LAN7800/LAN7800i Software User's Guide

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

ISBN: 978-1-5224-1168-0

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Object of Declaration: LAN7800/LAN7800i

EU Declaration of Conformity

This declaration of conformity is issued by the manufacturer.
The development/evaluation tool is designed to be used for research and development in a laboratory environment. This development/evaluation tool is not a Finished Appliance, nor is it intended for incorporation into Finished Appliances that are made commercially available as single functional units to end users under EU EMC Directive 2004/108/EC and as supported by the European Commission's Guide for the EMC Directive 2004/108/EC (8th February 2010).
This development/evaluation tool complies with EU RoHS2 Directive 2011/65/EU.
This development/evaluation tool, when incorporating wireless and radio-telecom functionality, is in compliance with the essential requirement and other relevant provisions of the R&TTE Directive 1999/5/EC and the FCC rules as stated in the declaration of conformity provided in the module datasheet and the module product page available at www.microchip.com.
For information regarding the exclusive, limited warranties applicable to Microchip products, please see Microchip's standard terms and conditions of sale, which are printed on our sales documentation and available at www.microchip.com.
Signed for and on behalf of Microchip Technology Inc. at Chandler, Arizona, USA.

Derek Carlson
VP Development Tools

11-NOV-16
Date

NOTES:

Table of Contents

Preface

NOTICE TO CUSTOMERS

All documentation becomes dated, and this manual is no exception. Microchip tools and documentation are constantly evolving to meet customer needs, so some actual dialogs and/or tool descriptions may differ from those in this document. Please refer to our web site (www.microchip.com) to obtain the latest documentation available.

Documents are identified with a "DS" number. This number is located on the bottom of each page, in front of the page number. The numbering convention for the DS number is "DSXXXXXA", where "XXXXX" is the document number and "A" is the revision level of the document.

For the most up-to-date information on development tools, see the MPLAB® IDE online help. Select the Help menu, and then Topics to open a list of available online help files.

INTRODUCTION

This chapter contains general information that will be useful to know before using the LAN7800. Items discussed in this chapter include:

- Document Layout
- Conventions Used in this Guide
- The Microchip Web Site
- Development Systems Customer Change Notification Service
- Customer Support
- Document Revision History

DOCUMENT LAYOUT

This document describes how to use the EVB-LAN7850 Evaluation Board as a development tool for the LAN7800, USB 3.0 to Gigabit Ethernet Controller.

- **Chapter 1. "Overview"** – Provides a brief description of the LAN7800/LAN7800i Software User's Guide.
- **Chapter 2. "Windows Vista 32/64-Bit Driver"** – This chapter details the installation and uninstallation of the Windows Vista 32/64-bit driver.
- **Chapter 3. "Windows 7 32/64-Bit Driver"** – This chapter details the installation and uninstallation of the Windows 7 32/64-bit driver.
- **Chapter 4. "Windows 8 32/64-Bit Driver"** – This chapter details the installation and uninstallation of the Windows 8 32/64-bit driver.
- **Chapter 5. "Windows 10 32/64-Bit Driver"** – This chapter details the installation and uninstallation of the Windows 10 32/64-bit driver.
- **Chapter 6. "MAC OS X Driver"** – This chapter details the installation and uninstallation of the Mac OS X driver.
- **Chapter 7. "Linux Driver"** – This chapter details the load and unload of the Linux driver.

- **Chapter 8. "UEFI Driver"** – This chapter details the installation and uninstallation of the UEFI driver.
- **Chapter 9. "Advanced Driver Parameters"** – This chapter provides an overview of how to access advanced parameters in various operating systems.
- **Chapter 10. "Legacy Pre-Execution Environment (PXE) Support"** – This chapter provides an overview of PXE support.
- **Chapter 11. "LAN78XX Command Line Utility"** – This chapter details how to use the command line utility.
- **Appendix A. "Customer Requirements"** – This appendix details customer requirements for device utilization.

CONVENTIONS USED IN THIS GUIDE

This manual uses the following documentation conventions:

DOCUMENTATION CONVENTIONS

Description	Represents	Examples
Arial font:		
Italic characters	Referenced books	*MPLAB® IDE User's Guide*
	Emphasized text	...is the *only* compiler...
Initial caps	A window	the Output window
	A dialog	the Settings dialog
	A menu selection	select Enable Programmer
Quotes	A field name in a window or dialog	"Save project before build"
Underlined, italic text with right angle bracket	A menu path	*File>Save*
Bold characters	A dialog button	Click **OK**
	A tab	Click the **Power** tab
N'Rnnnn	A number in verilog format, where N is the total number of digits, R is the radix and n is a digit.	4'b0010, 2'hF1
Text in angle brackets < >	A key on the keyboard	Press <Enter>, <F1>
Courier New font:		
Plain Courier New	Sample source code	`#define START`
	Filenames	`autoexec.bat`
	File paths	`c:\mcc18\h`
	Keywords	`_asm, _endasm, static`
	Command-line options	`-Opa+, -Opa-`
	Bit values	`0, 1`
	Constants	`0xFF, 'A'`
Italic Courier New	A variable argument	*`file`*`.o`, where *`file`* can be any valid filename
Square brackets [ ]	Optional arguments	`mcc18 [options]` *`file`* `[options]`
Curly brackets and pipe character: { | }	Choice of mutually exclusive arguments; an OR selection	`errorlevel {0|1}`
Ellipses...	Replaces repeated text	`var_name [, var_name...]`
	Represents code supplied by user	`void main (void) { ... }`

THE MICROCHIP WEB SITE

Microchip provides online support via our web site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

DEVELOPMENT SYSTEMS CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com, click on Customer Change Notification and follow the registration instructions.

The Development Systems product group categories are:

- **Compilers** – The latest information on Microchip C compilers, assemblers, linkers and other language tools. These include all MPLAB C compilers; all MPLAB assemblers (including MPASM assembler); all MPLAB linkers (including MPLINK object linker); and all MPLAB librarians (including MPLIB object librarian).
- **Emulators** – The latest information on Microchip in-circuit emulators.This includes the MPLAB REAL ICE and MPLAB ICE 2000 in-circuit emulators.
- **In-Circuit Debuggers** – The latest information on the Microchip in-circuit debuggers. This includes MPLAB ICD 3 in-circuit debuggers and PICkit 3 debug express.
- **MPLAB IDE** – The latest information on Microchip MPLAB IDE, the Windows Integrated Development Environment for development systems tools. This list is focused on the MPLAB IDE, MPLAB IDE Project Manager, MPLAB Editor and MPLAB SIM simulator, as well as general editing and debugging features.
- **Programmers** – The latest information on Microchip programmers. These include production programmers such as MPLAB REAL ICE in-circuit emulator, MPLAB ICD 3 in-circuit debugger and MPLAB PM3 device programmers. Also included are nonproduction development programmers such as PICSTART Plus and PIC-kit 2 and 3.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at:
http://www.microchip.com/support

DOCUMENT REVISION HISTORY

Revisions	Section/Figure/Entry	Correction
Rev. A (12-07-16)	Initial release.	

NOTES:

Chapter 1. Overview

1.1 INTRODUCTION

Note: Driver are shipped within current versions of the operating systems discussed in this document. The instructions here are for guidance when manually installing a driver.

This manual provides detailed instructions on the installation and uninstallation of LAN7800/LAN7800i software drivers under various operating systems:

- Windows Vista 32/64-Bit Driver
- Windows 7 32/64-Bit Driver
- Windows 8 32/64-Bit Driver
- Windows 10 32/64-Bit Driver
- MAC OS X Driver (10.10.x/10.11.x)
- Linux Driver (Kernel 3.13.0-29 or greater)
- UEFI Driver

Additional software related information is provided, including appendices for EEPROM and customer requirement related information:

- Advanced Driver Parameters
- Legacy Pre-Execution Environment (PXE) Support
- LAN78XX Command Line Utility
- Customer Requirements

The latest drivers and supporting documentation may be obtained by visiting the Microchip website:

http://www.microchip.com/

Note: The screen shots contained in this document are for illustration purposes only. The text contained therein may be different from what the user observes on his screen, due to driver or OS customization.

Note: Please refer to the respective software release notes for the latest information.

NOTES:

Chapter 2. Windows Vista 32/64-Bit Driver

This chapter details the installation and uninstallation of the Windows Vista 32/64-bit driver.

The Windows Vista 32/64-bit driver may be installed in two ways:

- **Windows Vista 32/64-Bit Driver Installation via EXE** (preferred method)
- **Windows Vista 32/64-Bit Driver Installation via INF**

Windows Vista 32/64-bit may be uninstalled in two ways:

- **Windows Vista 32/64 Bit Driver Uninstallation**
- **Windows Vista 32/64 Bit Driver Uninstallation via Control Panel**

2.1 WINDOWS VISTA 32/64-BIT DRIVER INSTALLATION VIA EXE

Copy the folder containing the distribution files to the desktop or any other convenient, known place within the directory structure. The folder contains the installer package and release notes file.

To install:

1. Click the installer icon.

   The setup window displays, as in Figure 2-1.

FIGURE 2-1: DEVICE INSTALLER INVOCATION

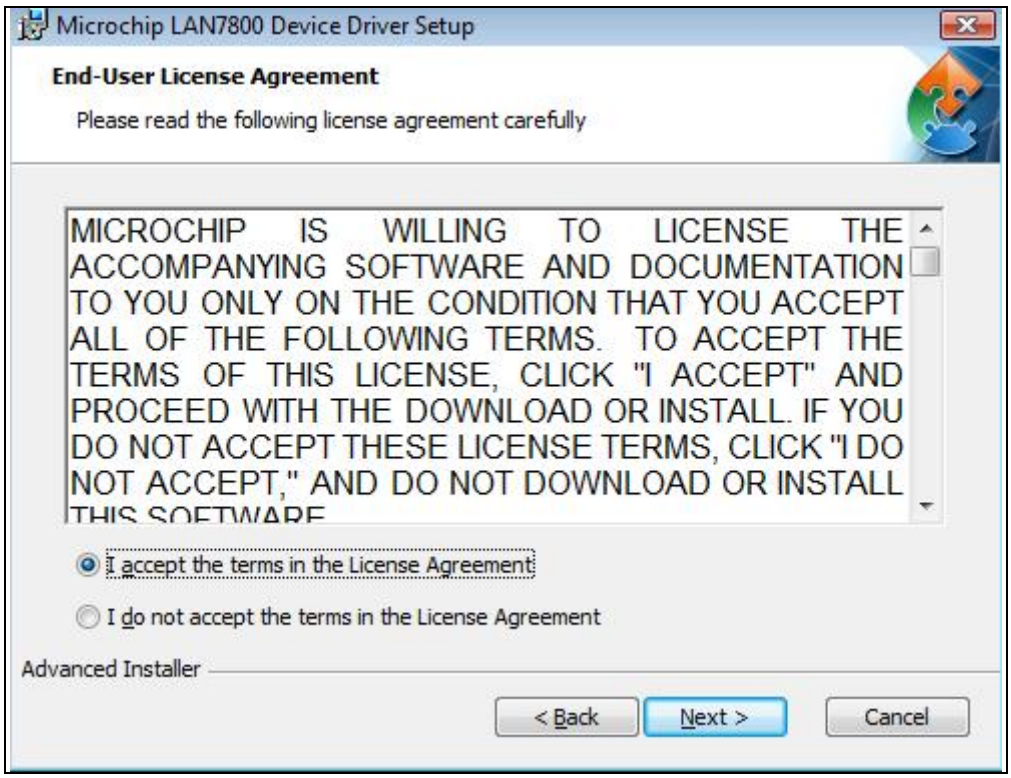
2. Click **Next >**.

   The End User License Agreement (EULA) displays.

3. Click the "I accept the terms in the License Agreement" radio button, as displayed in Figure 2-2.

   Once this has been selected, the **Next >** button is enabled, allowing the installation process to continue.

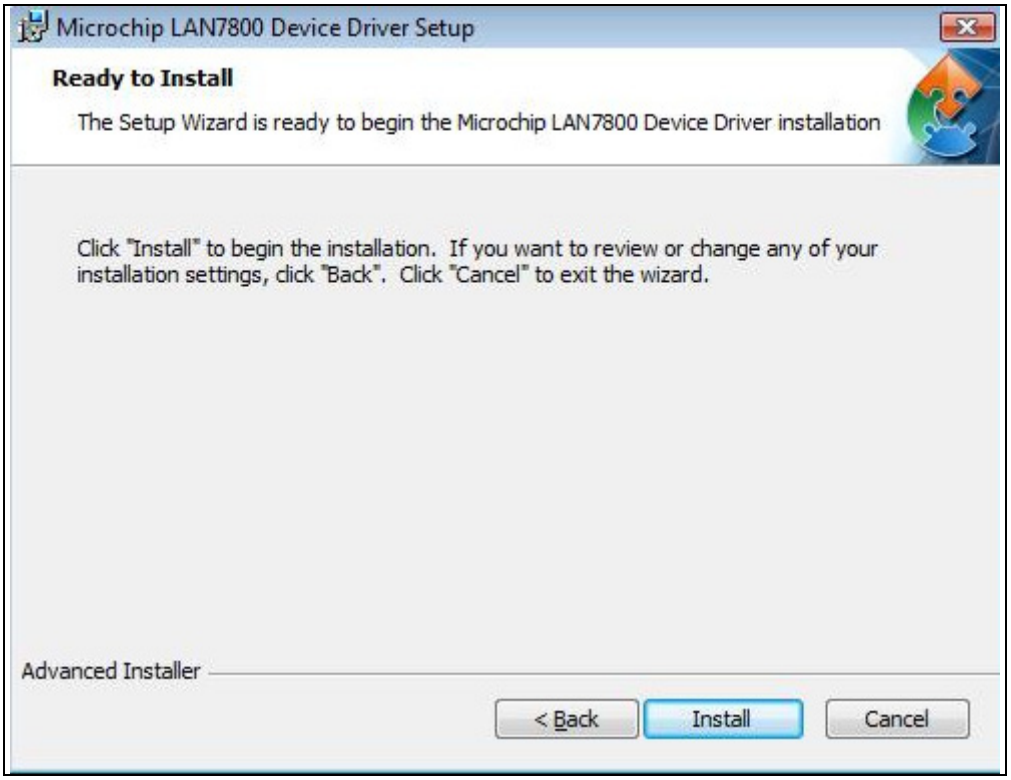
FIGURE 2-2: END USER LICENSE AGREEMENT



4. Click **Next >**.

   The Device Installer window prompts to either go back to review or change installation settings, proceed to install, or to cancel completely, as displayed in Figure 2-3.

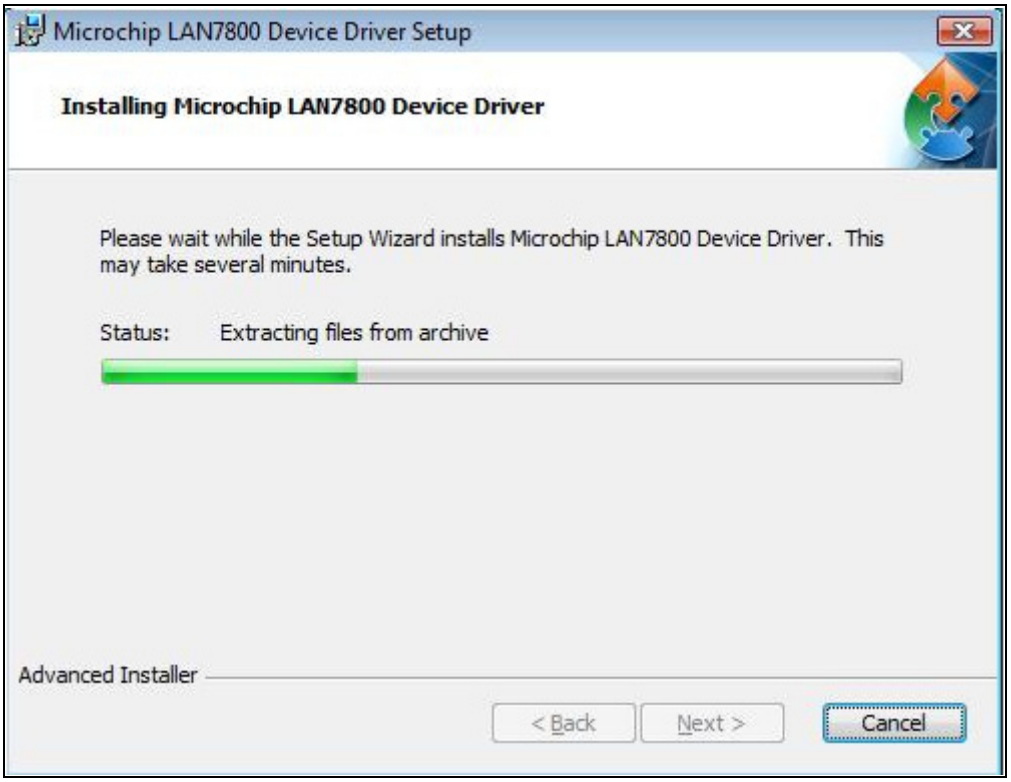
FIGURE 2-3: INSTALLATION WINDOW



5. Click **Install**.

   The Installation Progress window displays, as in Figure 2-4. The “Status” field displays the progress.

FIGURE 2-4: INSTALLATION PROGRESS WINDOW



When setup is done, the window displays a confirmation message, as displayed in Figure 2-5.

6. Click **Finish**.

FIGURE 2-5: DEVICE DRIVER INSTALLATION COMPLETE SCREEN - WINDOWS VISTA 32/64-BIT

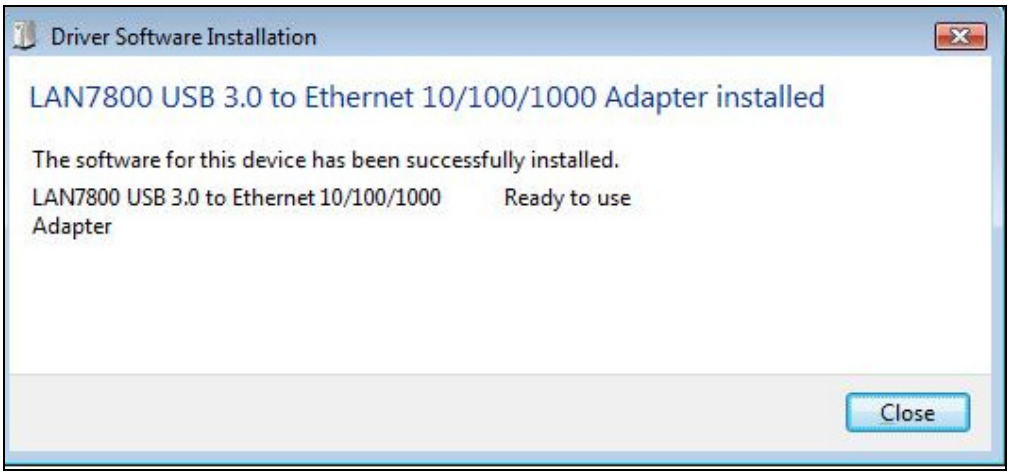
7. Plug the device into an available USB port on the computer.

   A balloon notification message displays in the task bar, as in Figure 2-6.

FIGURE 2-6: INSTALLING DEVICE DRIVER SOFTWARE TASK BAR NOTIFICATION

Once the software installation completes, the Driver Software Installation window displays, as in Figure 2-7.

FIGURE 2-7: DEVICE DRIVER SOFTWARE INSTALLED SUCCESSFULLY TASK BAR NOTIFICATION



The device installation is complete, and the device will be setup to have its IP address assigned by a DHCP server. This configuration may be changed to use a manually assigned IP address, using the device's Internet protocol properties found in Control Panel's Network Connections. Details are outside of the scope of this document.

2.2 WINDOWS VISTA 32/64-BIT DRIVER INSTALLATION VIA INF

Alternately, the device driver may be installed using a setup information file (`.inf`). This section details its installation method.

> **Note:** The method described in **Section 2.1 "Windows Vista 32/64-Bit Driver Installation via EXE"** is the preferred method of installation.

Copy the folder containing the Microchip INF distribution files to the desktop or any other convenient, known place within the directory structure.

To install:

1. Connect the device to an available USB port on the computer.

   The Found New Hardware window displays, as in Figure 2-8.

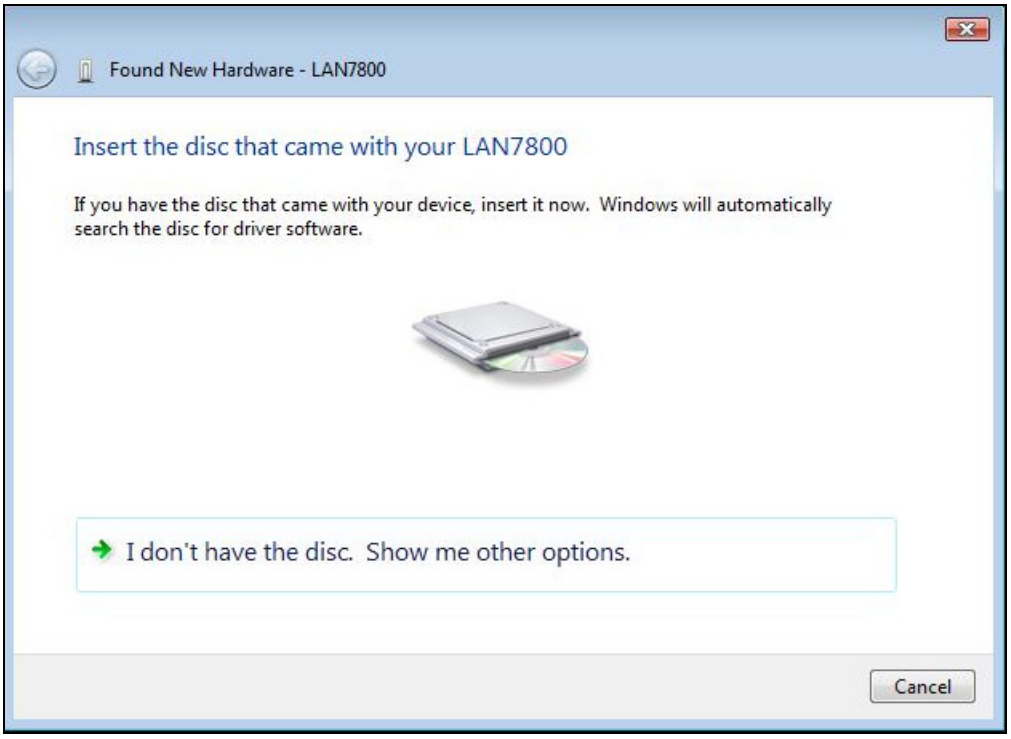
FIGURE 2-8: FOUND NEW HARDWARE WINDOW

2. Click **Locate and install driver software (recommended)**.

   A prompt asks for an installer disc.

3. Click **I don't have the disc. Show me other options.**, as displayed in Figure 2-9.

FIGURE 2-9: INSERT DISK WINDOW



4. Click **Next**.
5. Click **Browse my computer for driver software (advanced)**, as displayed in Figure 2-10.

FIGURE 2-10: DRIVER INSTALL OPTIONS WINDOW

6. Click **Browse…**.

The Browse For Folder dialog displays.

7. Browse to the location of the copied Microchip INF distribution files.
8. Click **OK**.
9. Click **Next**.

Note: Windows Vista installations must use the driver under the `ndis60` folder. For Windows 32-bit installations, browse to the `x86` folder within the `ndis60` folder of the Microchip INF distribution files, as displayed in Figure 2-11. For Windows 64-bit installations, browse to the `x64` folder within the `ndis60` folder of the Microchip INF distribution files, as displayed in Figure 2-12.

