stop the
mxPrpService\n");
        printf("Exit the program\n");
        return 0;
    }
```

```
    /* Get user selection for smbus slave address */
    printf("Enter the boardIndex:\n");
    scanf("%d", &boardIndex);
    slaveAddr = SMBUS_SLAVE_ADDRESS;
    slaveAddr += boardIndex;

    /* Configure Counter Control bit to 1 to update the values in counter registers
*/
    printf("Select Port to configure: 2: Port0, 4: Port1, 8: Port2");
    scanf("%d", &portSel);

    /* Get TX count */
    txCount =GetTxOctets(slaveAddr, portSel);
    printf("txCount = %d\n", txCount);

    return 0;
}

unsigned int GetTxOctets(UINT8 slaveAddr, UINT8 portSel)
{
    unsigned int txCount = 0;
    unsigned short txCountLow = 0;
    unsigned short txCountHigh = 0;
    unsigned short recvData = 0x0;

     /* Get low part of TX counter */
    GetTxCounter(slaveAddr, portSel, (short)CNT_CFG_REG_OFFSET_TX_OCTETS_L,
&recvData);
    txCountLow = ReverseBytes(recvData);

    /* Get high part of TX counter */
    GetTxCounter(slaveAddr, portSel, (short)CNT_CFG_REG_OFFSET_TX_OCTETS_H,
&recvData);
    txCountHigh = ReverseBytes(recvData);

    /* Combine low part and high part data */
    txCount = (unsigned int)(txCountHigh << 16) + txCountLow;

    return txCount;
}

unsigned short ReverseBytes(unsigned short value)
{
    return (unsigned short)((value & 0xFFU) << 8 | (value & 0xFF00U) >> 8);
}

bool CheckMxPrpService()
{
    bool isServiceExists = false;
    HANDLE hMutex;

    hMutex = CreateMutex(
       NULL, // default security descriptor
       FALSE, // mutex not owned
       TEXT("Global\\MxPrpSvcMutex")); // object name
```

```
    if (hMutex == NULL)
     {
        printf("CreateMutex error: %d\n", GetLastError() );
     }
    else
     {
       if ( GetLastError() == ERROR_ALREADY_EXISTS )
         {
             isServiceExists = true;
         }
       else
         {
             isServiceExists = false;

             /* No service is running, now we can close the mutex and run our program
*/

             CloseHandle(hMutex);
         }
     }


    return isServiceExists;
}
```