FIGURE 2-11: BROWSE WINDOW - NDIS60 32-BIT

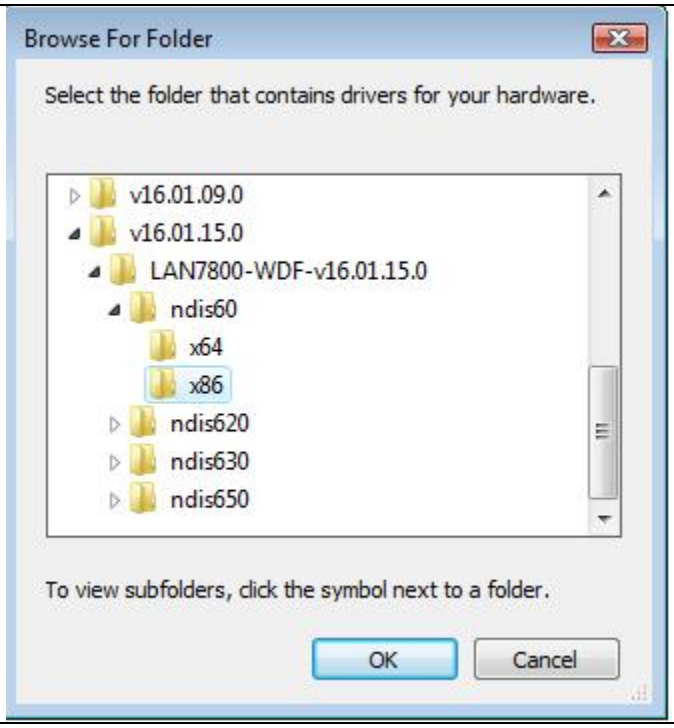
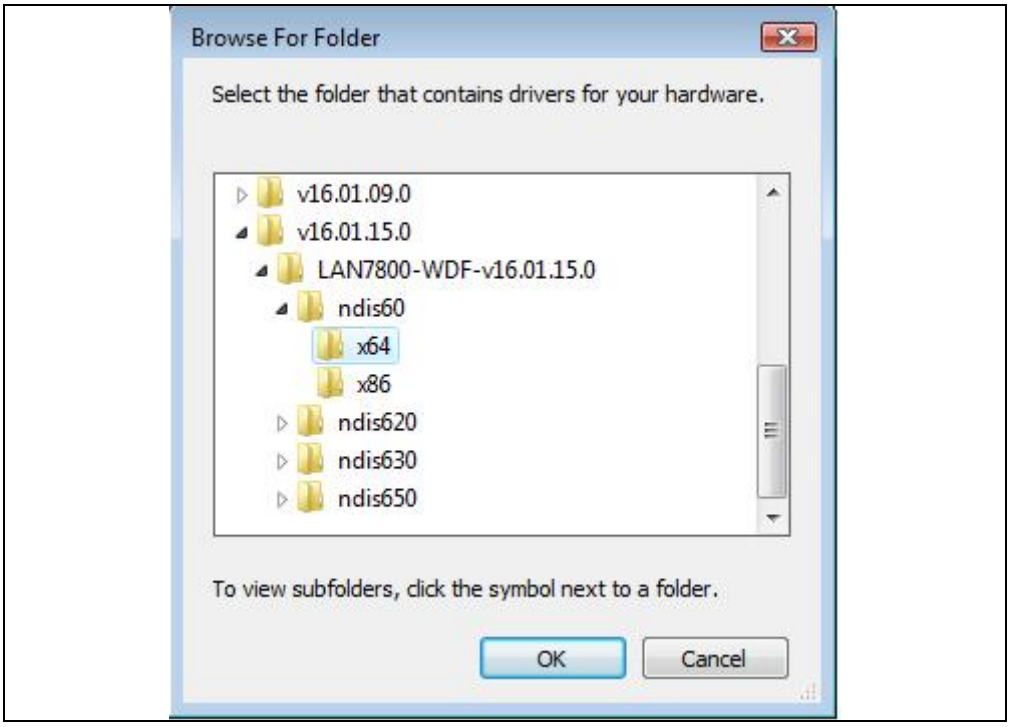
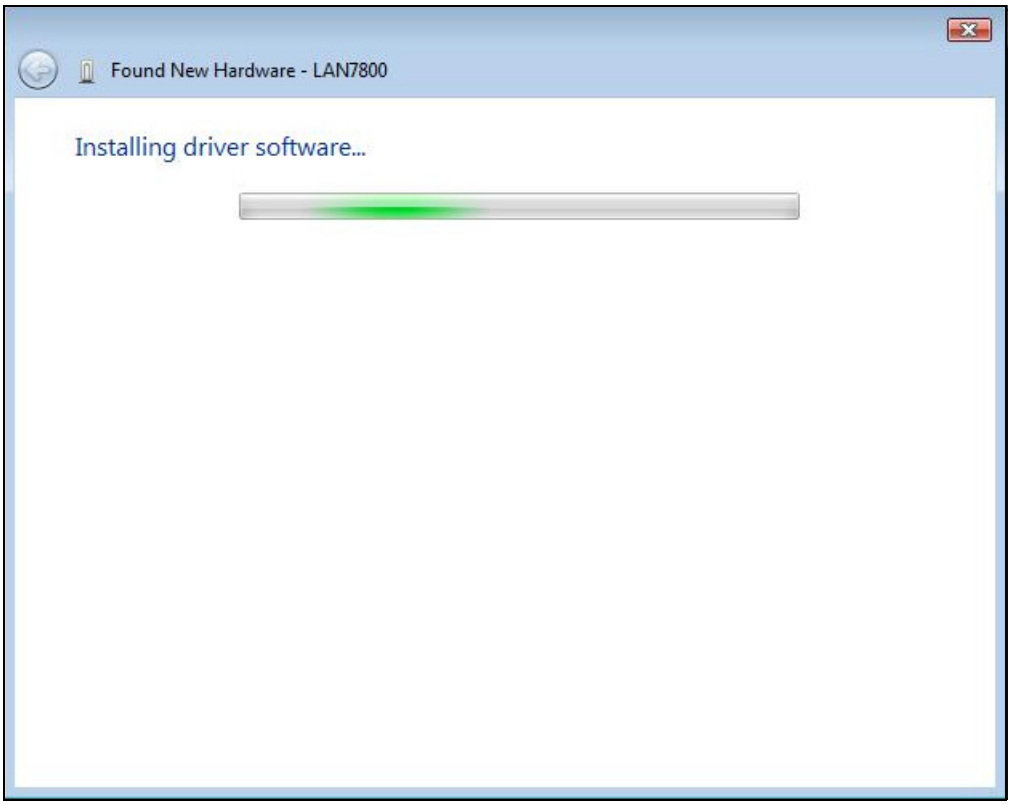


FIGURE 2-12: BROWSE WINDOW -NDIS60 64-BIT



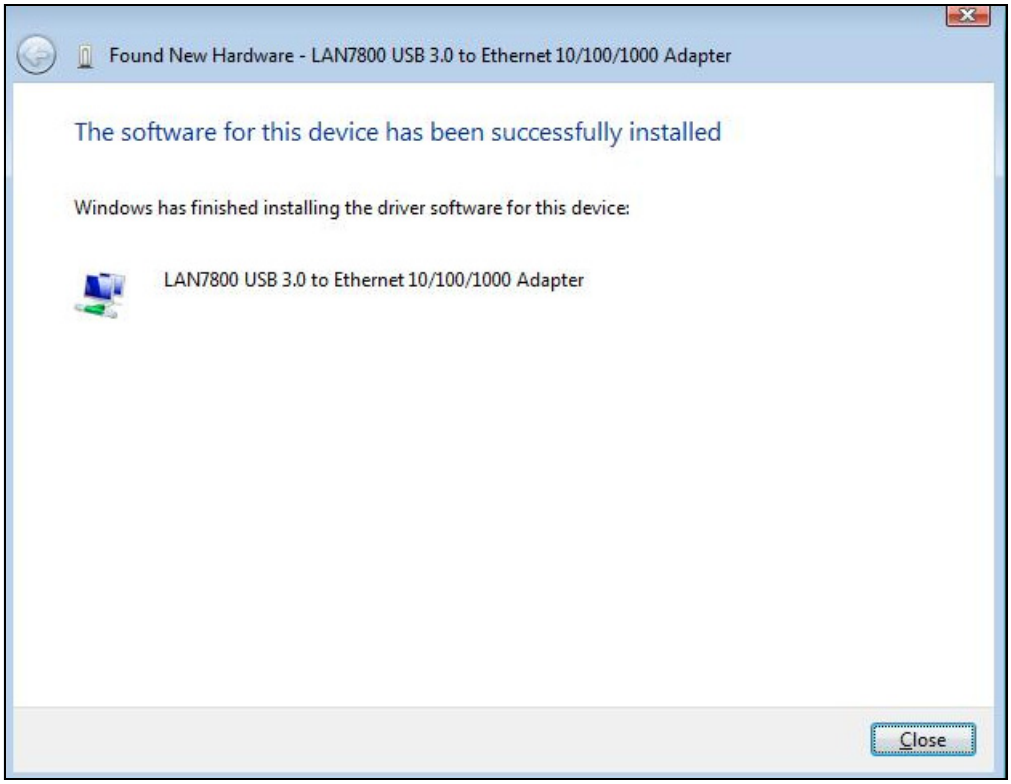
The installation progress displays, as in Figure 2-13. This indicates that some time may be necessary for the installation to complete.

FIGURE 2-13: INSTALLATION PROGRESS WINDOW



When finished, the installation progress window will change to indicate the driver has been installed, as displayed in Figure 2-14.

FIGURE 2-14: DEVICE DRIVER INSTALLATION COMPLETE SCREEN



10. Click **Close**.

    A device driver installation notification will pop-up as displayed in Figure 2-15, indicating the driver installation is complete.

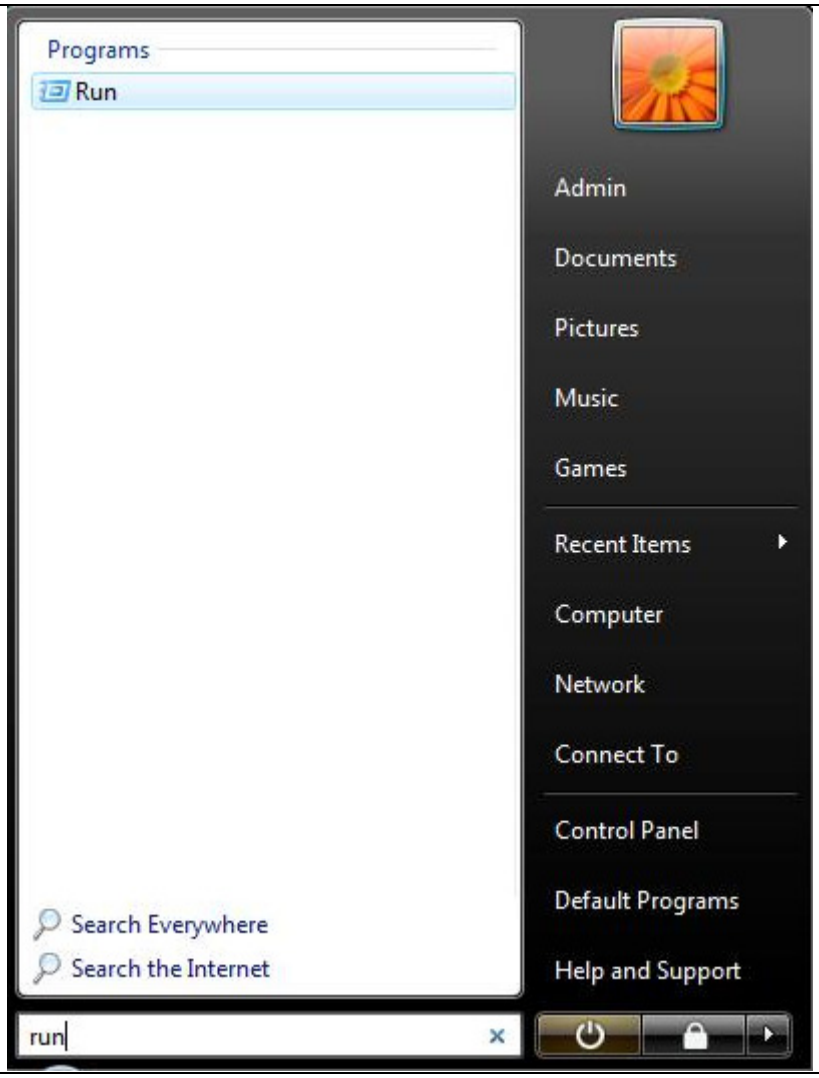
FIGURE 2-15: DEVICE DRIVER INSTALLATION SUCCESS TASK BAR NOTIFICATION

2.3 WINDOWS VISTA 32/64 BIT DRIVER UNINSTALLATION

To uninstall the Windows Vista 32/64-bit software:

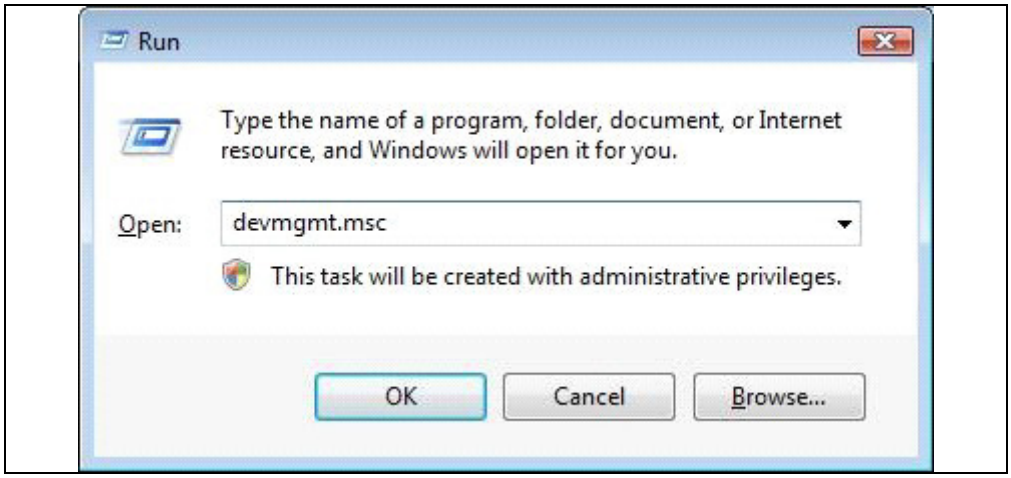
1. From the Start menu, type `Run` in the search field.
   Results from the Programs list display.
2. Click **Run**, as displayed in Figure 2-16.
   The Run dialog displays.

FIGURE 2-16: START MENU SEARCH



3. Type `devmgmt.msc`, as displayed in Figure 2-17.

FIGURE 2-17: RUN WINDOW



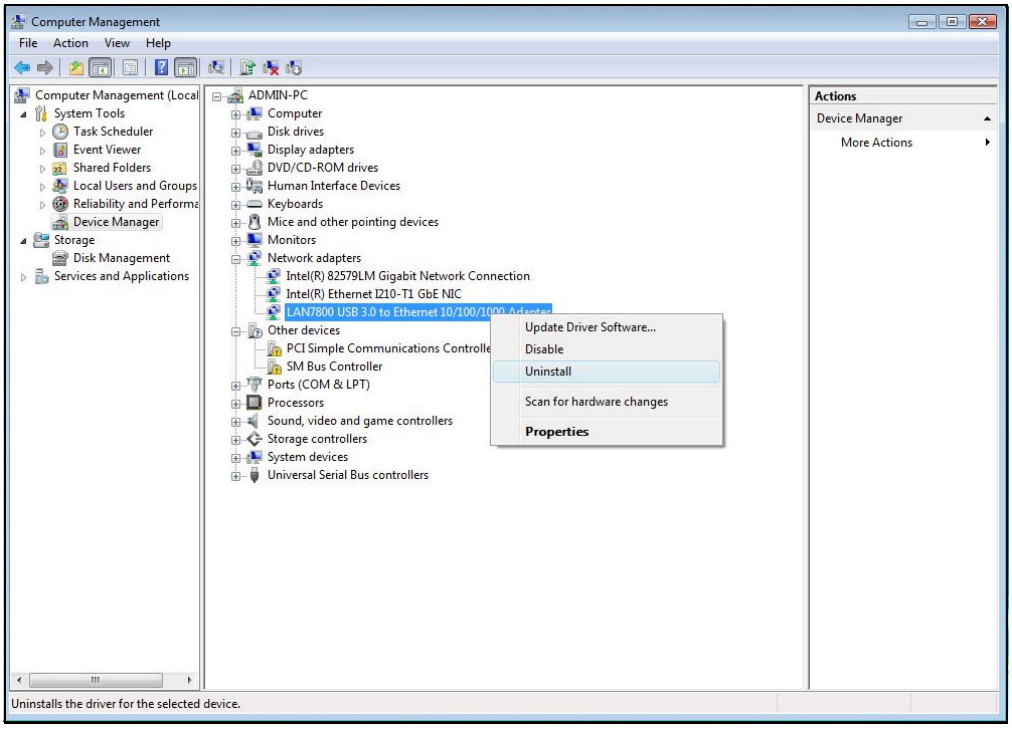
4. Click **OK**.

   The Device Manager window displays.

5. Select the device from the "Network adapters" section.
6. Right-click the device.

   A context menu displays, as in Figure 2-18.

FIGURE 2-18: DEVICE MANAGER WINDOW



7. Click Uninstall.

   The Confirm Device Uninstall window displays

8. Select the "Delete the driver software for this device." checkbox, as displayed in Figure 2-19.

FIGURE 2-19: CONFIRM DEVICE REMOVAL WINDOW

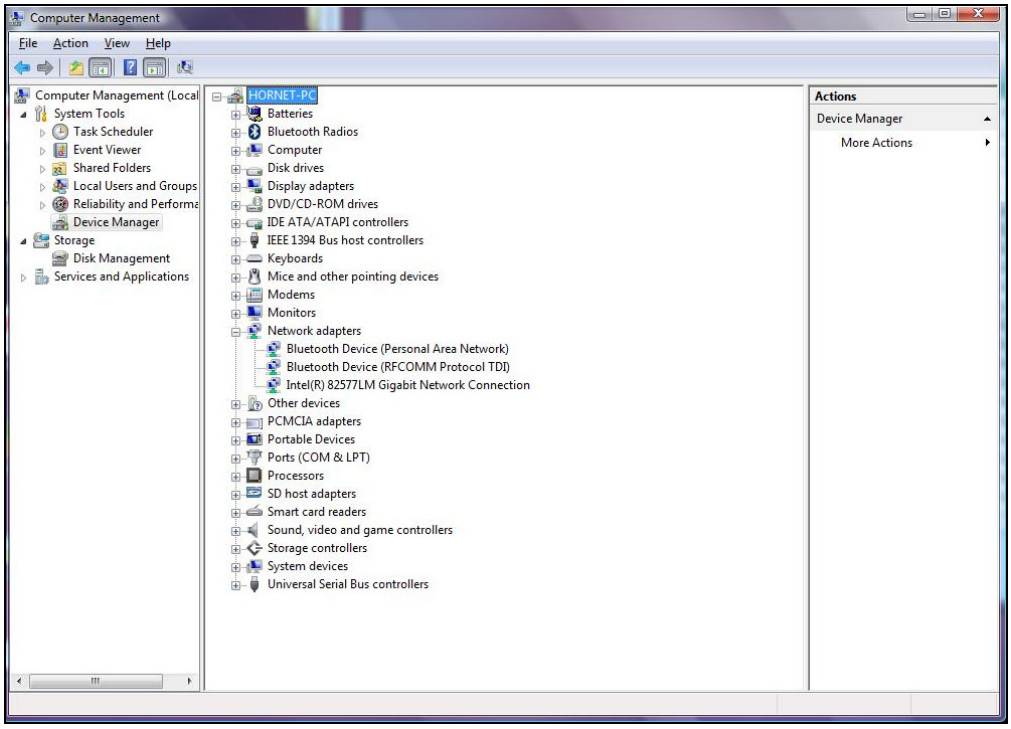
9. Click **OK**.

   The Confirm Device Uninstall window displays a progress indicator as the device drivers are being removed, as displayed in Figure 2-20.

FIGURE 2-20: DEVICE UNINSTALL PROGRESS WINDOW

The device driver is no longer listed in the Device Manager window, as displayed in Figure 2-21. The uninstallation process is now complete.

FIGURE 2-21: DEVICE REMOVED FROM DEVICE MANAGER



2.4 WINDOWS VISTA 32/64 BIT DRIVER UNINSTALLATION VIA CONTROL PANEL

This section details how to use the automated uninstaller using Windows Vista's “Programs and Features”.

To uninstall the Windows Vista 32/64-bit software:

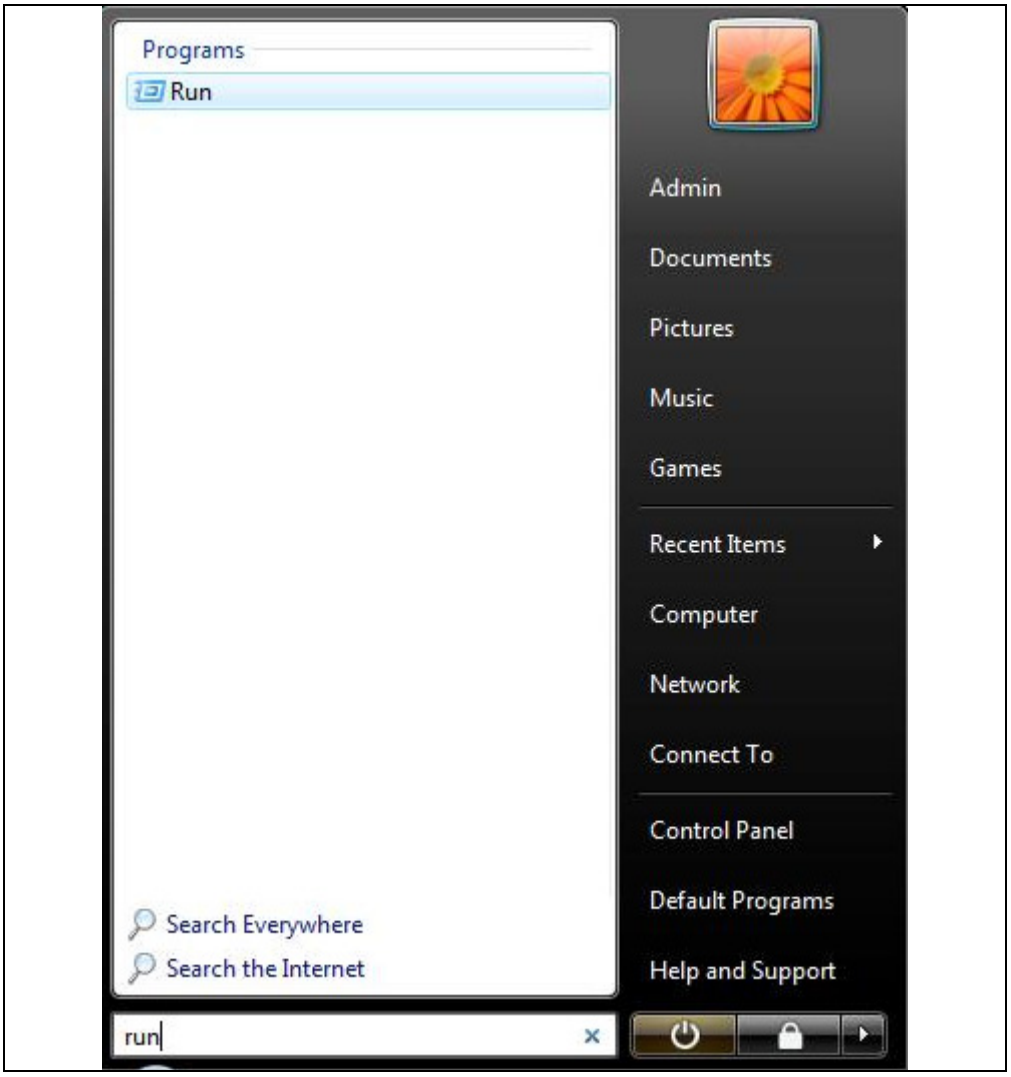
1. From the Start menu, type `Run` in the search field.

   Results from the Programs list display.

2. Click **Run**, as displayed in Figure 2-22.

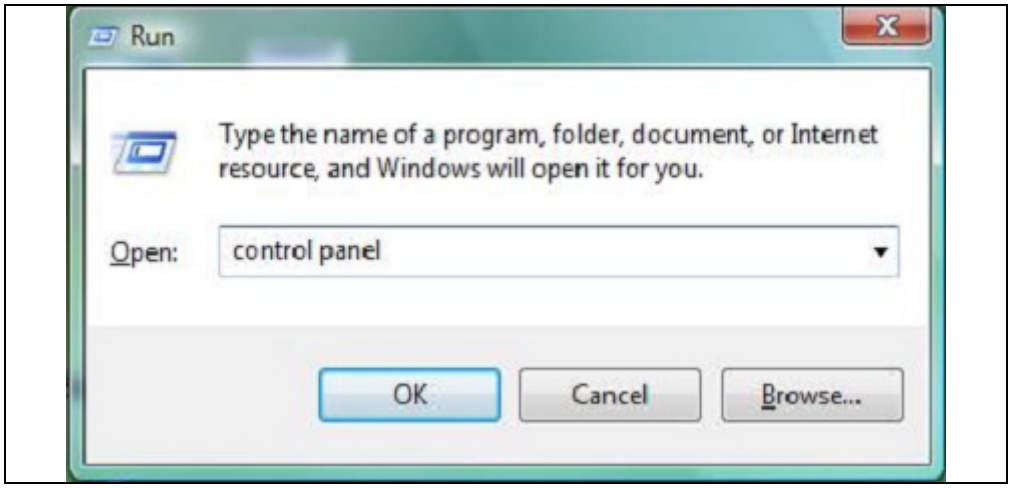
   The Run dialog displays.

FIGURE 2-22: START MENU SEARCH



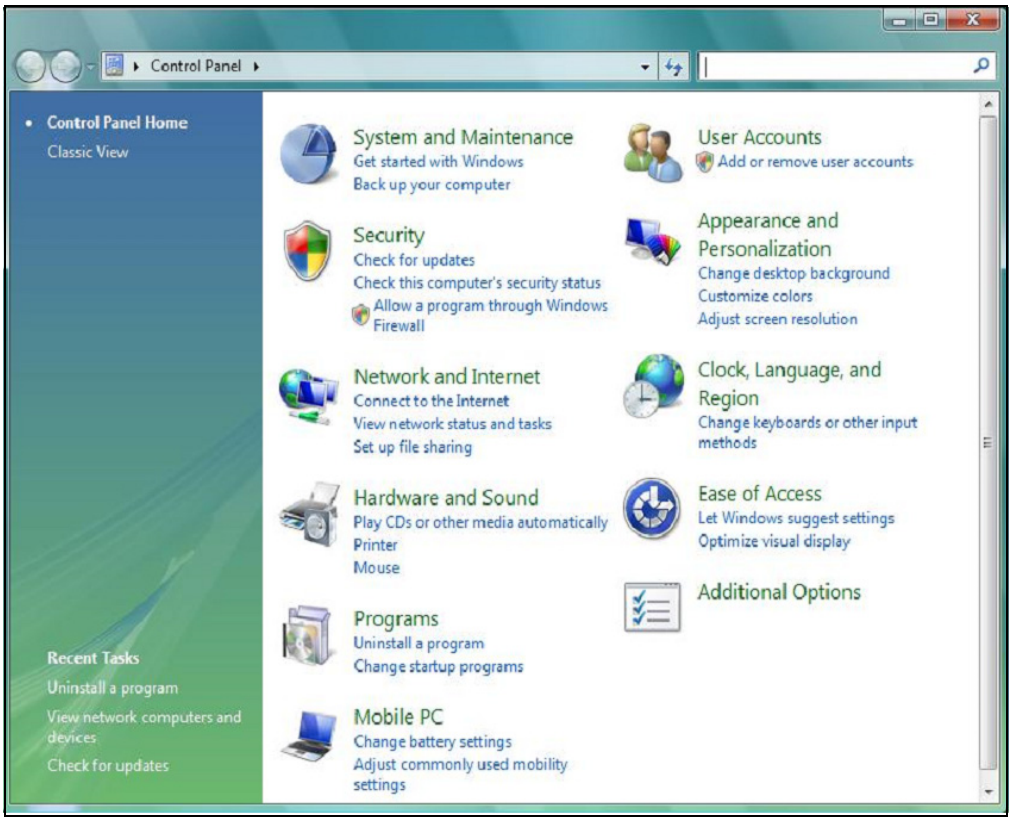
3. Type `control panel`, as displayed in Figure 2-23.

FIGURE 2-23: RUN WINDOW



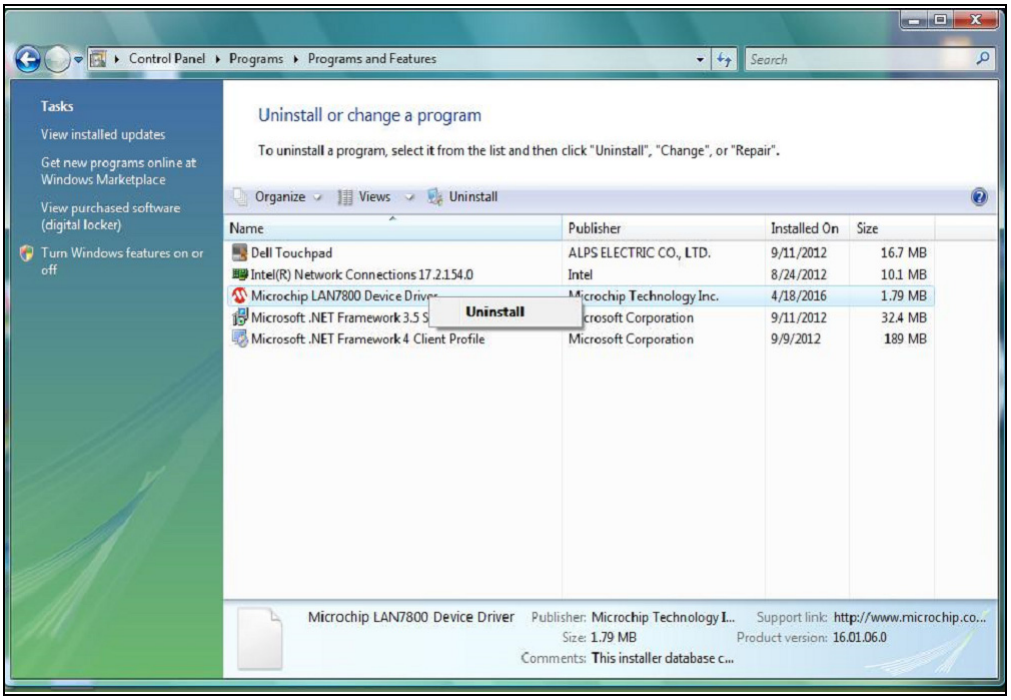
The Control Panel window displays, as in Figure 2-24.

FIGURE 2-24: CONTROL PANEL WINDOW



4. Click **Uninstall a program**.
   The Programs and Features window displays.
5. Right-click Microchip LAN7800 Device Driver.
   A context menu displays, as displayed in Figure 2-25.

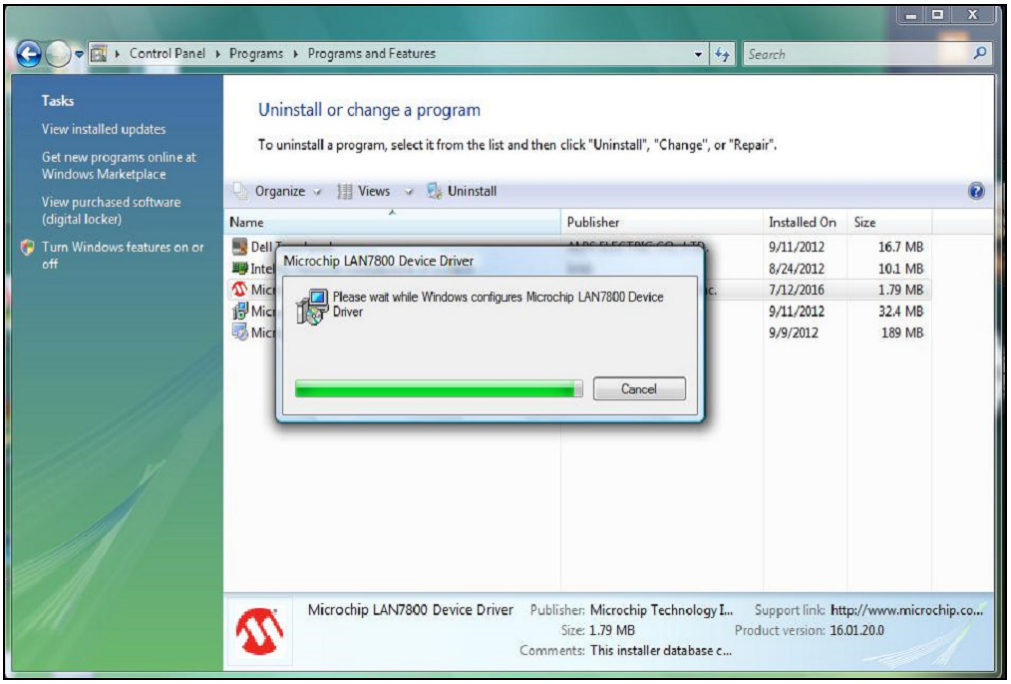
FIGURE 2-25: DRIVER UNINSTALL WINDOW



6. Click **Uninstall**.

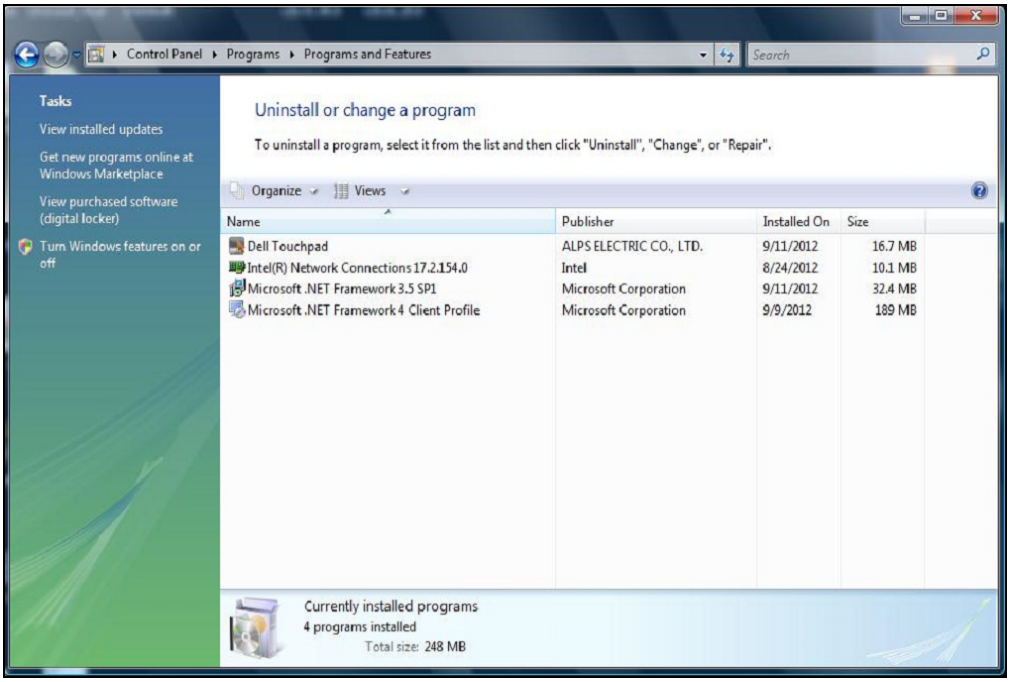
   The LAN7800 driver uninstalls, as displayed in Figure 2-26.

FIGURE 2-26: DRIVER UNINSTALL PROGRESS WINDOW



The LAN7800 driver Control Panel entry is removed, as displayed in Figure 2-27.

FIGURE 2-27: DRIVER REMOVED FROM CONTROL PANEL WINDOW



Chapter 3. Windows 7 32/64-Bit Driver

This chapter details the installation and uninstallation of the Windows 7 32/64-bit driver.

The Windows 7 32/64-bit driver may be installed in two ways:

- **Windows 7 32/64-Bit Driver Installation via EXE** (preferred method)
- **Windows 7 32/64-Bit Driver Installation via INF**

Windows 7 32/64-bit driver may be uninstalled in two ways:

- **Windows 7 32/64 Bit Driver Uninstallation via Device Manager**.
- **Windows 7 32/64 Bit Driver Uninstallation via Control Panel**.

3.1 WINDOWS 7 32/64-BIT DRIVER INSTALLATION VIA EXE

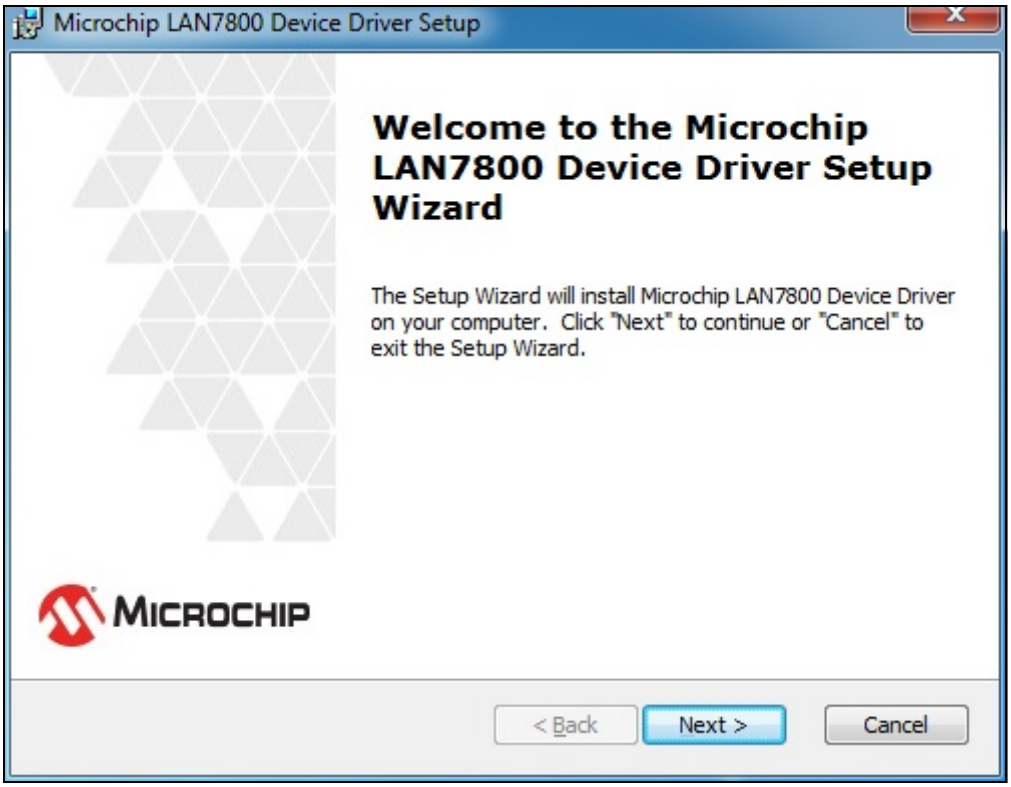
Copy the folder containing the distribution files to the desktop or any other convenient, known place within the directory structure. The folder contains the installer package and release notes file.

To install:

1. Click the installer icon.

   The setup window displays, as in Figure 3-1.

FIGURE 3-1: DEVICE INSTALLER INVOCATION



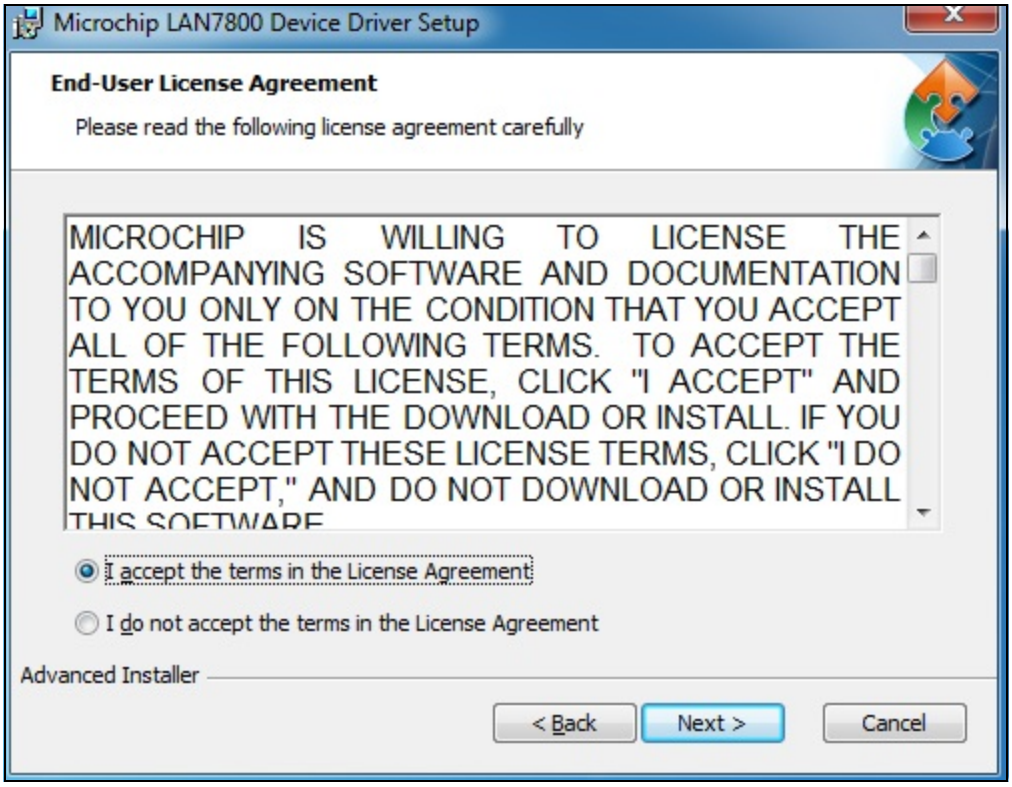
2. Click **Next >**.

   The End User License Agreement (EULA) displays.

3. Click the "I accept the terms in the License Agreement" radio button, as displayed in Figure 3-2.

   Once this has been selected, the **Next >** button is enabled, allowing the installation process to continue.

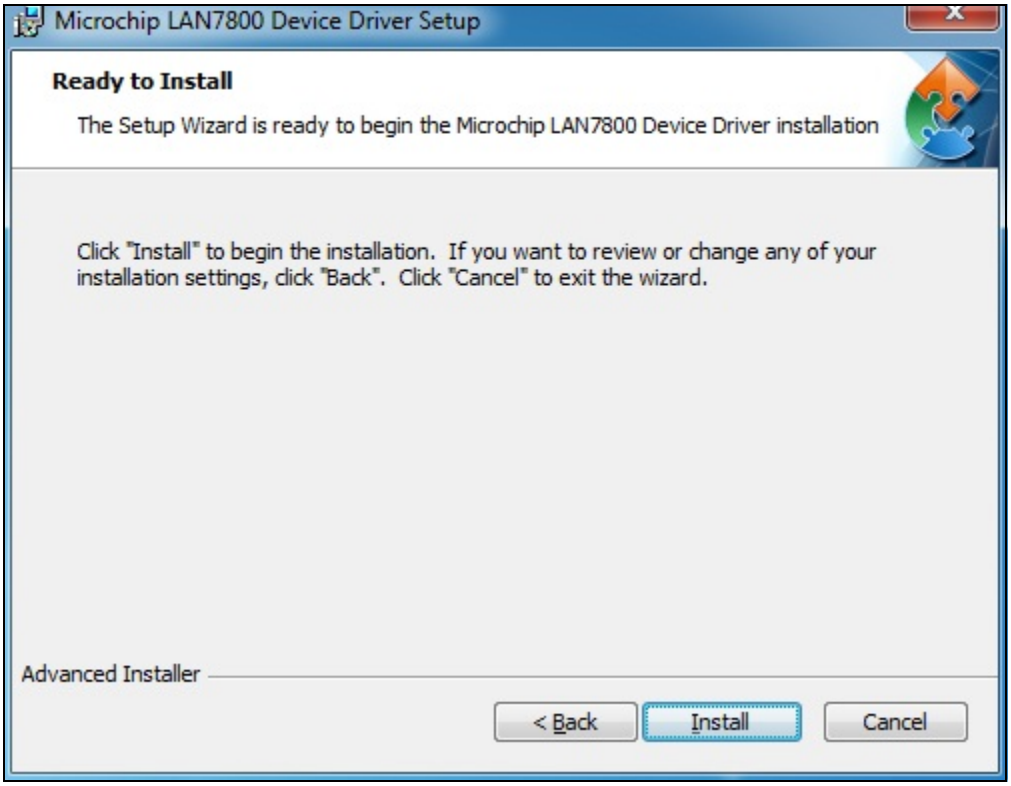
FIGURE 3-2: END USER LICENSE AGREEMENT



4. Click **Next >**.

   The Device Installer window prompts to either go back to review or change installation settings, proceed to install, or to cancel completely, as displayed in Figure 3-3.

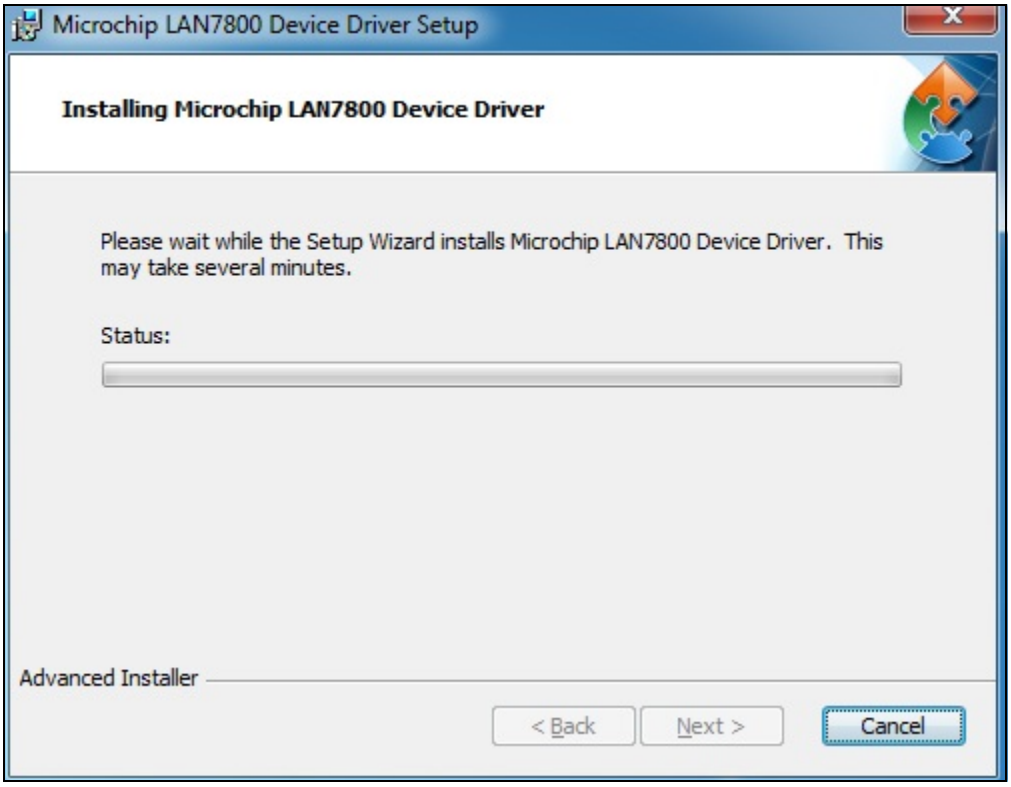
FIGURE 3-3: INSTALLATION WINDOW



5. Click **Install**.

   The Installation Progress window displays, as in Figure 3-4. The “Status” field displays the progress.

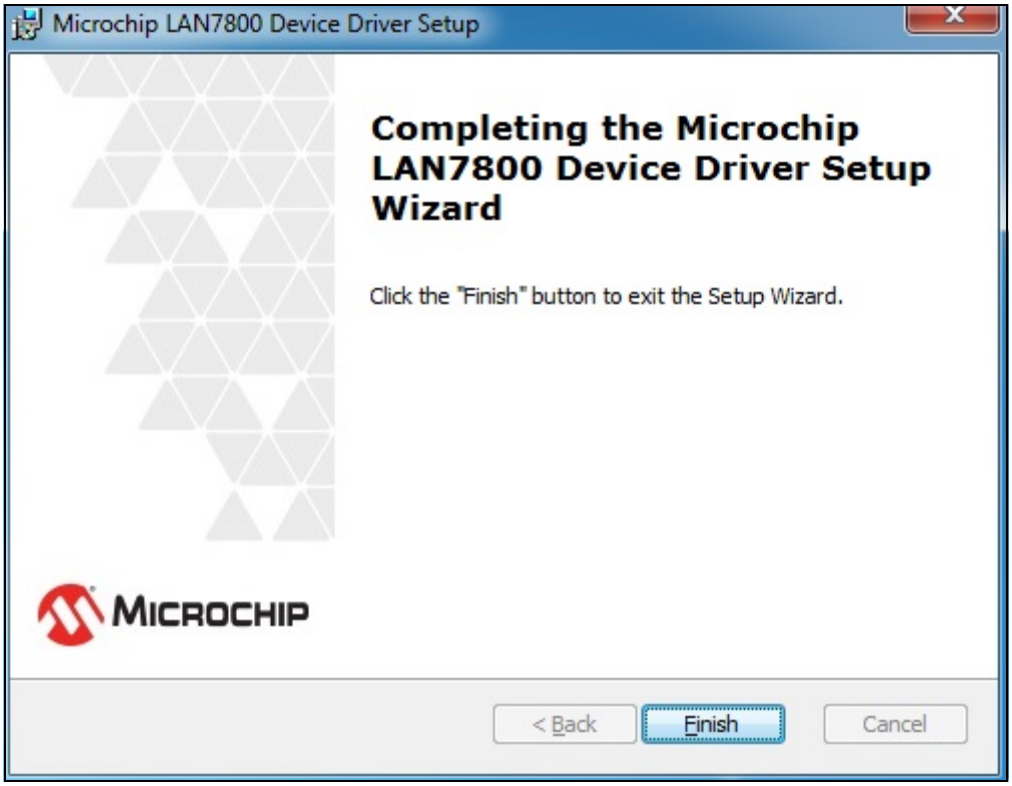
FIGURE 3-4: INSTALLATION PROGRESS WINDOW



When setup is done, the window displays a confirmation message, as displayed in Figure 3-5.

6. Click **Finish**.

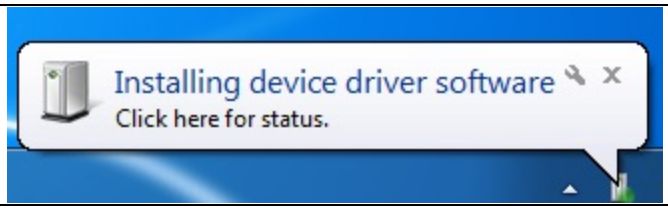
FIGURE 3-5: DEVICE DRIVER INSTALLATION COMPLETE SCREEN - WINDOWS 7 32/64-BIT



7. Plug the device into an available USB port on the computer.

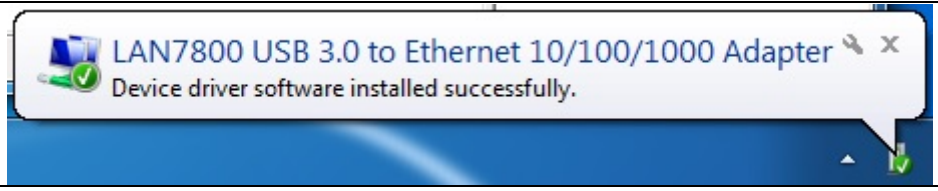
   A balloon notification message displays in the task bar, as in Figure 3-6.

FIGURE 3-6: INSTALLING DEVICE DRIVER SOFTWARE TASK BAR NOTIFICATION



Once the software installation completes, the Driver Software Installation window displays, as in Figure 3-7.

FIGURE 3-7: DEVICE DRIVER SOFTWARE INSTALLED SUCCESSFULLY TASK BAR NOTIFICATION



The device installation is complete, and the device will be setup to have its IP address assigned by a DHCP server. This configuration may be changed to use a manually assigned IP address, using the device's Internet protocol properties found in Control Panel's Network Connections. Details are outside of the scope of this document.

3.2 WINDOWS 7 32/64-BIT DRIVER INSTALLATION VIA INF

Alternately, the device driver may be installed using a setup information file (`.inf`). This section details its installation method.

> **Note:** The method described in **Section 3.1 "Windows 7 32/64-Bit Driver Installation via EXE"** is the preferred method of installation.

Copy the folder containing the Microchip INF distribution files to the desktop or any other convenient, known place within the directory structure.

To install:

1. Connect the device to an available USB port on the computer.

   A balloon notification displays, as in Figure 3-8.

FIGURE 3-8: DEVICE DRIVER NOT INSTALLED TASK BAR NOTIFICATION

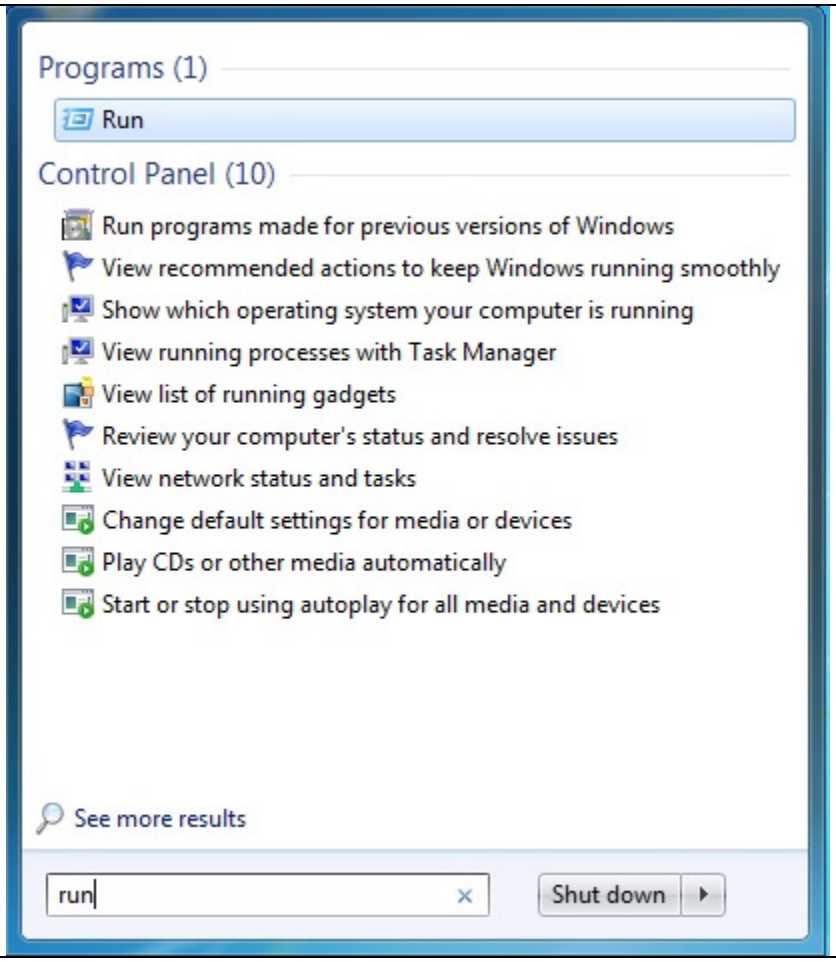
2. From the Start menu, type `Run` in the search field.

   Results from the Programs list display.

3. Click **Run**, as displayed in Figure 3-9.

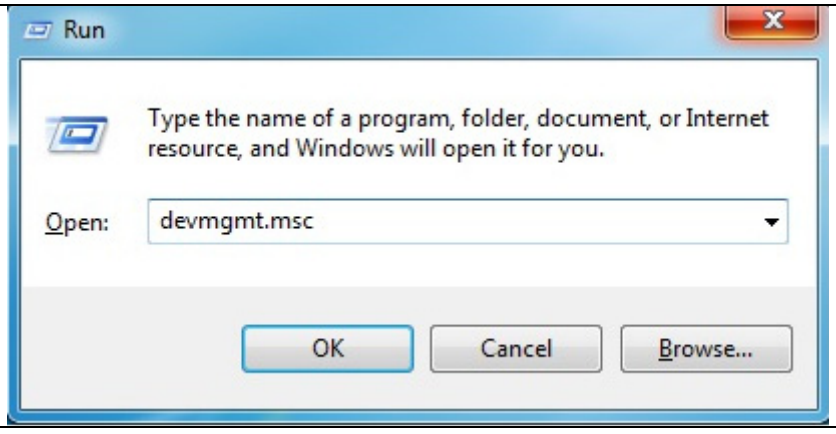
   The Run dialog displays.

FIGURE 3-9: START MENU SEARCH



4. Type `devmgmt.msc`, as displayed in Figure 3-10.

FIGURE 3-10: RUN WINDOW

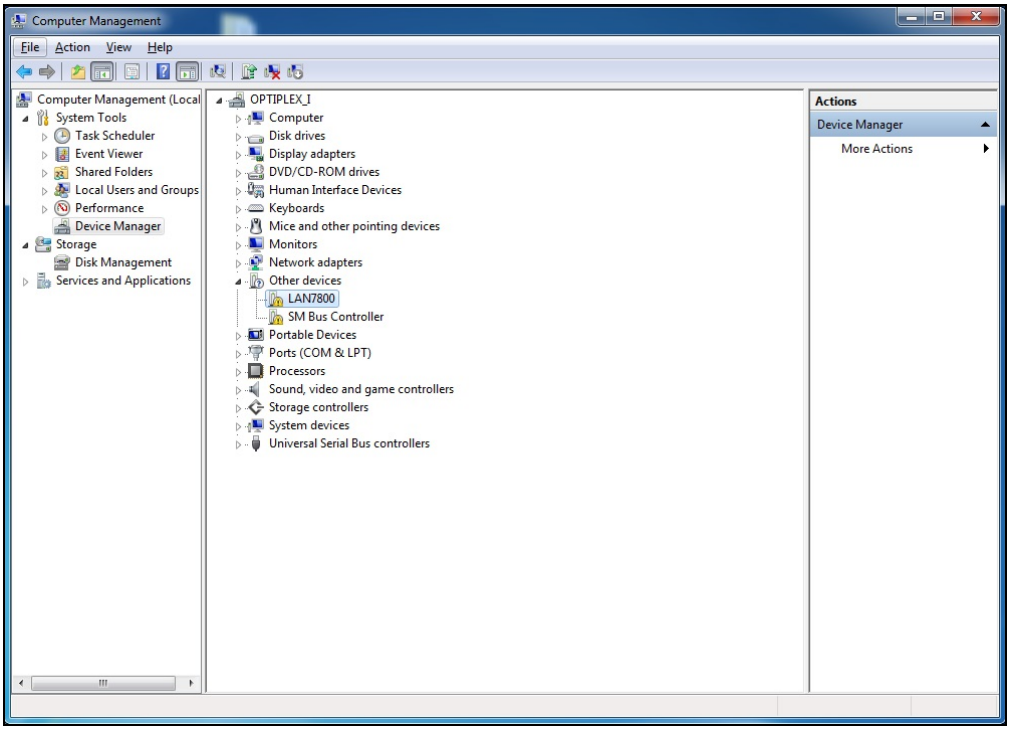


5. Click **OK**.

   The Device Manager window displays.

6. Select the device from the "Other devices" section, as displayed in Figure 3-11.

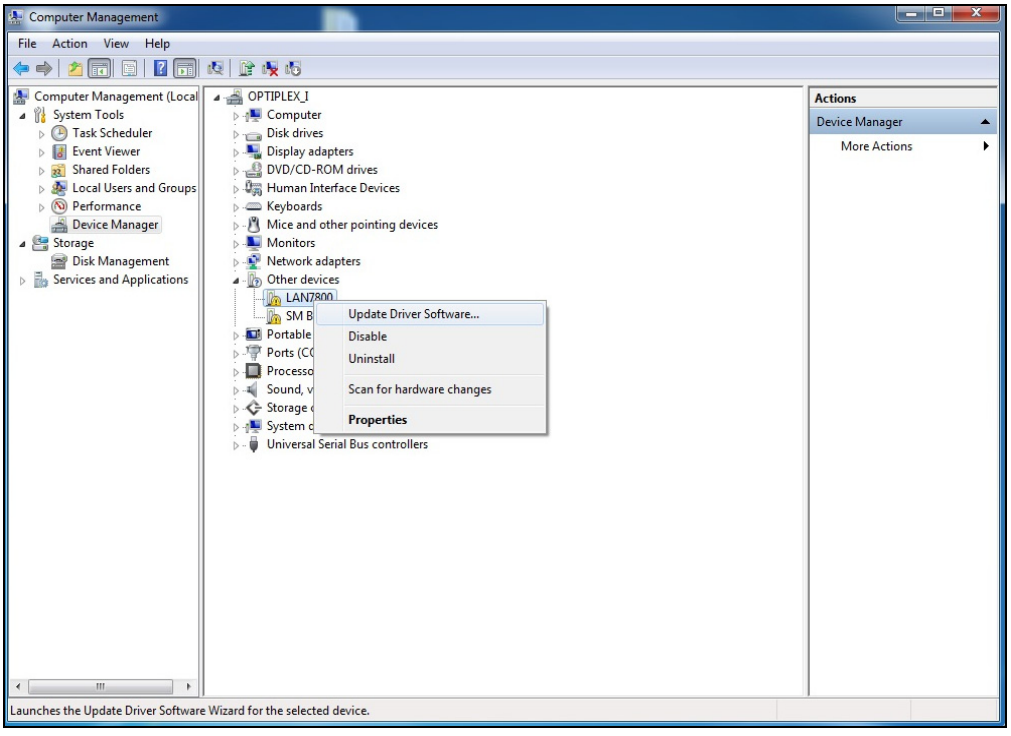
FIGURE 3-11: DEVICE MANAGER WINDOW - OTHER DEVICES



7. Right-click the device.

   A context menu displays, as in Figure 3-12.

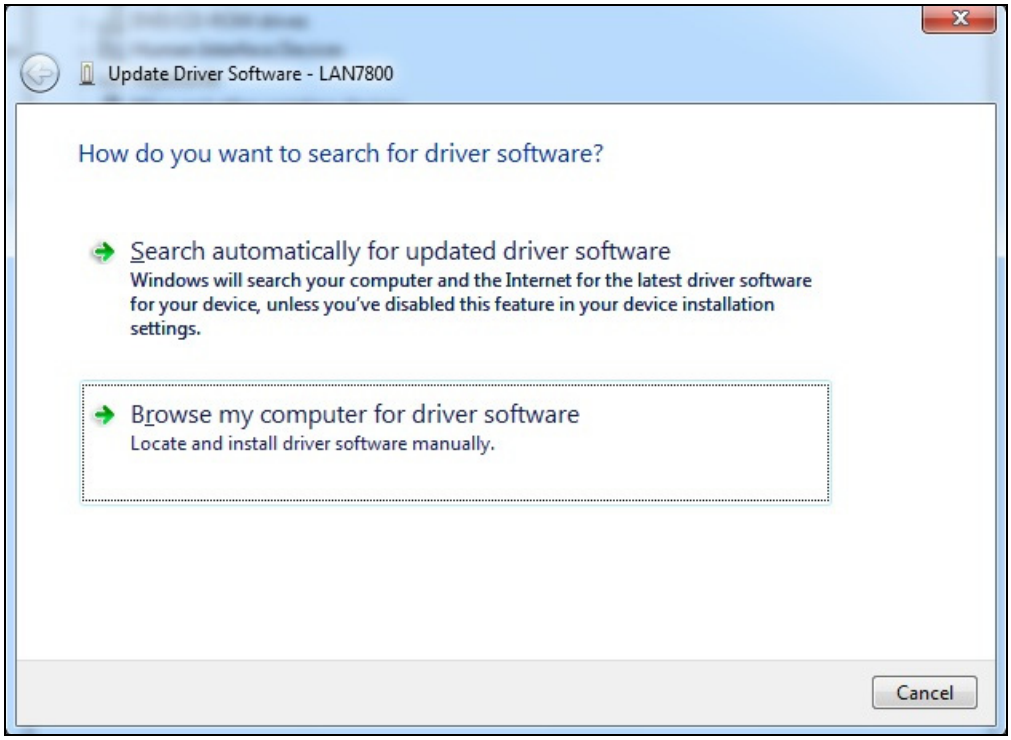
FIGURE 3-12: DEVICE MANAGER WINDOW - UPDATE DRIVER SOFTWARE



8. Click Update Driver Software.

   The Update Driver Software window displays, as in Figure 3-13.

FIGURE 3-13: DRIVER INSTALL OPTIONS WINDOW



9. Click **Browse my computer for driver software**.
10. Click **Browse…**.

    The Browse For Folder dialog displays.
11. Browse to the location of the copied Microchip INF distribution files.
12. Click **OK**.
13. Click **Next**.

Note: Windows 7 installations must use the driver under the `ndis620` folder. For Windows 32-bit installations, browse to the `x86` folder within the `ndis620` folder of the microchip INF distribution files, as displayed in Figure 3-14. For Windows 64-bit installations, browse to the `x64` folder within the `ndis620` folder of the Microchip INF distribution files, as displayed in Figure 3-15.

FIGURE 3-14: BROWSE WINDOW - WINDOWS 7 32-BIT

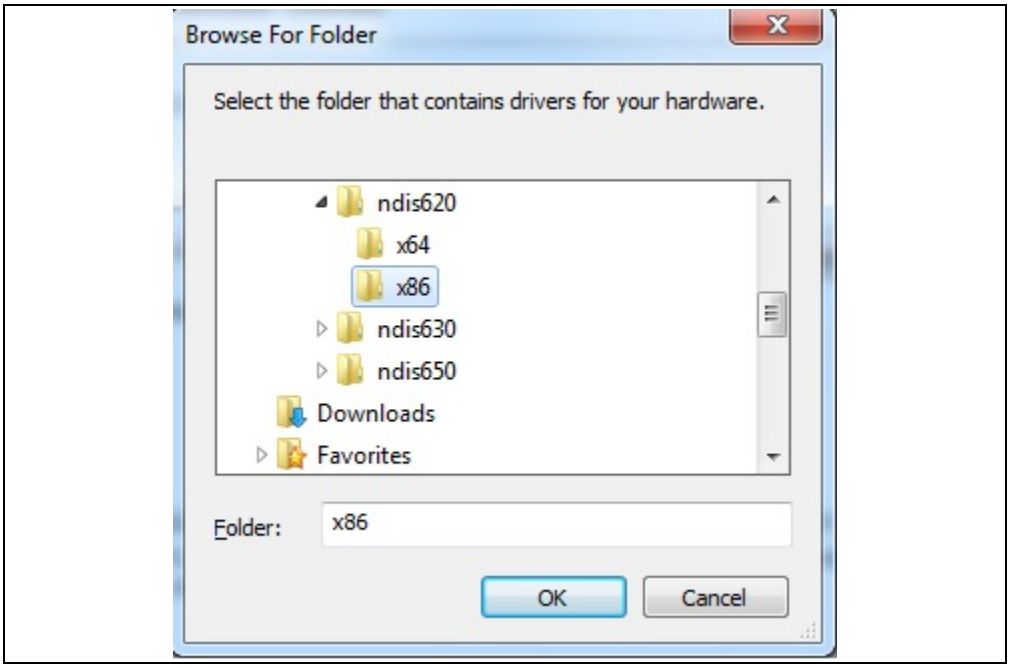
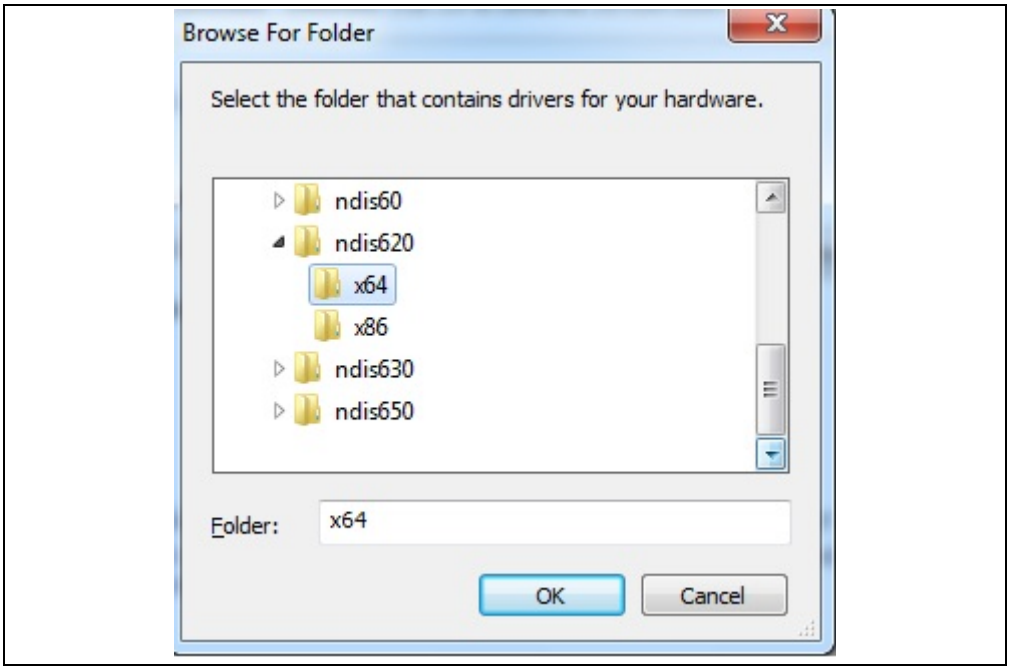
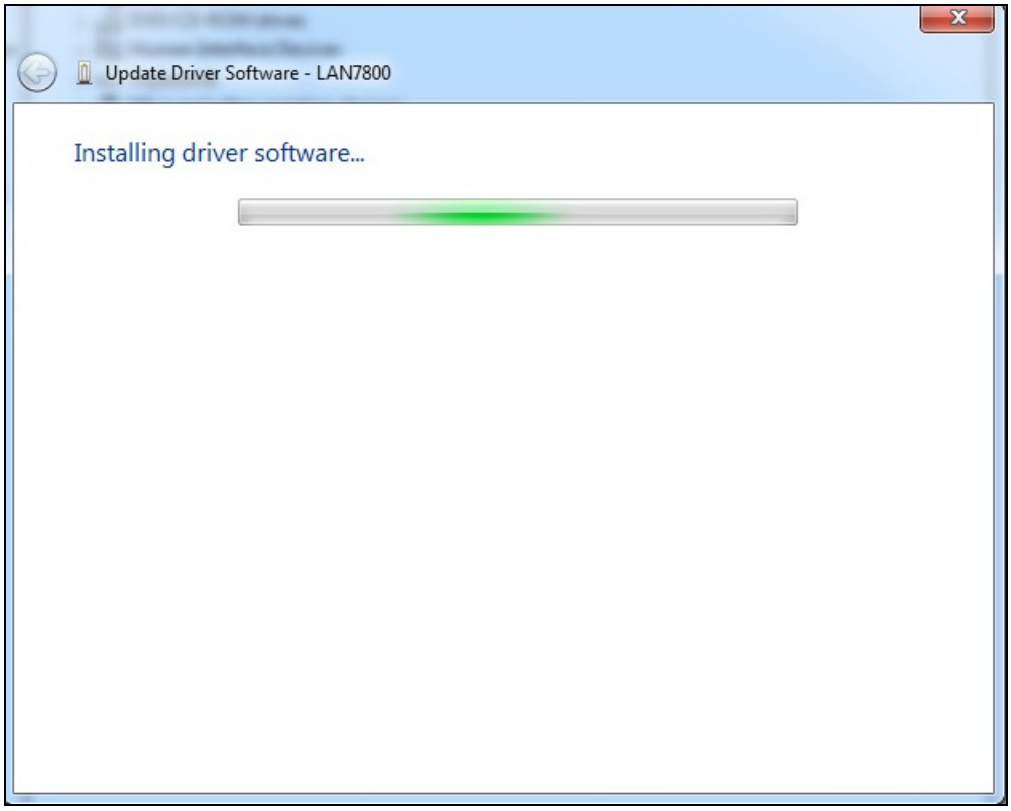


FIGURE 3-15: BROWSE WINDOW - WINDOWS 7 64-BIT



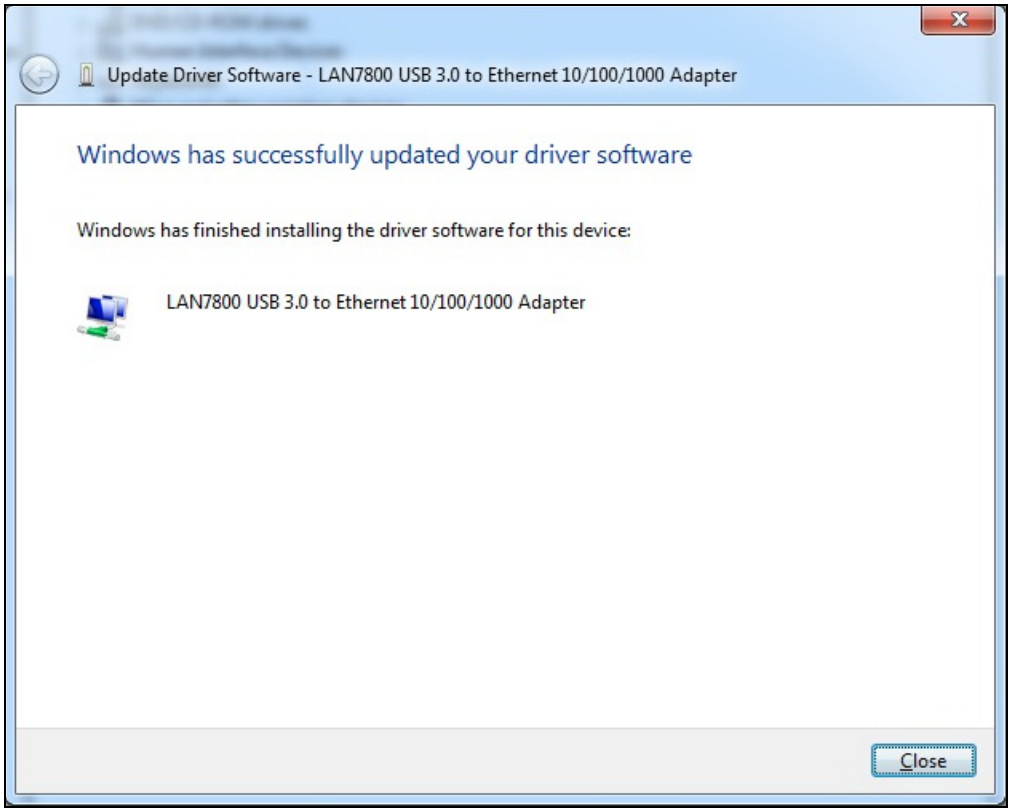
The installation progress displays, as in Figure 3-16. This indicates that some time may be necessary for the installation to complete.

FIGURE 3-16: INSTALLATION PROGRESS WINDOW



When finished, the installation progress window will change to indicate the driver has been installed, as displayed in Figure 3-17.

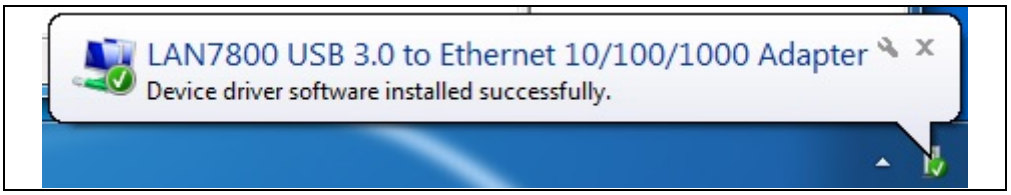
FIGURE 3-17: DEVICE DRIVER UPDATE COMPLETE SCREEN



14. Click **Close**.

    A device driver installation notification will pop-up as displayed in Figure 3-18, indicating the driver installation is complete.

FIGURE 3-18: DEVICE DRIVER INSTALLATION SUCCESS TASK BAR NOTIFICATION



3.3 WINDOWS 7 32/64 BIT DRIVER UNINSTALLATION VIA DEVICE MANAGER

To uninstall the Windows 7 32/64-bit software:

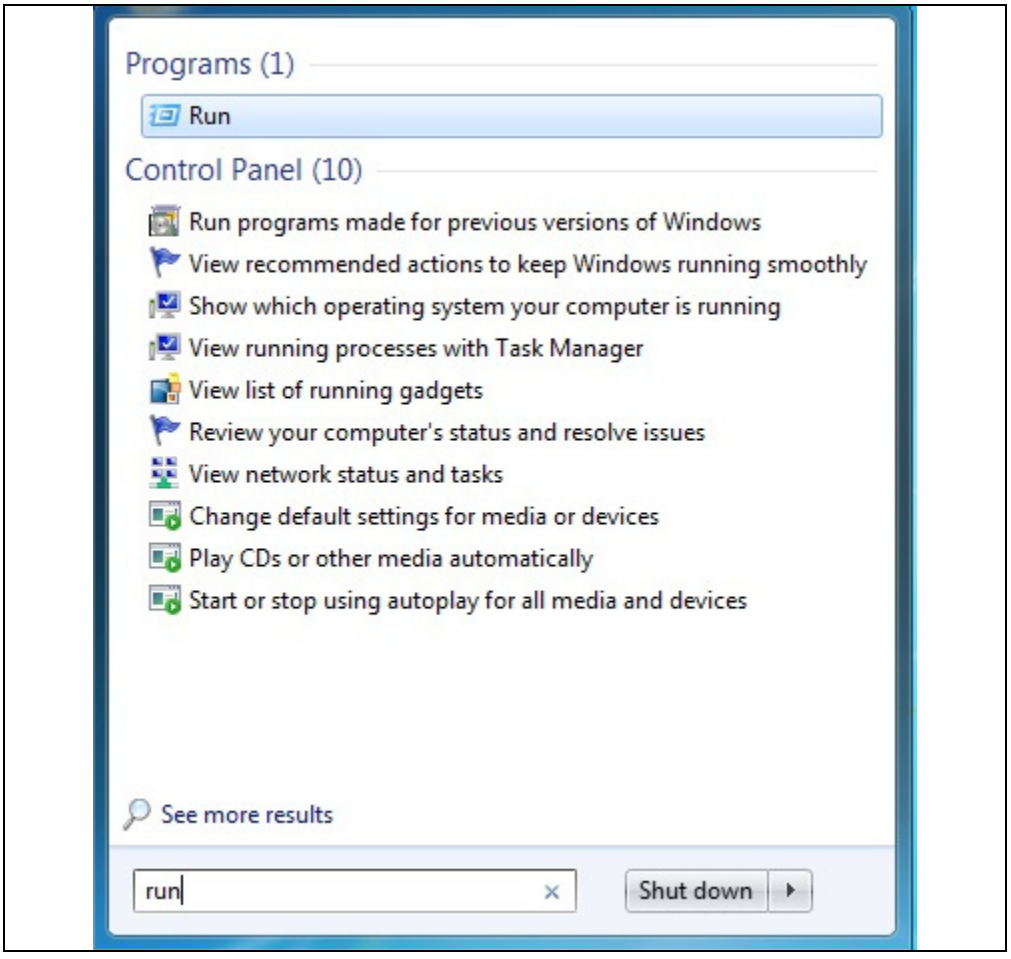
1. From the Start menu, type `Run` in the search field.

   Results from the Programs list display.

2. Click **Run**, as displayed in Figure 3-19.

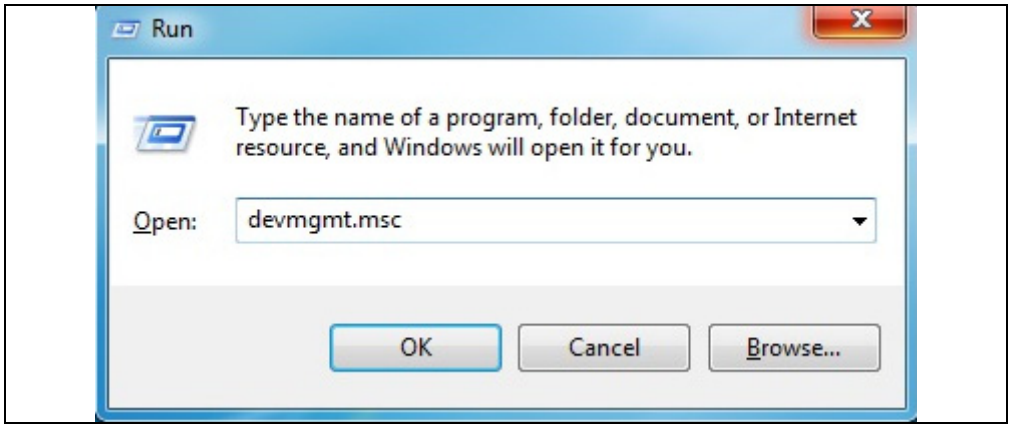
   The Run dialog displays.

FIGURE 3-19: START MENU SEARCH



3. Type `devmgmt.msc`, as displayed in Figure 3-20.

FIGURE 3-20: RUN WINDOW



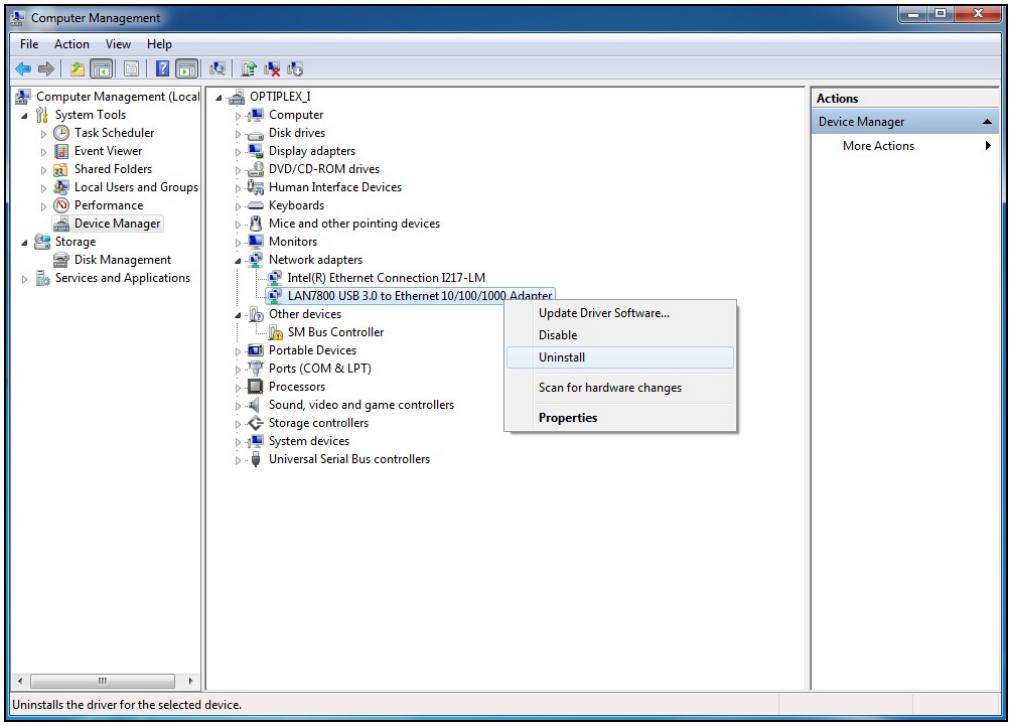
4. Click **OK**.

   The Device Manager window displays.

5. Select the device from the "Network adapters" section.
6. Right-click the device.

   A context menu displays, as in Figure 3-21.

FIGURE 3-21: DEVICE MANAGER WINDOW



7. Click Uninstall.

   The Confirm Device Uninstall window displays

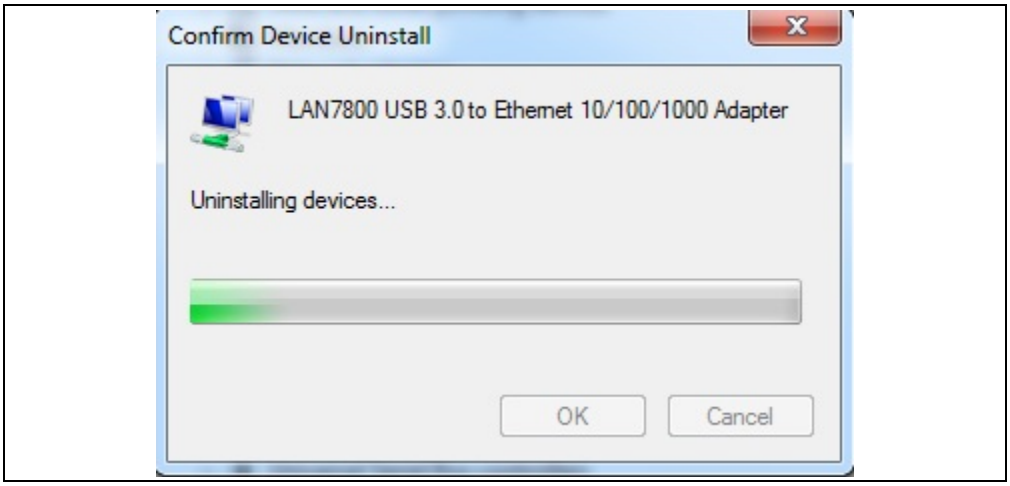
8. Select the "Delete the driver software for this device." checkbox, as displayed in Figure 3-22.

FIGURE 3-22: CONFIRM DEVICE REMOVAL WINDOW

9. Click **OK**.

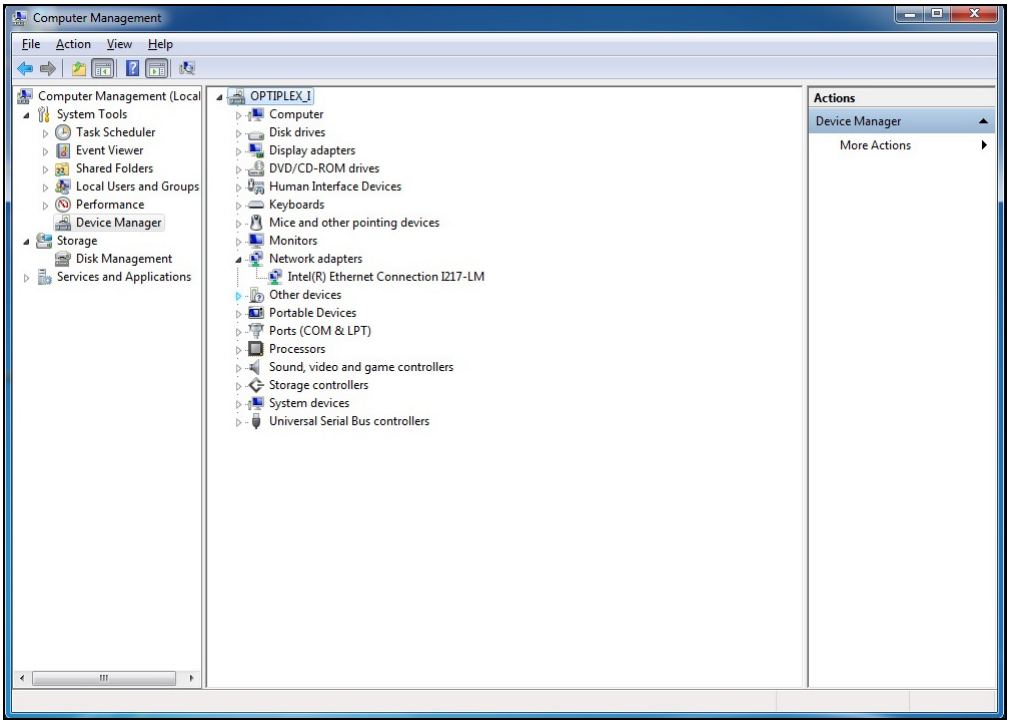
   The Confirm Device Uninstall window displays a progress indicator as the device drivers are being removed, as displayed in Figure 3-23.

FIGURE 3-23: DEVICE UNINSTALL PROGRESS WINDOW



The device driver is no longer listed in the Device Manager window, as displayed in Figure 3-24. The uninstallation process is now complete.

FIGURE 3-24: DEVICE REMOVED FROM DEVICE MANAGER



3.4 WINDOWS 7 32/64 BIT DRIVER UNINSTALLATION VIA CONTROL PANEL

This section details how to use the automated uninstaller using Windows 7's “Programs and Features”.

To uninstall the Windows 7 32/64-bit software:

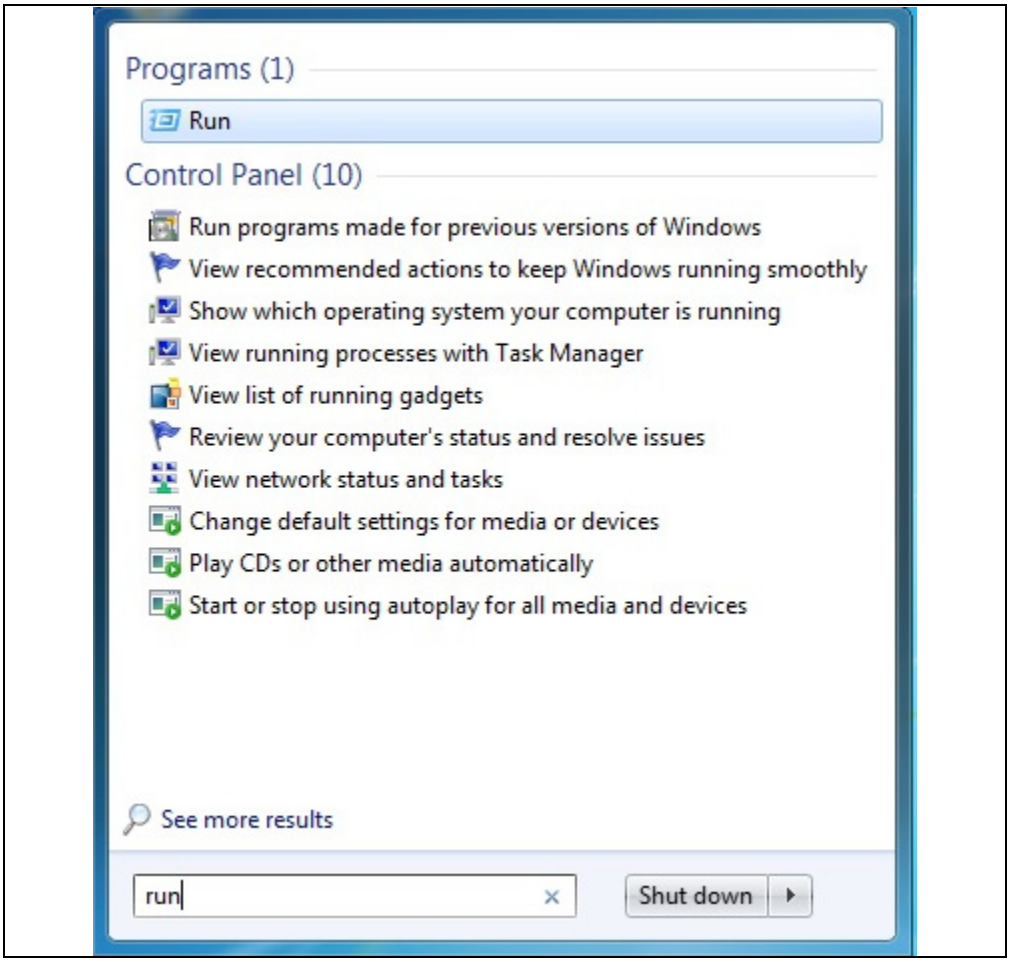
1. From the Start menu, type `Run` in the search field.

   Results from the Programs list display.

2. Click **Run**, as displayed in Figure 3-25.

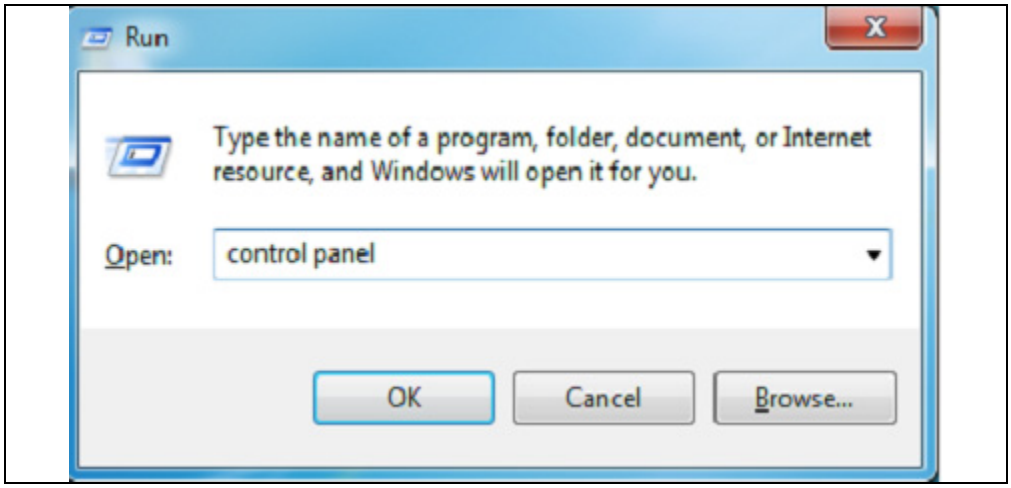
   The Run dialog displays.

FIGURE 3-25: START MENU SEARCH



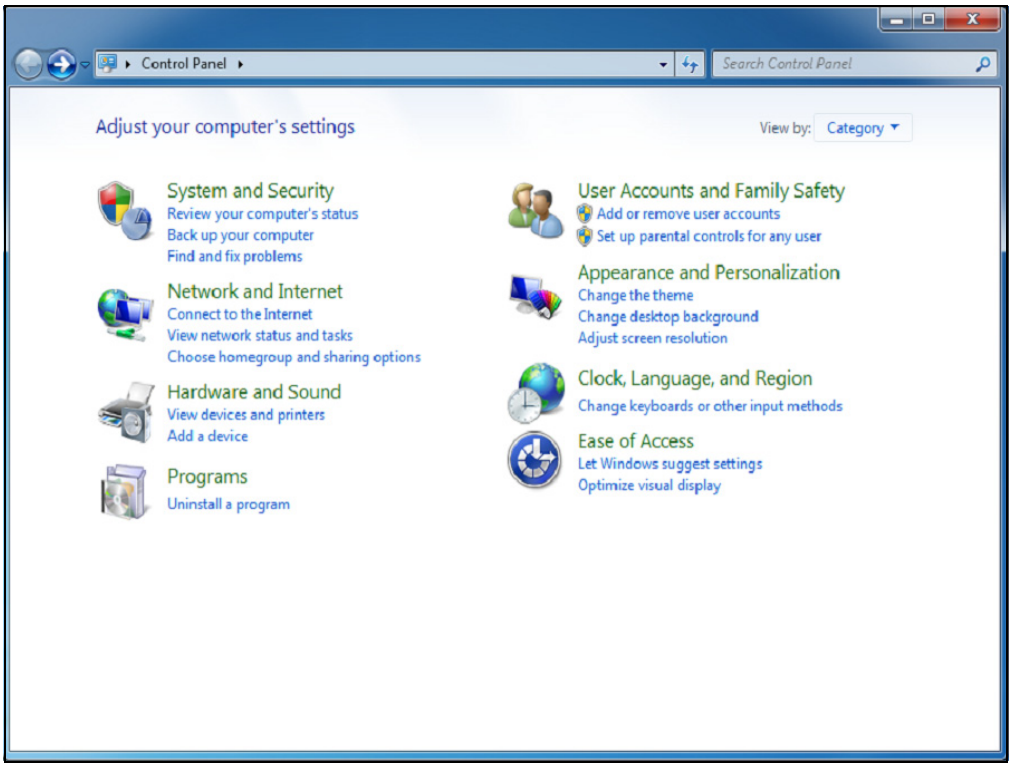
3. Type `control panel`, as displayed in Figure 3-26.

FIGURE 3-26: RUN WINDOW



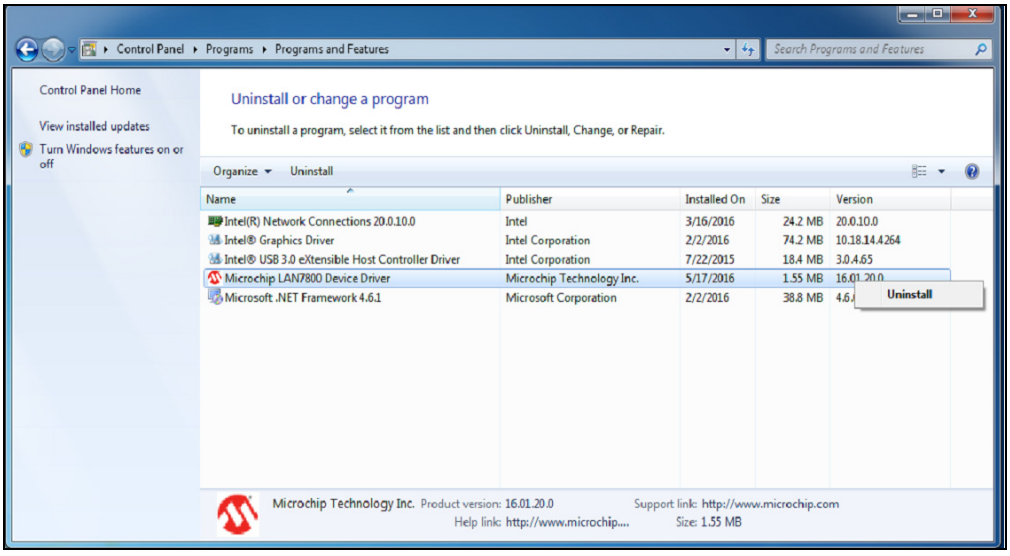
The Control Panel window displays, as in Figure 3-27.

FIGURE 3-27: CONTROL PANEL WINDOW



4. Click **Uninstall a program**.
   The Programs and Features window displays.
5. Right-click Microchip LAN7800 Device Driver.
   A context menu displays, as displayed in Figure 3-28.

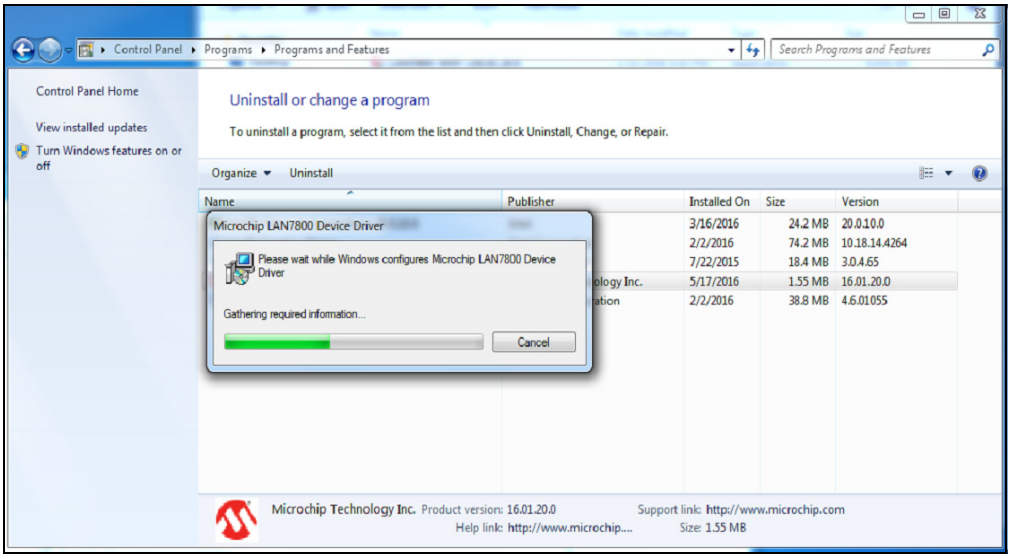
FIGURE 3-28: DRIVER UNINSTALL WINDOW



6. Click **Uninstall**.

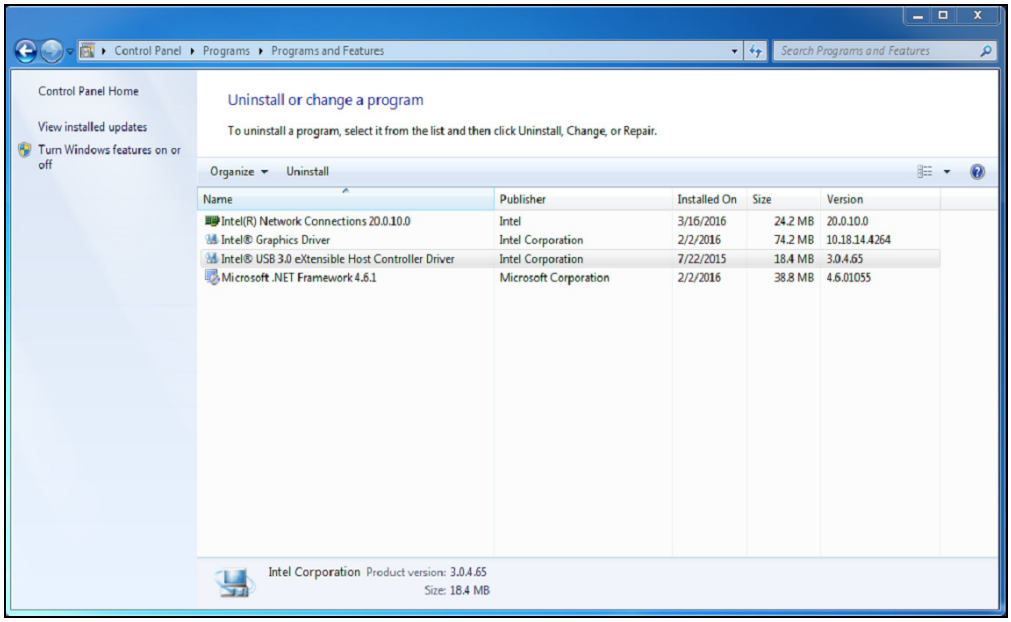
   The LAN7800 driver uninstalls, as displayed in Figure 3-29.

FIGURE 3-29: DEVICE UNINSTALL PROGRESS WINDOW



The LAN7800 driver Control Panel entry is removed, as displayed in Figure 3-30.

FIGURE 3-30: DEVICE REMOVED FROM CONTROL PANEL WINDOW



Chapter 4. Windows 8 32/64-Bit Driver

This chapter details the installation and uninstallation of the Windows 8 32/64-bit driver.

The Windows 8 32/64-bit driver may be installed in two ways:

- **Windows 8 32/64-Bit Driver Installation via EXE** (preferred method)
- **Windows 8 32/64-Bit Driver Installation via INF**

Windows 8 32/64-bit driver may be uninstalled in two ways:

- **Windows 8 32/64 Bit Driver Uninstallation via Device Manager**
- **Windows 8 32/64 Bit Driver Uninstallation via Control Panel**

Note: Windows 8.1 driver installer is the same as the Windows 8 driver installer.

4.1 WINDOWS 8 32/64-BIT DRIVER INSTALLATION VIA EXE

Copy the folder containing the distribution files to the desktop or any other convenient, known place within the directory structure. The folder contains the installer package and release notes file.

To install:

1. Click the installer icon.

   The setup window displays, as in Figure 4-1.

FIGURE 4-1: DEVICE INSTALLER INVOCATION

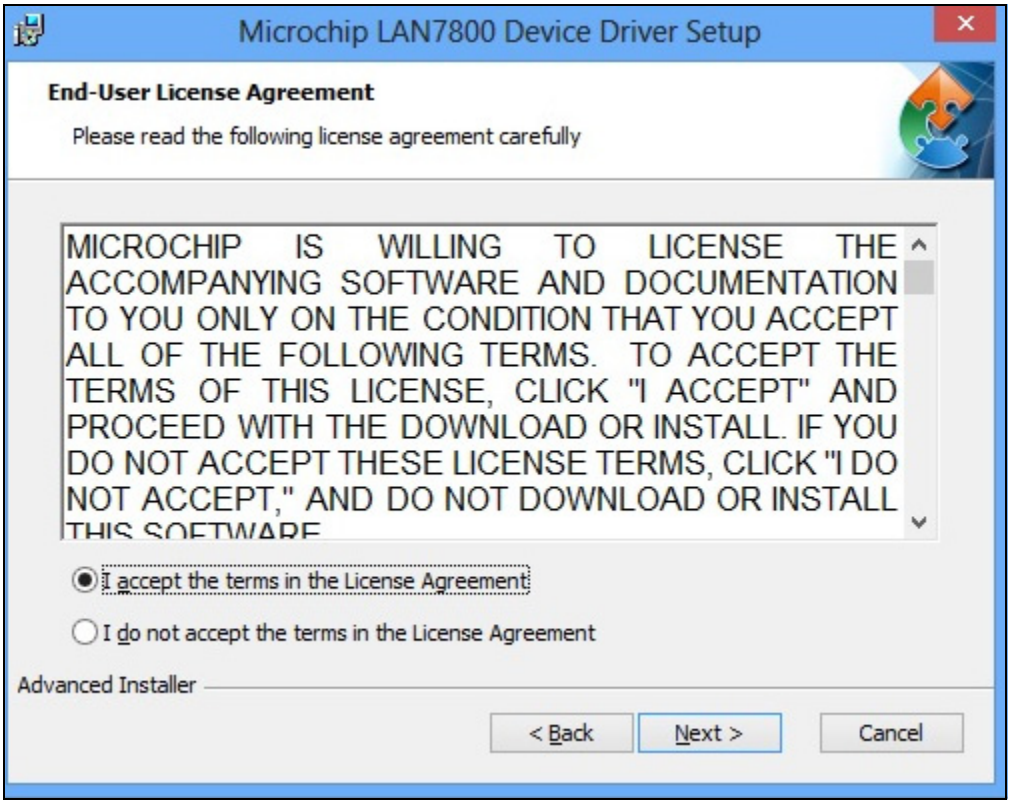
2. Click **Next >**.

   The End User License Agreement (EULA) displays.

3. Click the "I accept the terms in the License Agreement" radio button, as displayed in Figure 4-2.

   Once this has been selected, the **Next >** button is enabled, allowing the installation process to continue.

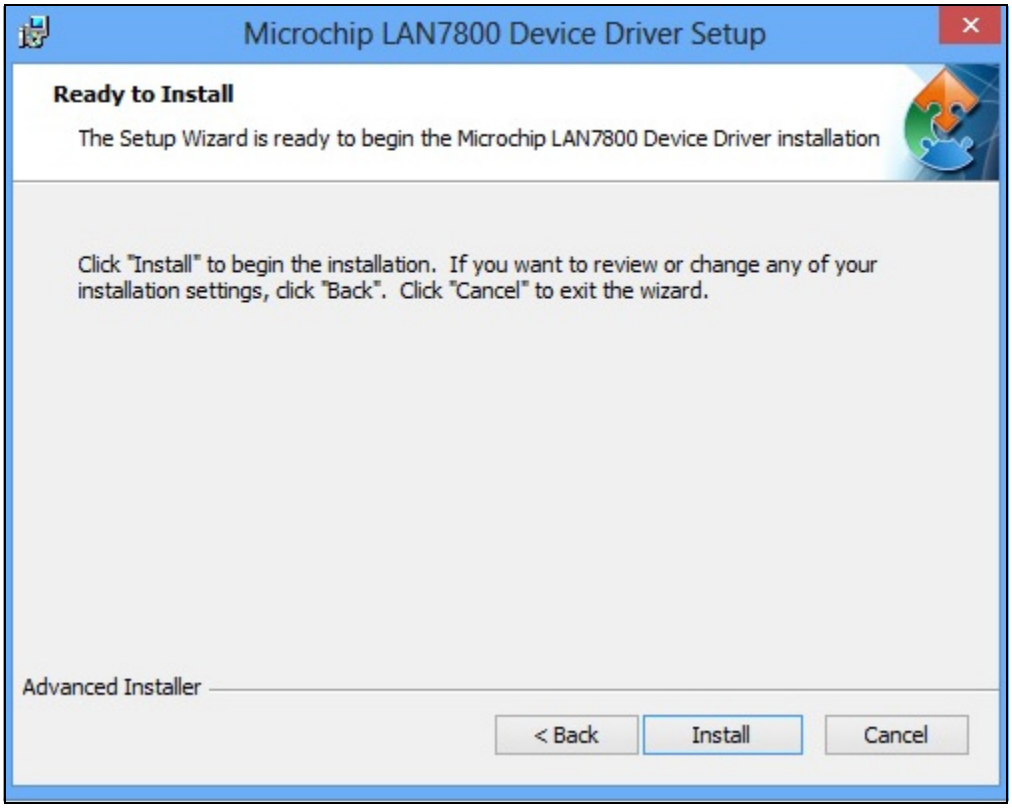
FIGURE 4-2: END USER LICENSE AGREEMENT



4. Click **Next >**.

   The Device Installer window prompts to either go back to review or change installation settings, proceed to install, or to cancel completely, as displayed in Figure 4-3.

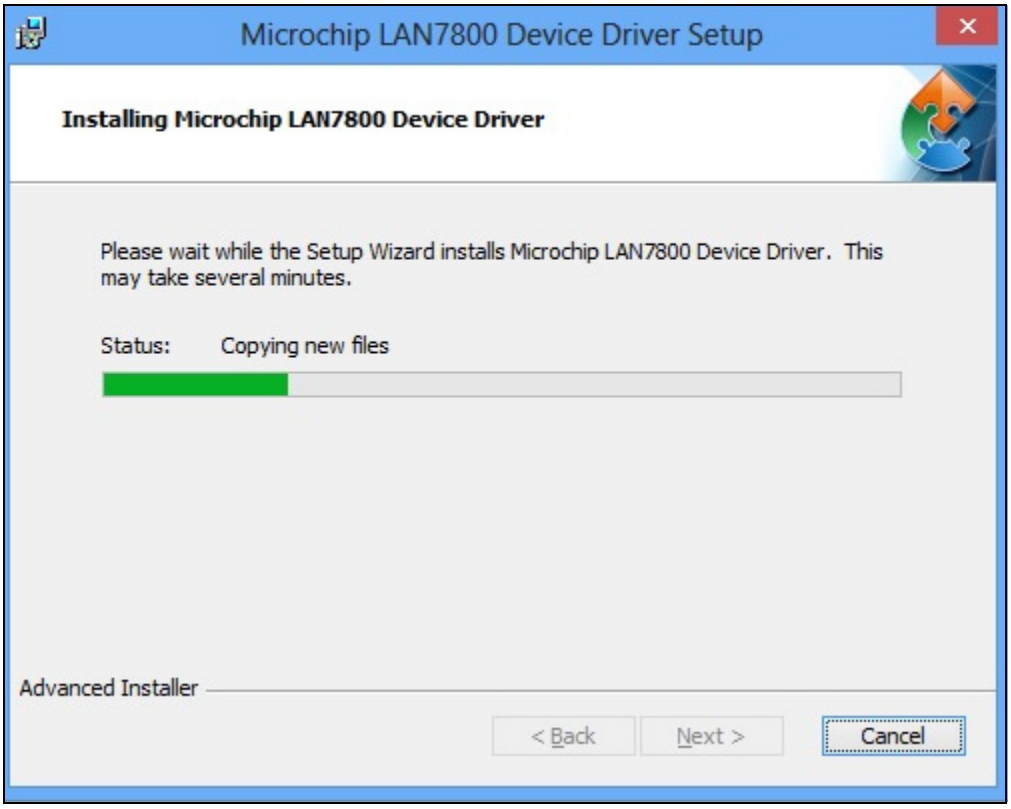
FIGURE 4-3: INSTALLATION WINDOW



5. Click **Install**.

   The Installation Progress window displays, as in Figure 4-4. The "Status" field displays the progress.

FIGURE 4-4: INSTALLATION PROGRESS WINDOW



When setup is done, the window displays a confirmation message, as displayed in Figure 4-5.

6. Click **Finish**.

FIGURE 4-5: DEVICE DRIVER INSTALLATION COMPLETE SCREEN - WINDOWS 8 32/64-BIT

7. Plug the device into an available USB port on the computer.

   The device installation is complete, and the device will be setup to have its IP address assigned by a DHCP server. This configuration may be changed to use a manually assigned IP address, using the device's Internet protocol properties found in Control Panel's Network Connections. Details are outside of the scope of this document.

4.2 WINDOWS 8 32/64-BIT DRIVER INSTALLATION VIA INF

Alternately, the device driver may be installed using a setup information file (`.inf`). This section details its installation method.

> **Note:** The method described in **Section 4.1 "Windows 8 32/64-Bit Driver Installation via EXE"** is the preferred method of installation.

Copy the folder containing the Microchip INF distribution files to the desktop or any other convenient, known place within the directory structure.

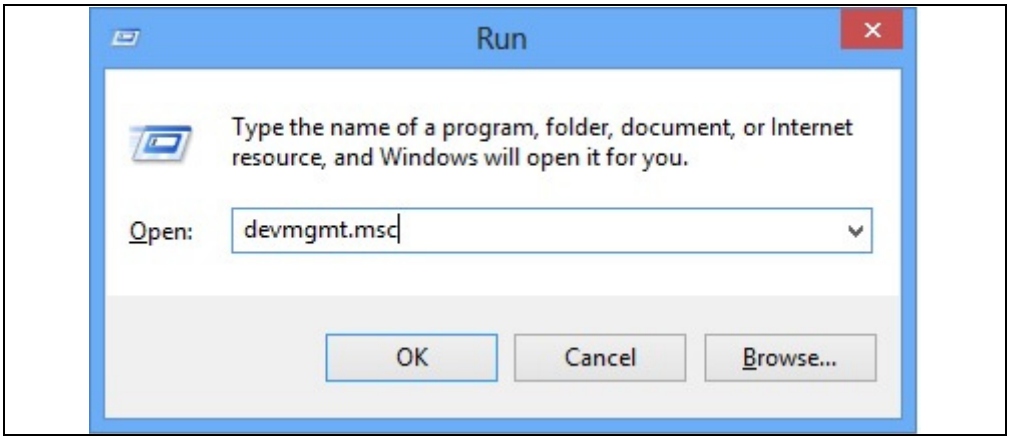
To install:

1. Connect the device to an available USB port on the computer.
2. From the Start menu, type `Run` in the search field.

   Results from the Programs list display.
3. Click **Run**.

   The Run dialog displays.
4. Type `devmgmt.msc`, as displayed in Figure 4-6.

FIGURE 4-6: RUN WINDOW

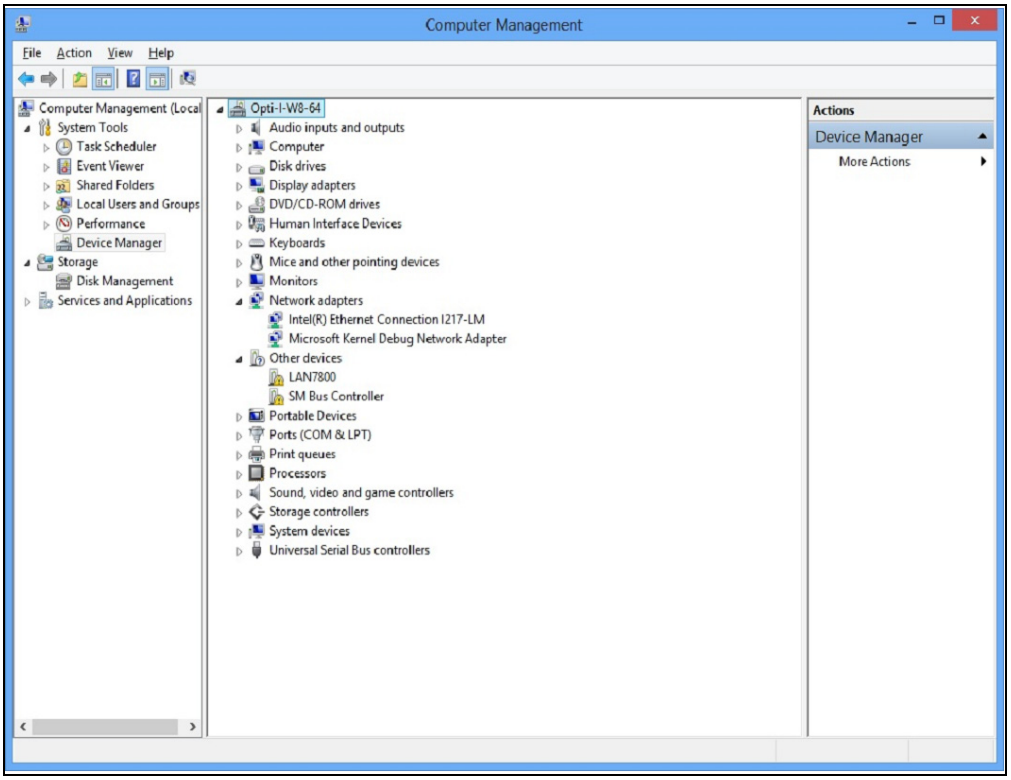


5. Click **OK**.

   The Device Manager window displays.

6. Select the device from the “Other devices” section, as displayed in Figure 4-7.

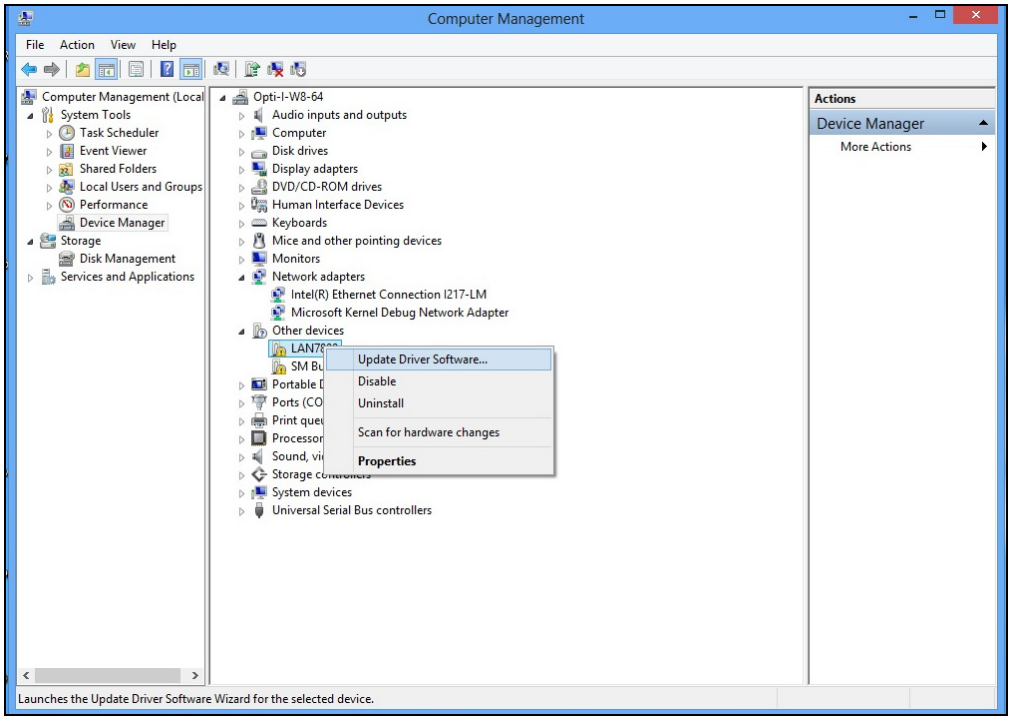
FIGURE 4-7: DEVICE MANAGER WINDOW - OTHER DEVICES



7. Right-click the device.

   A context menu displays, as in Figure 4-8.

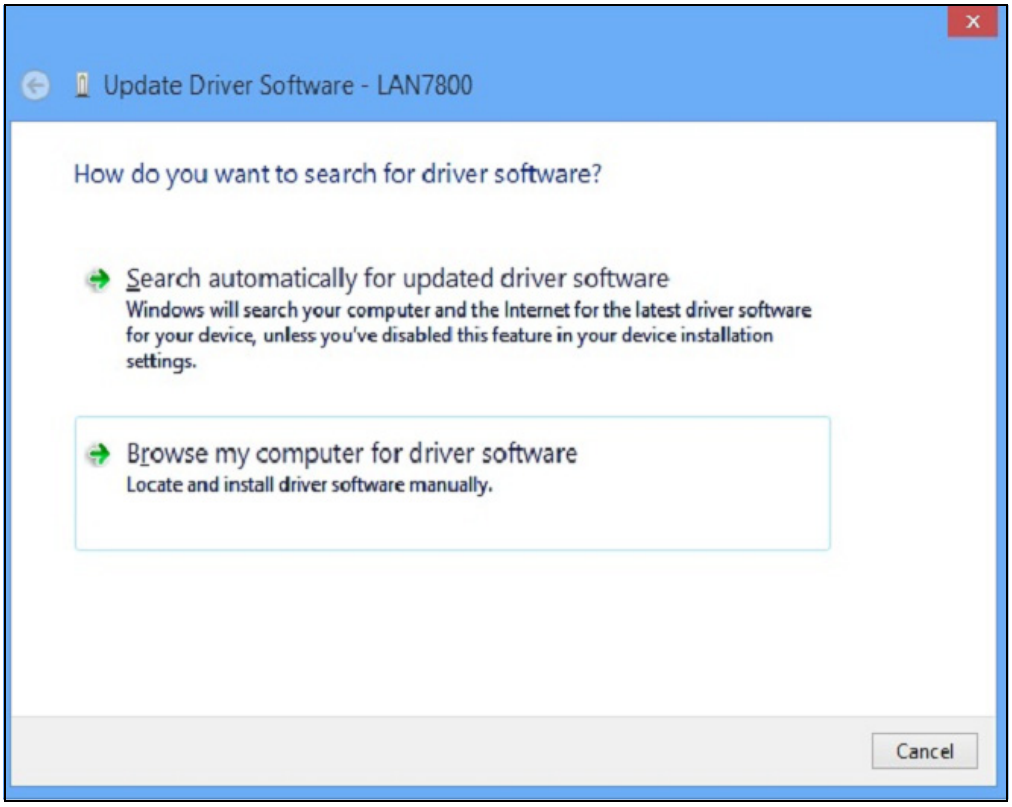
FIGURE 4-8: DEVICE MANAGER WINDOW - UPDATE DRIVER SOFTWARE



8. Click Update Driver Software.

   The Update Driver Software window displays, as in Figure 4-9.

FIGURE 4-9: DRIVER INSTALL OPTIONS WINDOW



9. Click **Browse my computer for driver software**.
10. Click **Browse…**.

The Browse For Folder dialog displays.

11. Browse to the location of the copied Microchip INF distribution files.
12. Click **OK**.
13. Click **Next**.

Note: Windows 8 installations must use the driver under the `ndis630` folder. For Windows 32-bit installations, browse to the `x86` folder within the `ndis630` folder of the Microchip INF distribution files, as displayed in Figure 4-10. For Windows 64-bit installations, browse to the `x64` folder within the `ndis630` folder of the Microchip INF distribution files, as displayed in Figure 4-11.

FIGURE 4-10: BROWSE WINDOW - WINDOWS 8 32-BIT

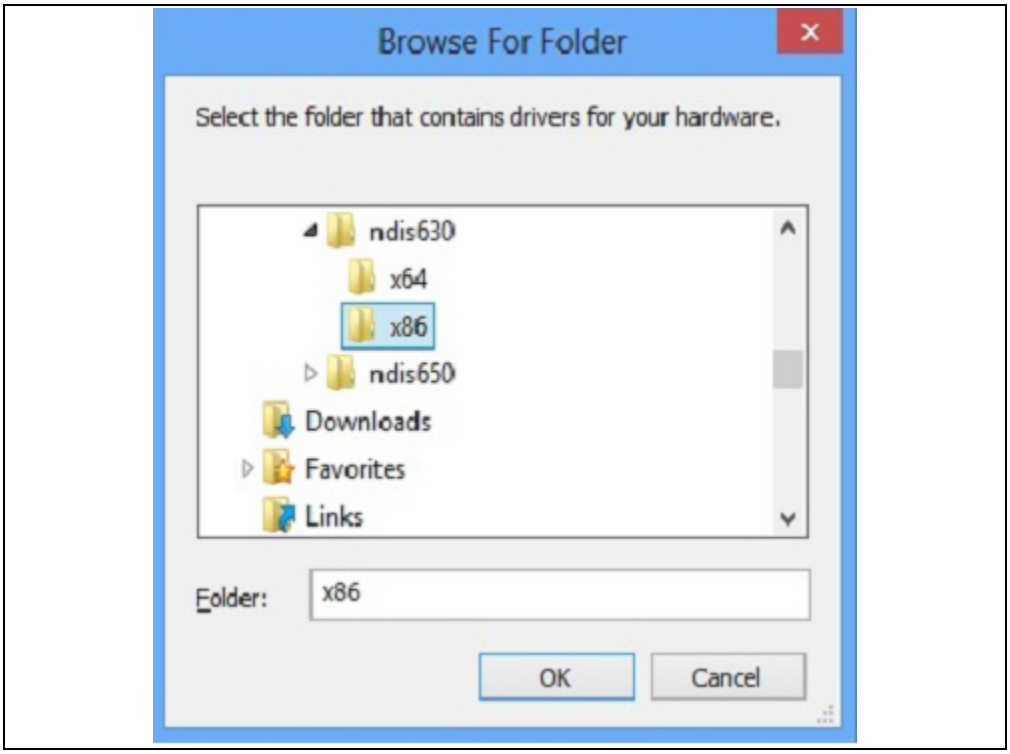
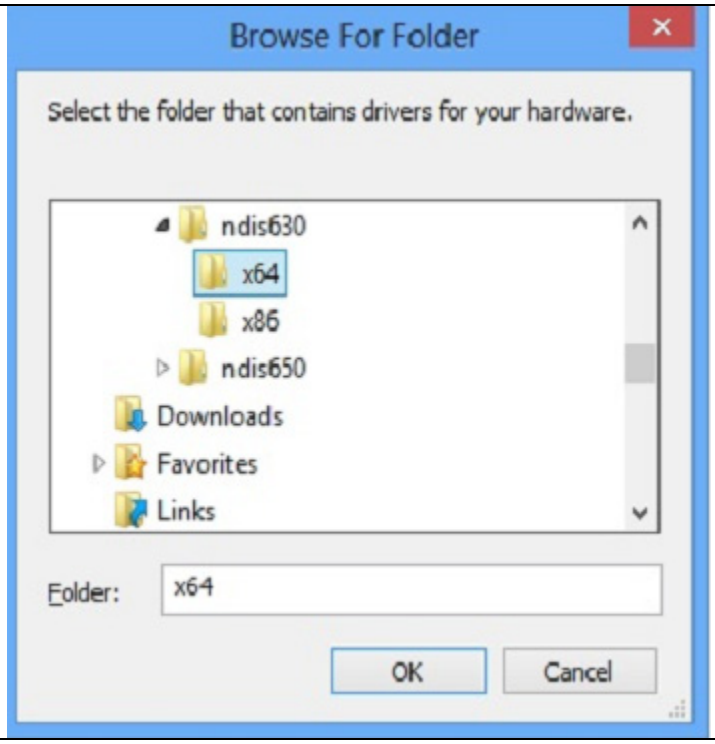
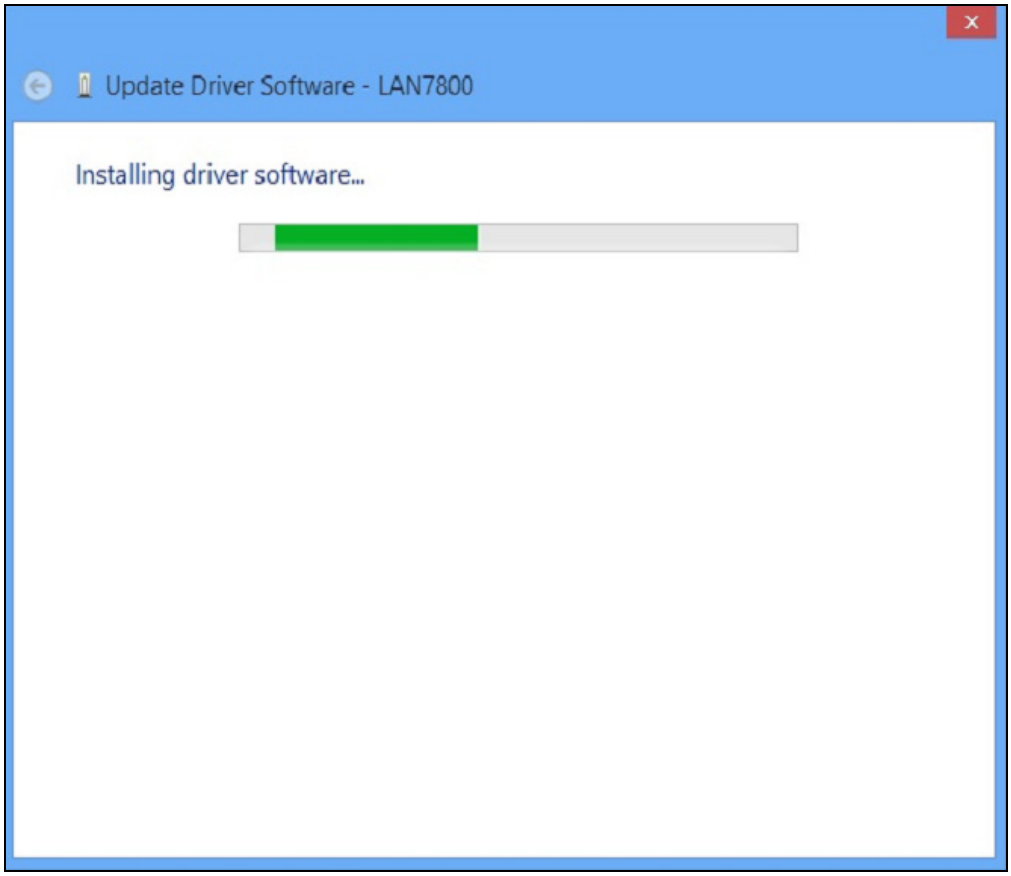


FIGURE 4-11: BROWSE WINDOW - WINDOWS 8 64-BIT



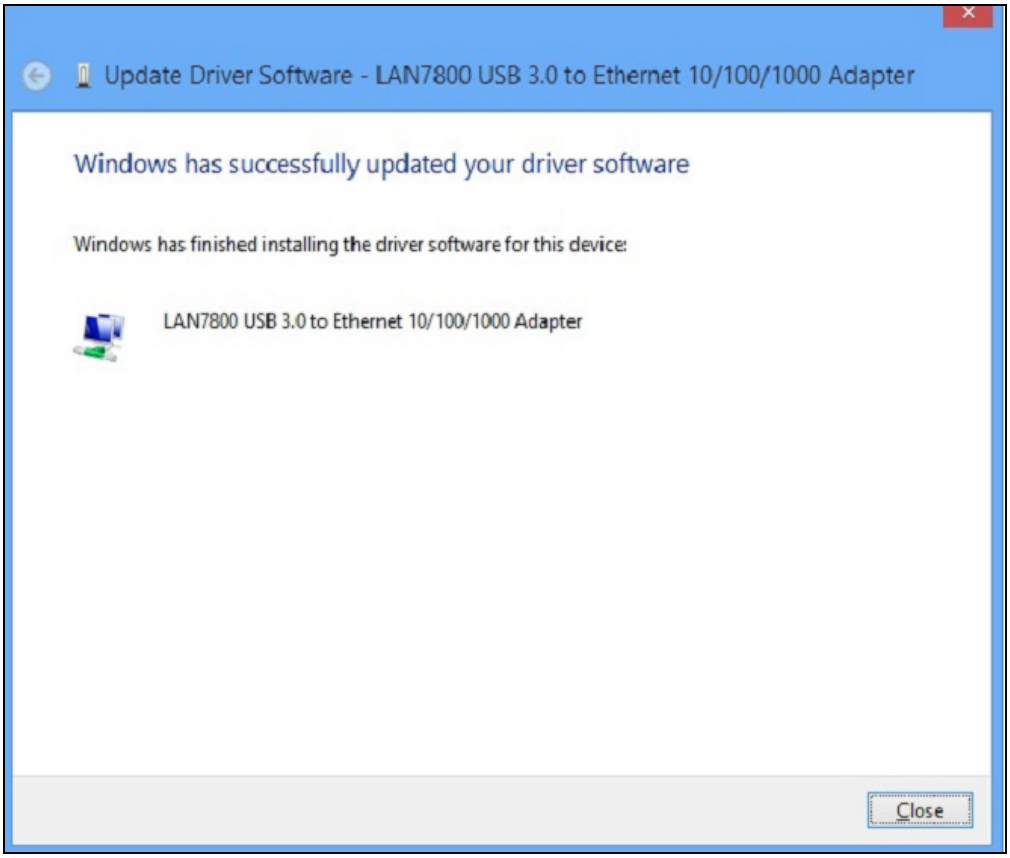
The installation progress displays, as in Figure 4-12. This indicates that some time may be necessary for the installation to complete.

FIGURE 4-12: INSTALLATION PROGRESS WINDOW



When finished, the installation progress window will change to indicate the driver has been installed. Click the “Close” button to complete the driver installation as displayed in Figure 4-13.

FIGURE 4-13: DEVICE DRIVER UPDATE COMPLETE SCREEN

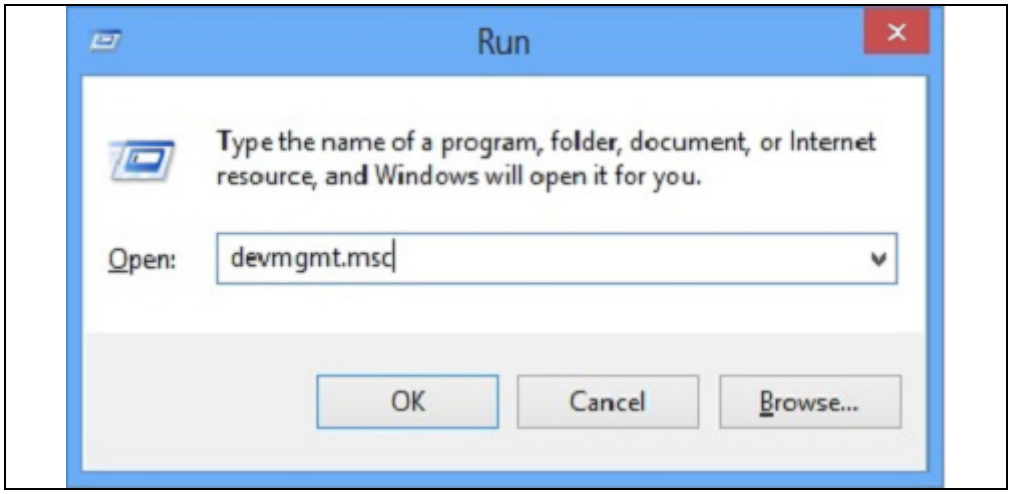


4.3 WINDOWS 8 32/64 BIT DRIVER UNINSTALLATION VIA DEVICE MANAGER

To uninstall the Windows 8 32/64-bit software:

1. From the Start menu, type `Run` in the search field.
   Results from the Programs list display.
2. Click **Run**.
   The Run dialog displays.
3. Type `devmgmt.msc`, as displayed in Figure 4-14.

FIGURE 4-14: RUN WINDOW



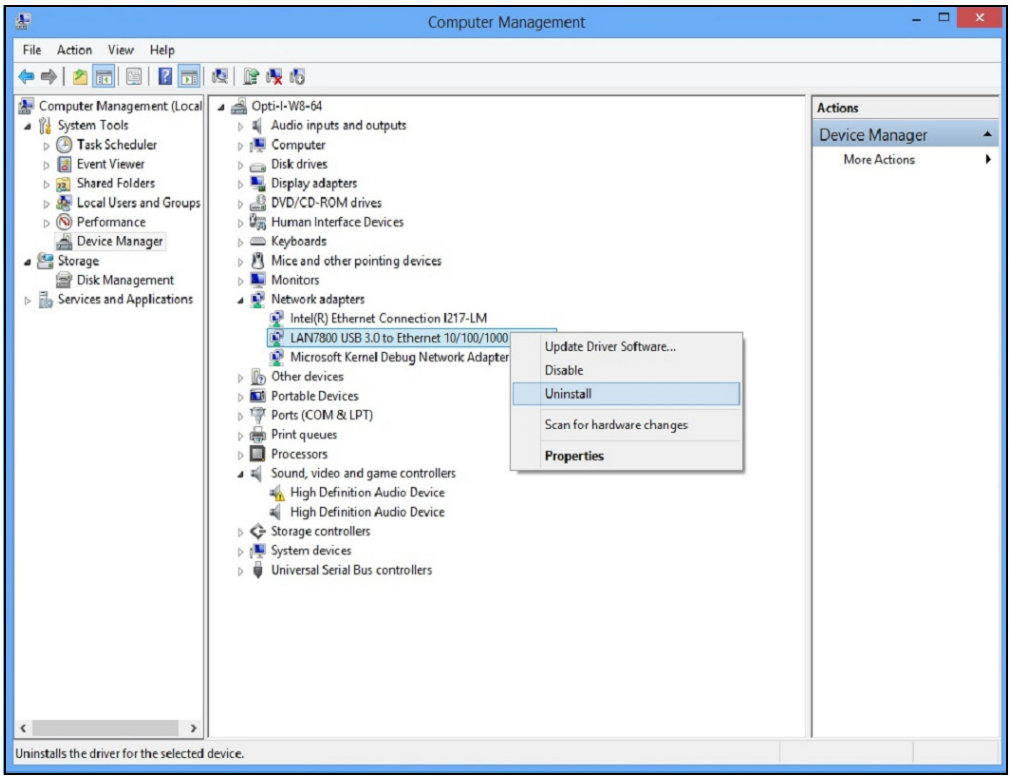
4. Click **OK**.

   The Device Manager window displays.

5. Select the device from the "Network adapters" section.
6. Right-click the device.

   A context menu displays, as in Figure 4-15.

FIGURE 4-15: DEVICE MANAGER WINDOW



7. Click Uninstall.

   The Confirm Device Uninstall window displays

8. Select the "Delete the driver software for this device." checkbox, as displayed in Figure 4-16.

FIGURE 4-16: CONFIRM DEVICE REMOVAL WINDOW

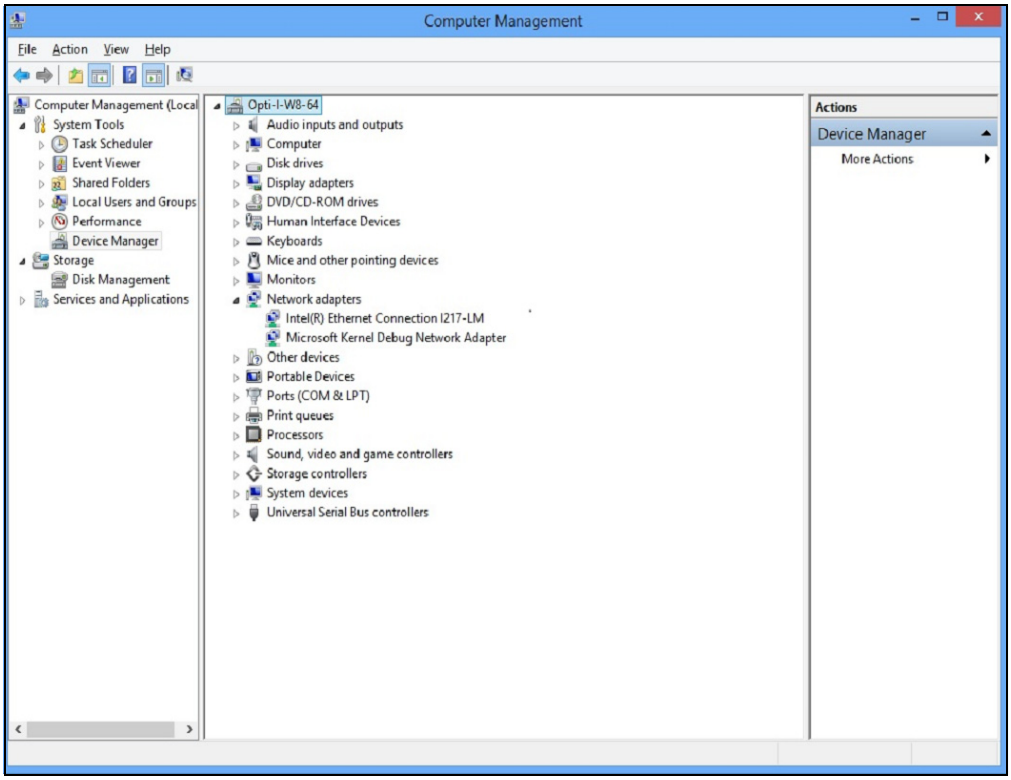
9. Click **OK**.

   The Confirm Device Uninstall window displays a progress indicator as the device drivers are being removed, as displayed in Figure 4-17.

FIGURE 4-17: DEVICE UNINSTALL PROGRESS WINDOW

The device driver is no longer listed in the Device Manager window, as displayed in Figure 4-18. The uninstallation process is now complete.

FIGURE 4-18: DEVICE REMOVED FROM DEVICE MANAGER



4.4 WINDOWS 8 32/64 BIT DRIVER UNINSTALLATION VIA CONTROL PANEL

This section details how to use the automated uninstaller using Windows 8's "Programs and Features".

To uninstall the Windows 8 32/64-bit software:

1. From the Start menu, type `Run` in the search field.

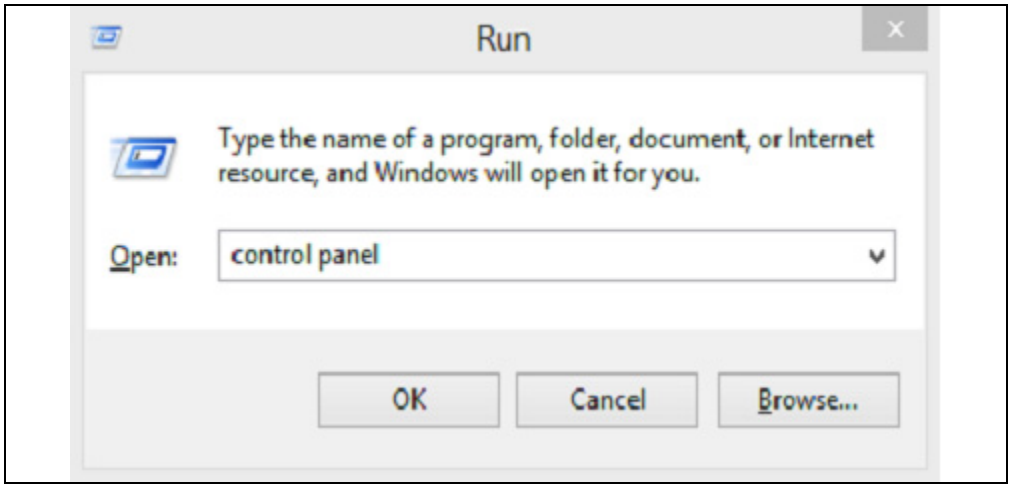
   Results from the Programs list display.

2. Click **Run**.

   The Run dialog displays.

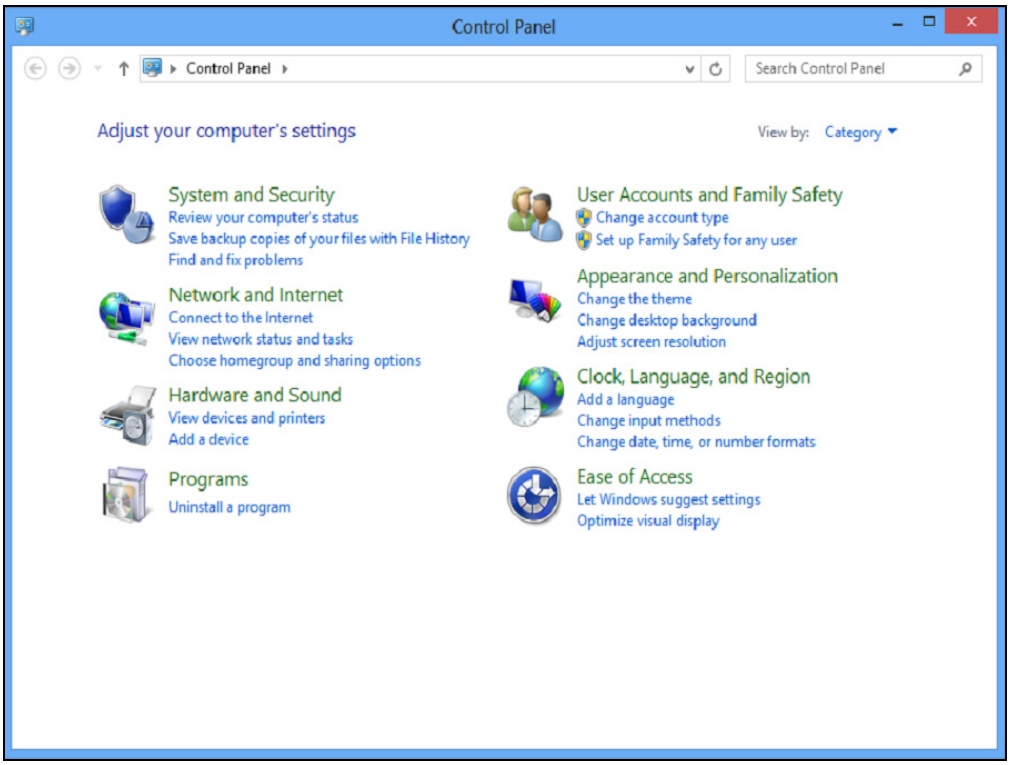
3. Type `control panel`, as displayed in Figure 4-19.

FIGURE 4-19: RUN WINDOW



The Control Panel window displays, as in Figure 4-20.

FIGURE 4-20: CONTROL PANEL WINDOW



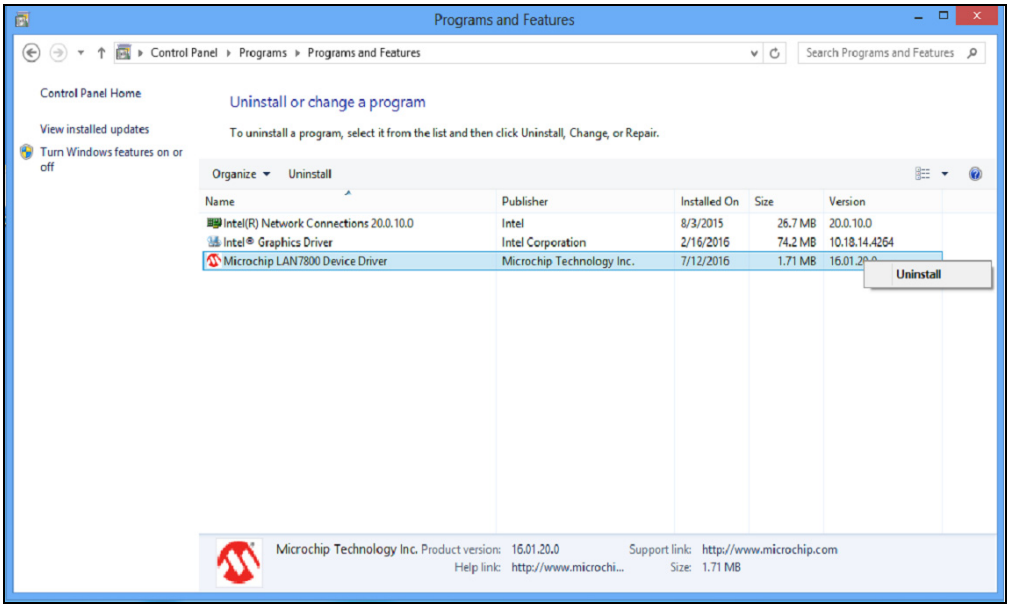
4. Click **Uninstall a program**.

   The Programs and Features window displays.

5. Right-click Microchip LAN7800 Device Driver.

   A context menu displays, as displayed in Figure 4-21.

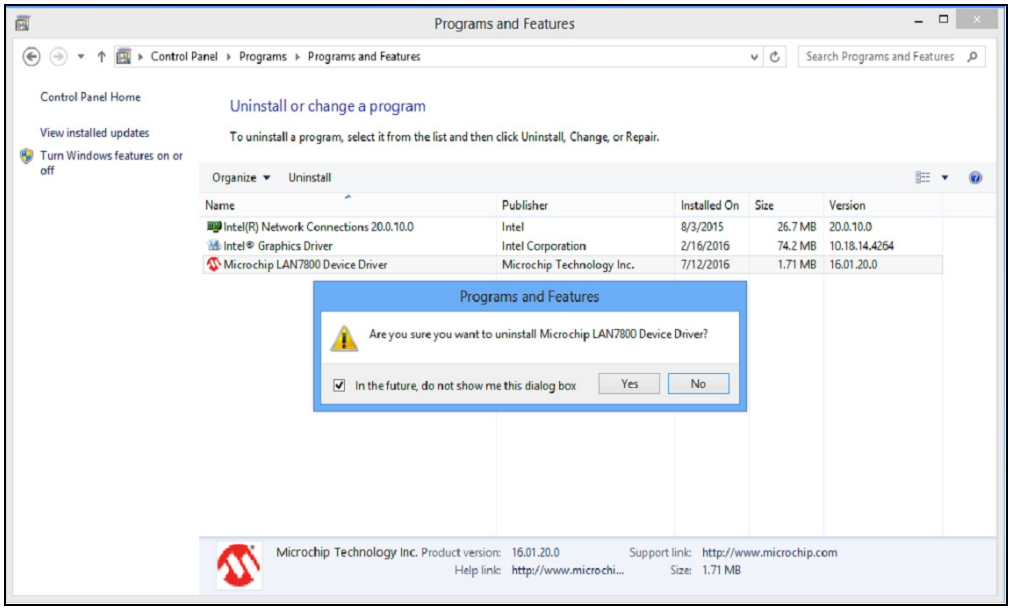
FIGURE 4-21: PROGRAMS AND FEATURES WINDOW



6. Click **Uninstall**.

   A confirmation dialog displays, as in Figure 4-22.

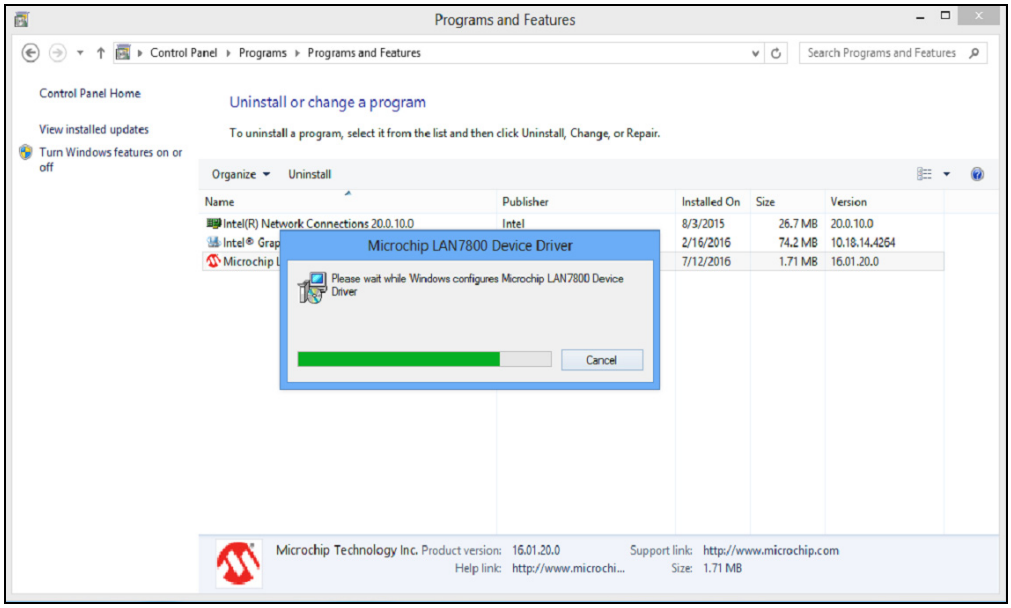
FIGURE 4-22: DRIVER UNINSTALL WINDOW



7. Click **Yes**.

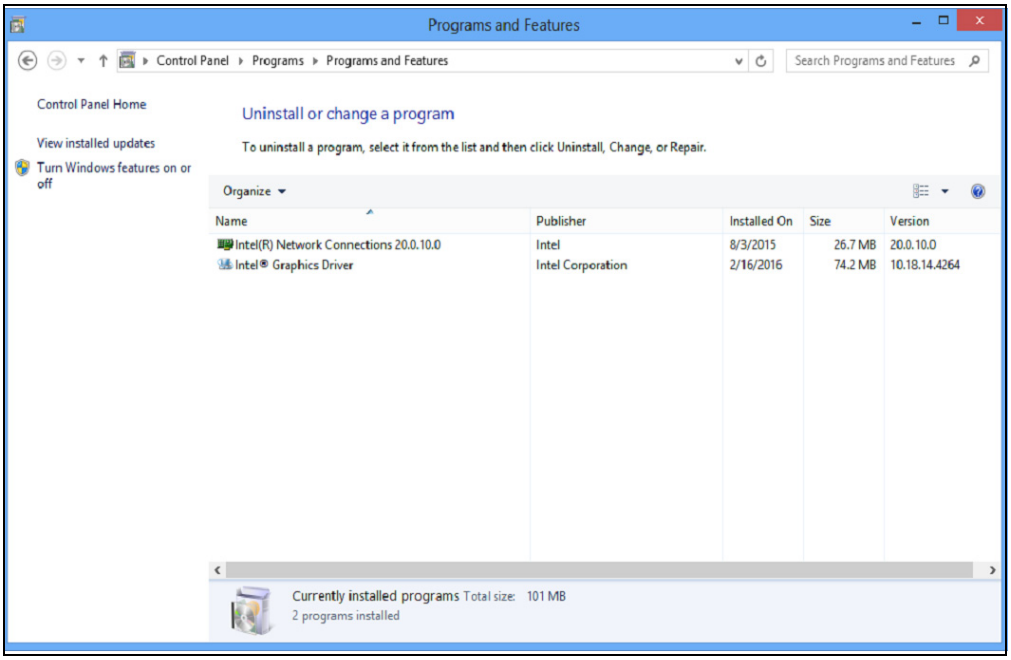
   The LAN7800 driver uninstalls, as displayed in Figure 4-23.

FIGURE 4-23: DRIVER UNINSTALL PROGRESS WINDOW



The LAN7800 driver Control Panel entry is removed, as displayed in Figure 4-24.

FIGURE 4-24: DEVICE REMOVED FROM CONTROL PANEL WINDOW



NOTES:

Chapter 5. Windows 10 32/64-Bit Driver

This chapter details the installation and uninstallation of the Windows 10 32/64-bit driver.

The Windows 10 32/64-bit driver may be installed in two ways:

- **Windows 10 32/64-Bit Driver Installation via EXE** (preferred method)
- **Windows 10 32/64-Bit Driver Installation via INF**

Windows 10 32/64-bit driver may be uninstalled in two ways:

- **Windows 10 32/64 Bit Driver Uninstallation via Device Manager**
- **Windows 10 32/64 Bit Driver Uninstallation via Control Panel**

5.1 WINDOWS 10 32/64-BIT DRIVER INSTALLATION VIA EXE

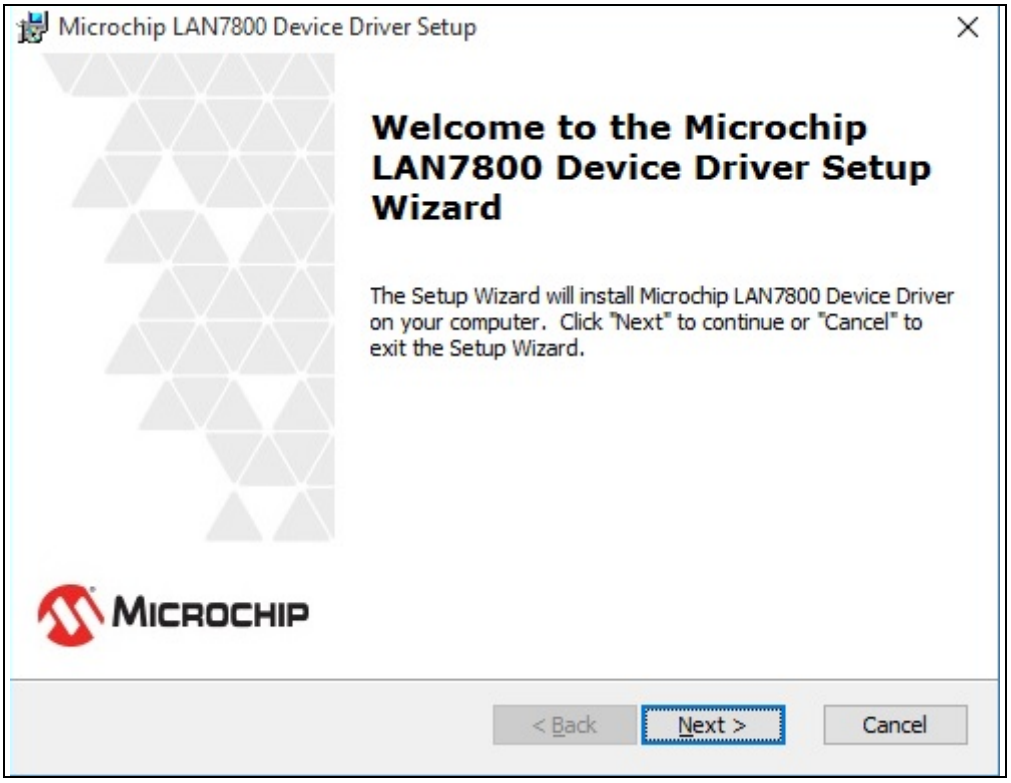
Copy the folder containing the distribution files to the desktop or any other convenient, known place within the directory structure. The folder contains the installer package and release notes file.

To install:

1. Click the installer icon.

   The setup window displays, as in Figure 5-1.

FIGURE 5-1: DEVICE INSTALLER INVOCATION



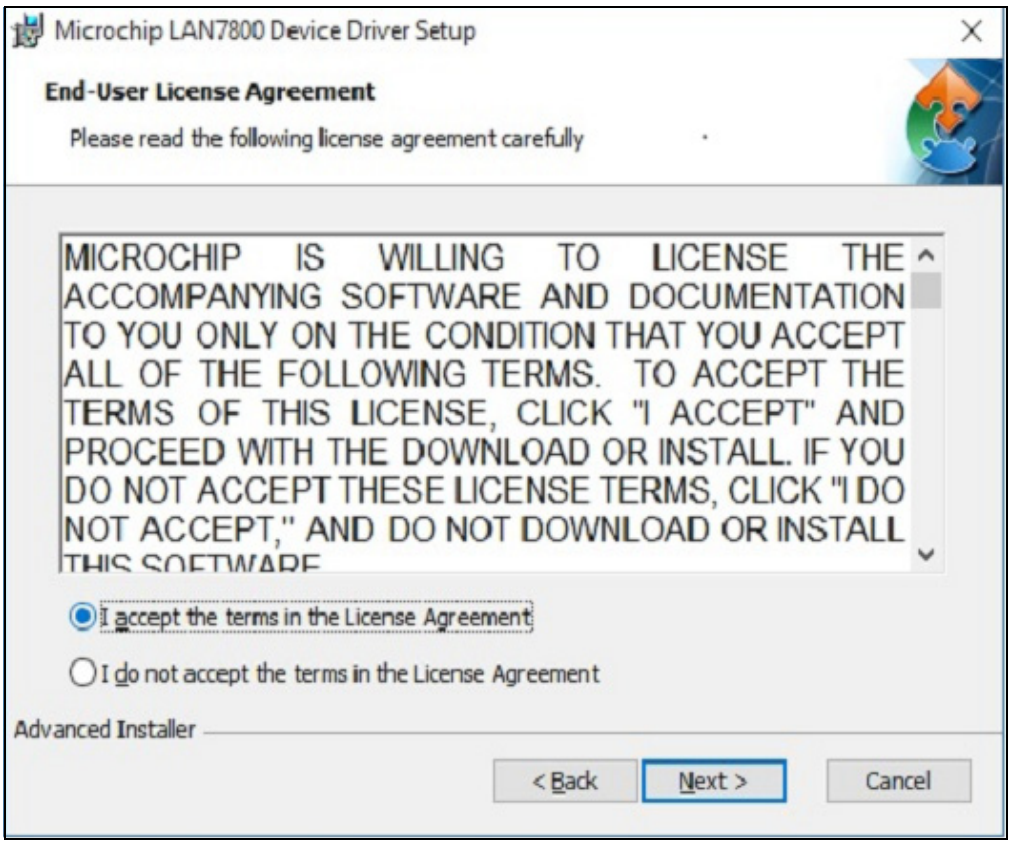
2. Click **Next >**.

   The End User License Agreement (EULA) displays.

3. Click the "I accept the terms in the License Agreement" radio button, as displayed in Figure 5-2.

   Once this has been selected, the **Next >** button is enabled, allowing the installation process to continue.

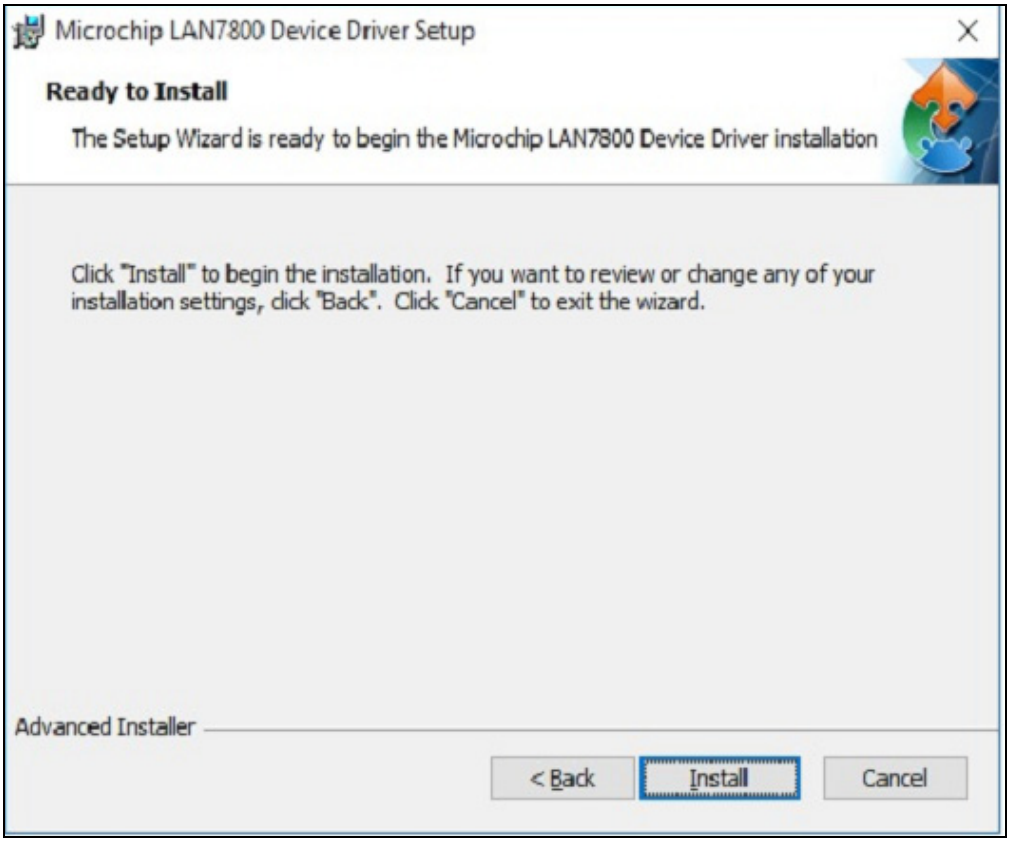
FIGURE 5-2: END USER LICENSE AGREEMENT



4. Click **Next >**.

   The Device Installer window prompts to either go back to review or change installation settings, proceed to install, or to cancel completely, as displayed in Figure 5-3.

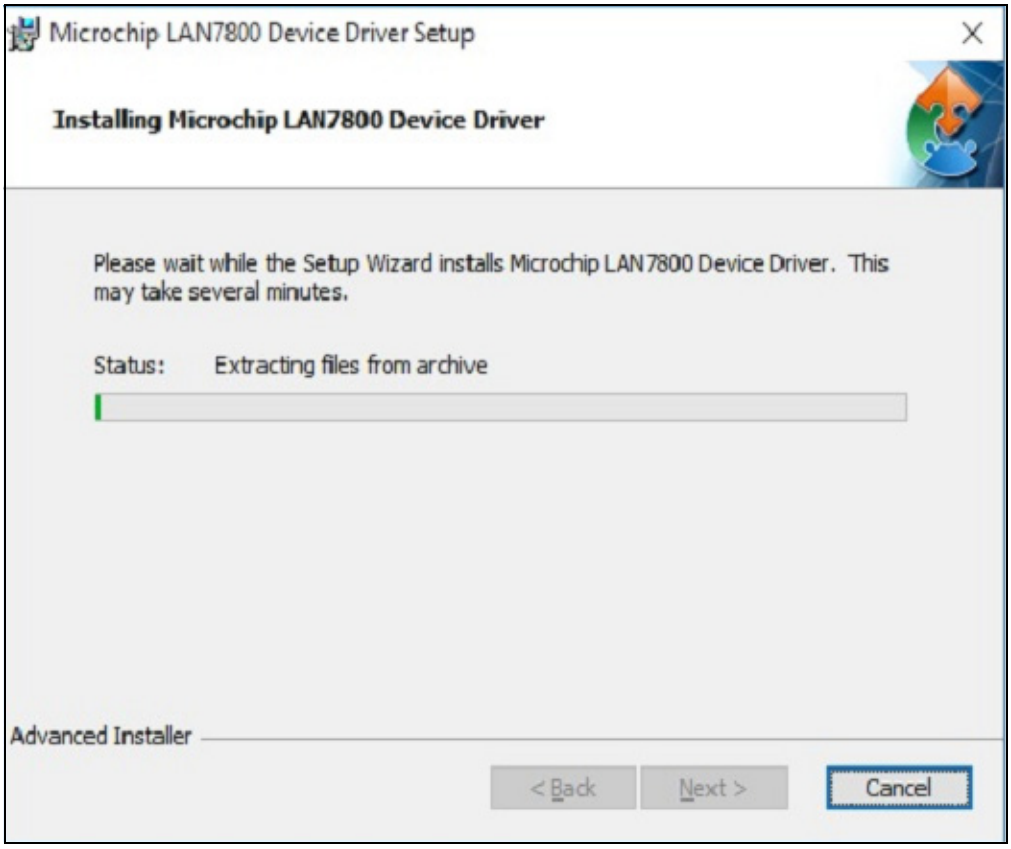
FIGURE 5-3: INSTALLATION WINDOW



5. Click **Install**.

   The Installation Progress window displays, as in Figure 5-4. The “Status” field displays the progress.

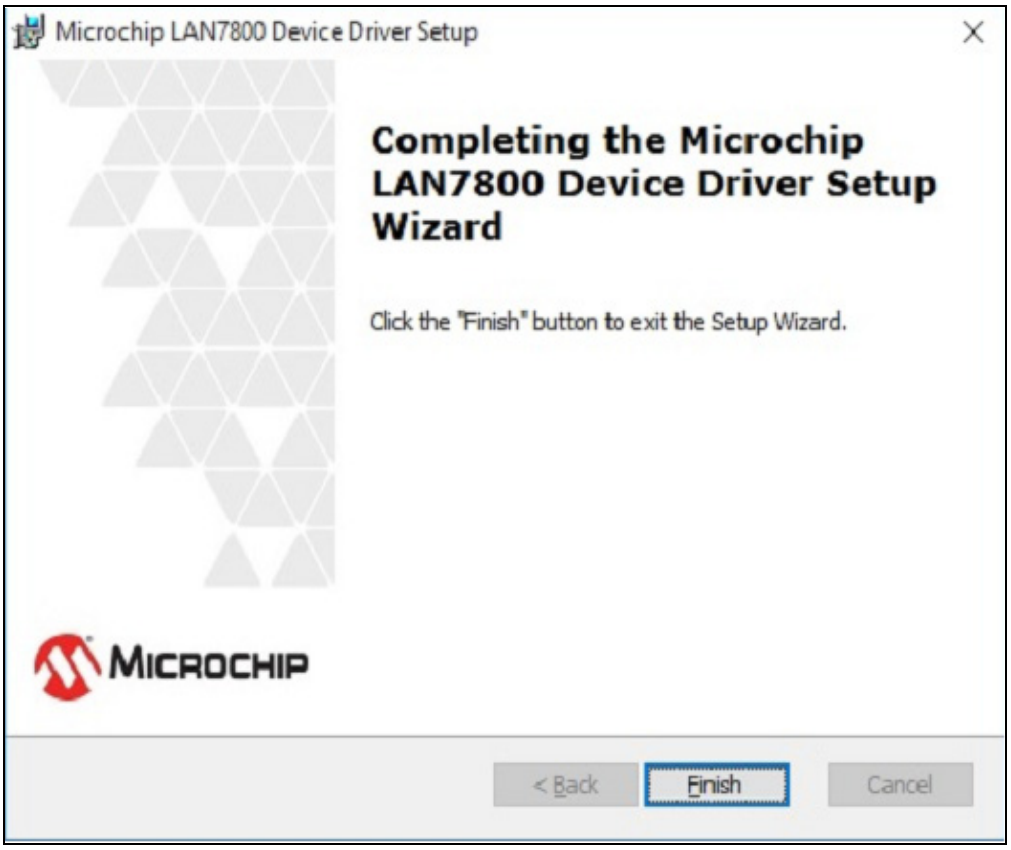
FIGURE 5-4: INSTALLATION PROGRESS WINDOW



When setup is done, the window displays a confirmation message, as displayed in Figure 5-5.

6. Click **Finish**.

FIGURE 5-5: DEVICE DRIVER INSTALLATION COMPLETE SCREEN - WINDOWS 10 32/64-BIT



7. Plug the device into an available USB port on the computer.

   The device installation is complete, and the device will be setup to have its IP address assigned by a DHCP server. This configuration may be changed to use a manually assigned IP address, using the device's Internet protocol properties found in Control Panel's Network Connections. Details are outside of the scope of this document.

5.2 WINDOWS 10 32/64-BIT DRIVER INSTALLATION VIA INF

Alternately, the device driver may be installed using a setup information file (`.inf`). This section details its installation method.

> **Note:** The method described in **Section 5.1 "Windows 10 32/64-Bit Driver Installation via EXE"** is the preferred method of installation.

Copy the folder containing the Microchip INF distribution files to the desktop or any other convenient, known place within the directory structure.

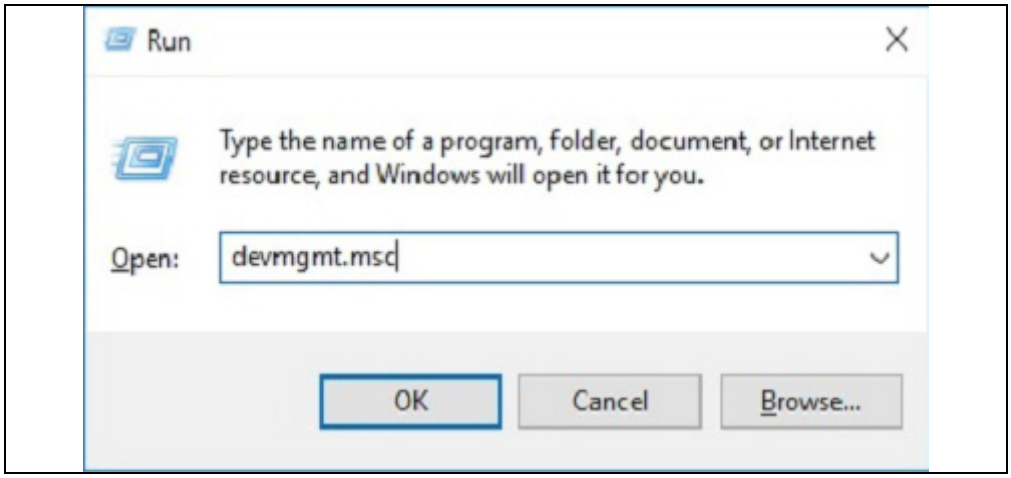
To install:

1. Connect the device to an available USB port on the computer.
2. From the Start menu, type `Run` in the search field.

   Results from the Programs list display.
3. Click **Run**.

   The Run dialog displays.
4. Type `devmgmt.msc`, as displayed in Figure 5-6.

FIGURE 5-6: RUN WINDOW

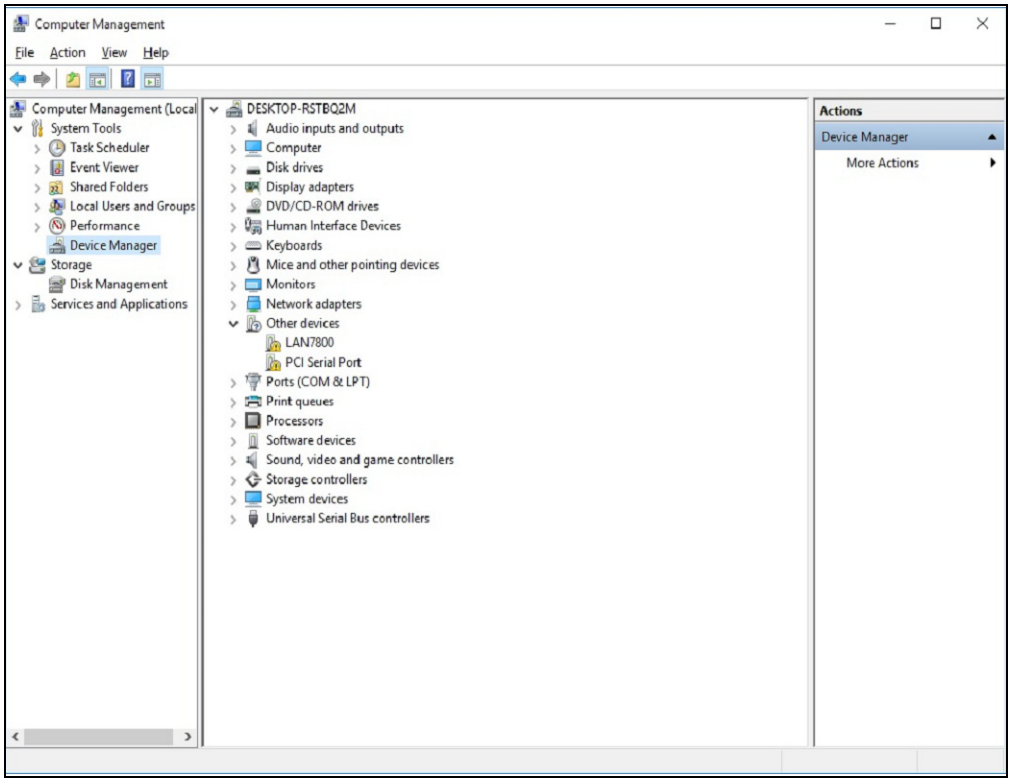


5. Click **OK**.

   The Device Manager window displays.

6. Select the device from the "Other devices" section, as displayed in Figure 5-7.

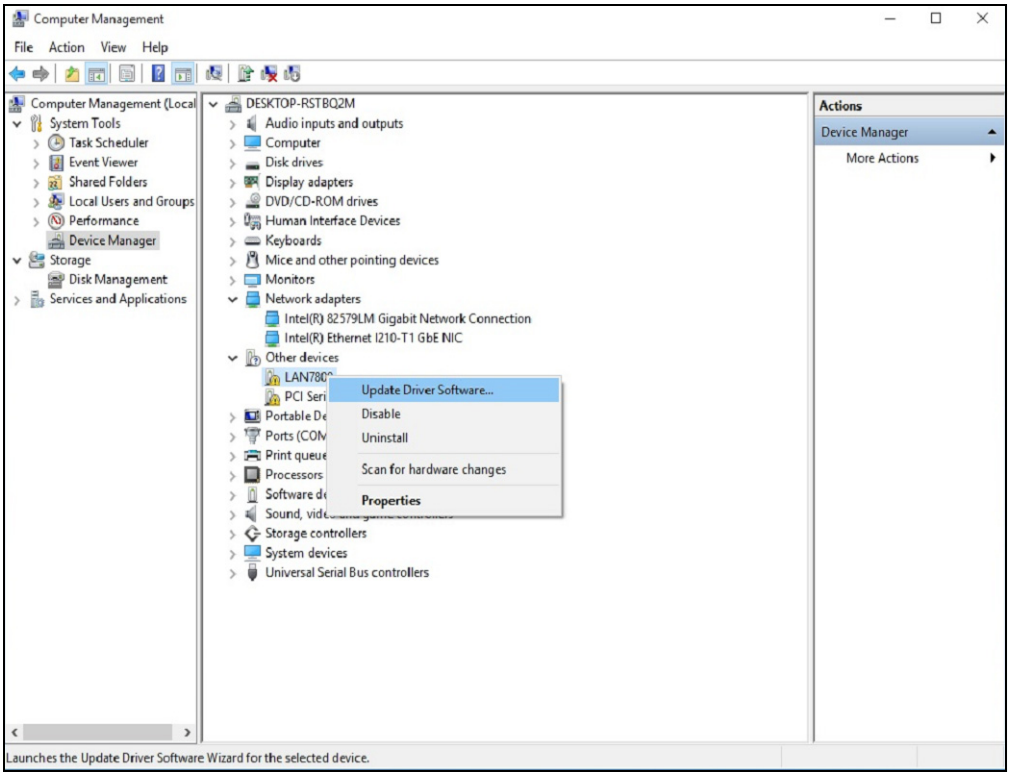
FIGURE 5-7: DEVICE MANAGER WINDOW - OTHER DEVICES



7. Right-click the device.

   A context menu displays, as in Figure 5-8.

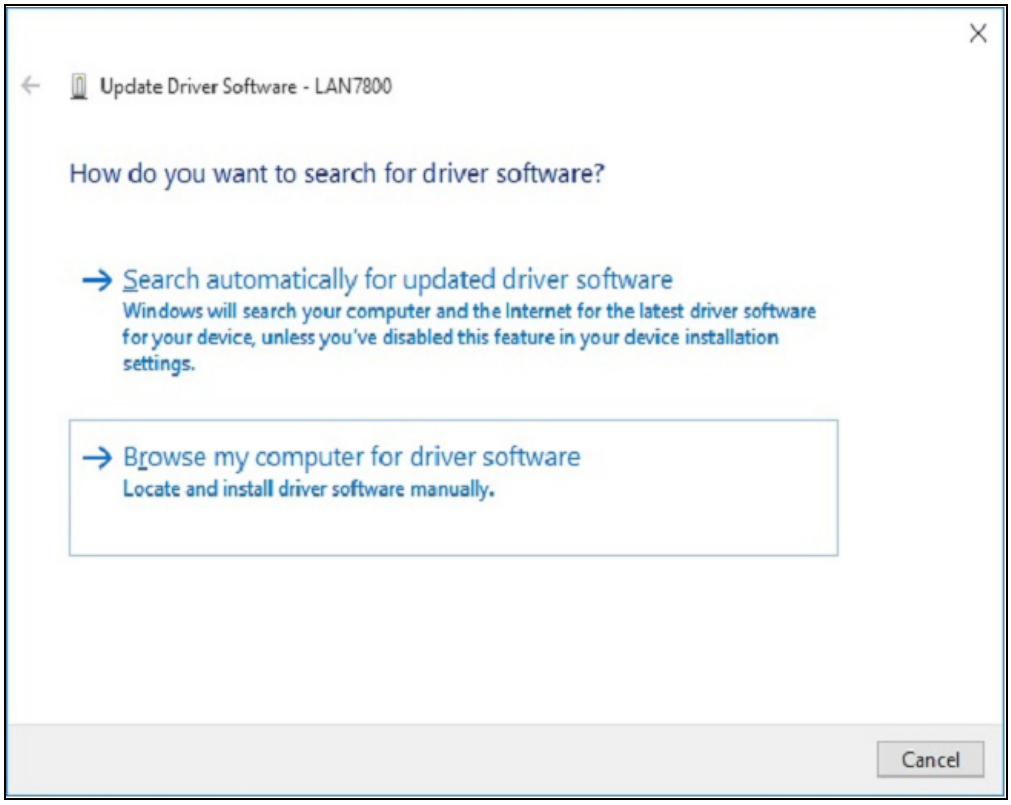
FIGURE 5-8: DEVICE MANAGER WINDOW - UPDATE DRIVER SOFTWARE



8. Click Update Driver Software.

   The Update Driver Software window displays, as in Figure 5-9.

FIGURE 5-9: DRIVER INSTALL OPTIONS WINDOW



9. Click **Browse my computer for driver software**.
10. Click **Browse…**.

    The Browse For Folder dialog displays.
11. Browse to the location of the copied Microchip INF distribution files.
12. Click **OK**.
13. Click **Next**.

Note:	Windows 10 installations must use the driver under the `ndis650` folder. For Windows 32-bit installations, browse to the `x86` folder within the `ndis650` folder of the Microchip INF distribution files, as displayed in Figure 5-10. For Windows 64-bit installations, browse to the `x64` folder within the `ndis650` folder of the Microchip INF distribution files, as displayed in Figure 5-11

FIGURE 5-10: BROWSE WINDOW - WINDOWS 10 32-BIT

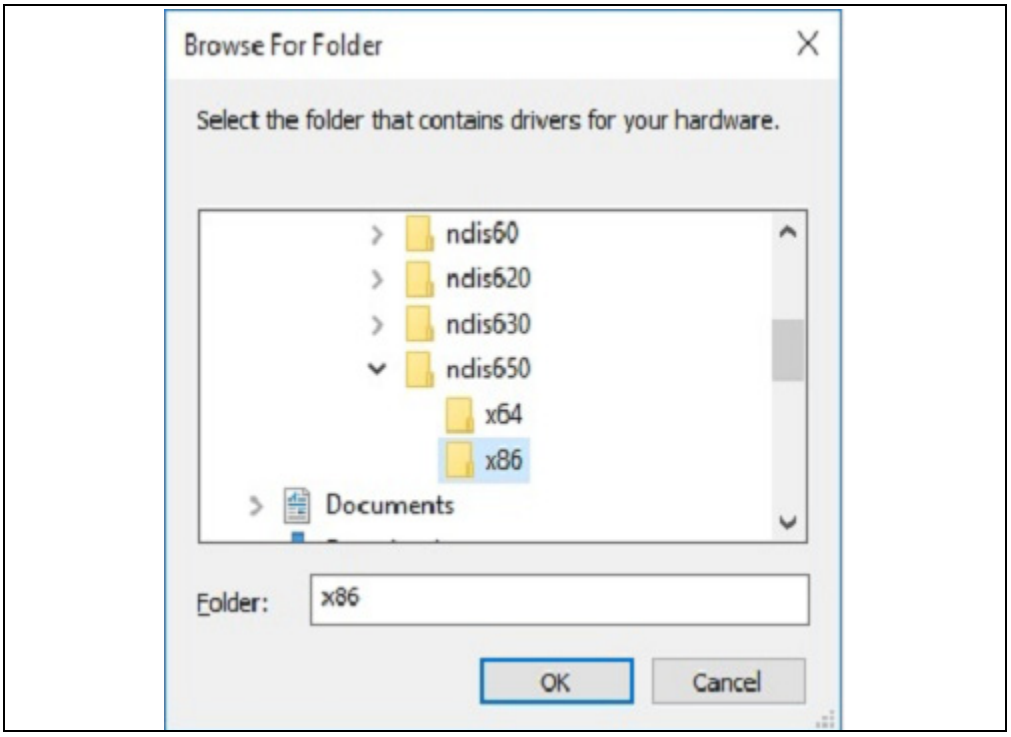
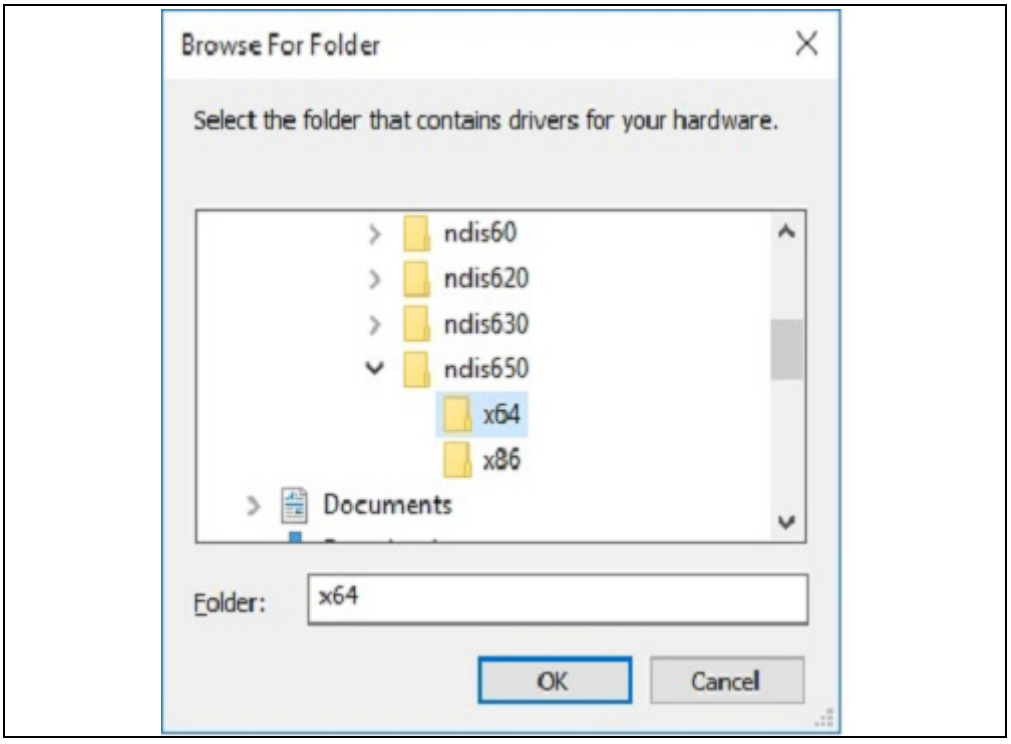
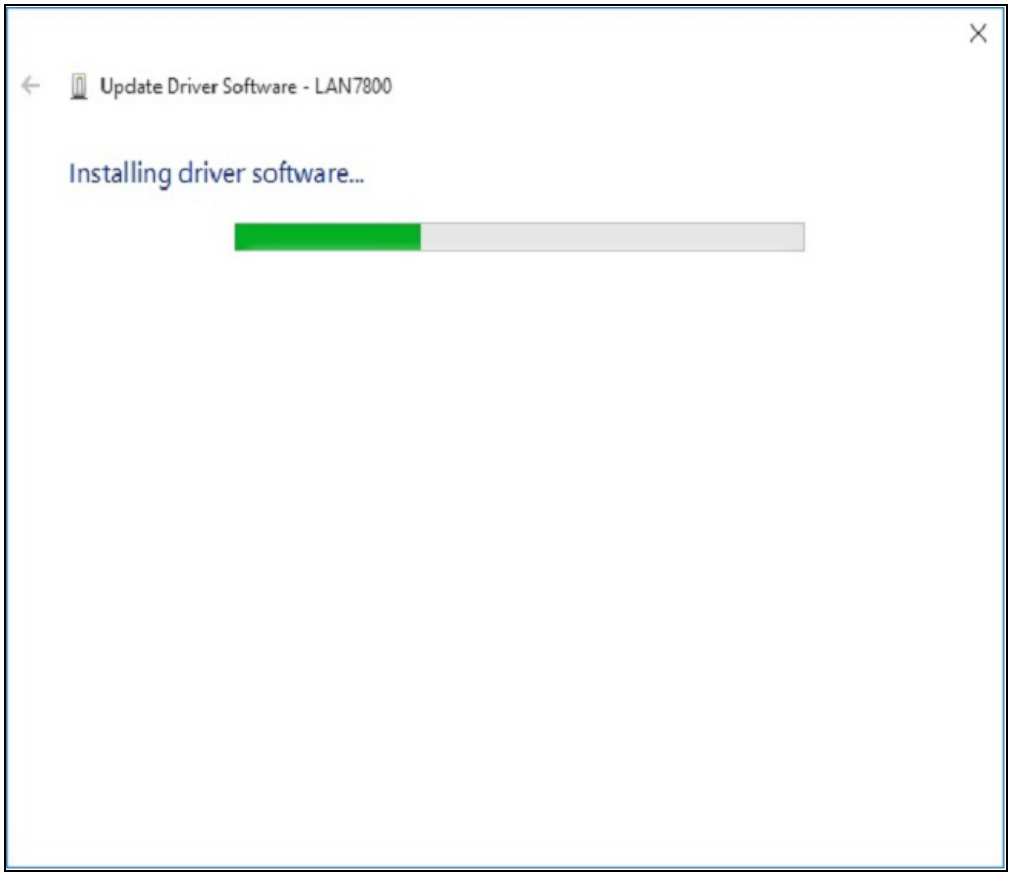


FIGURE 5-11: BROWSE WINDOW - WINDOWS 10 64-BIT



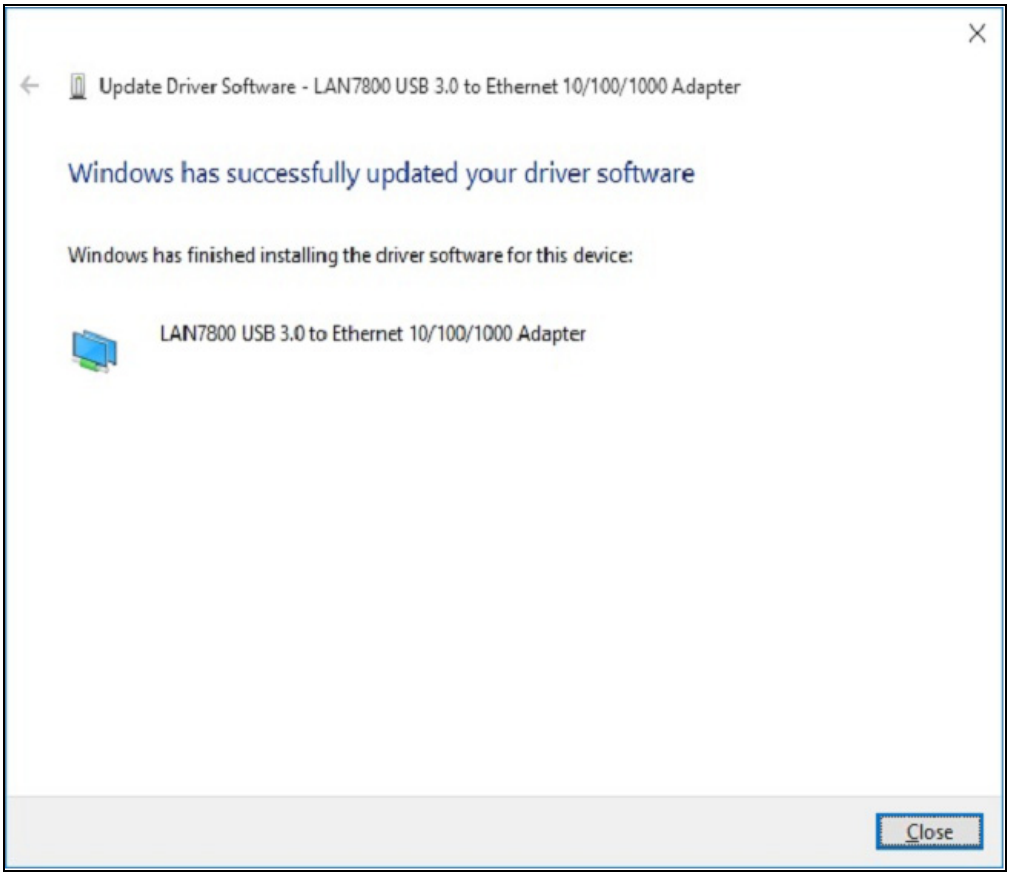
The installation progress displays, as in Figure 5-12. This indicates that some time may be necessary for the installation to complete.

FIGURE 5-12: INSTALLATION PROGRESS WINDOW



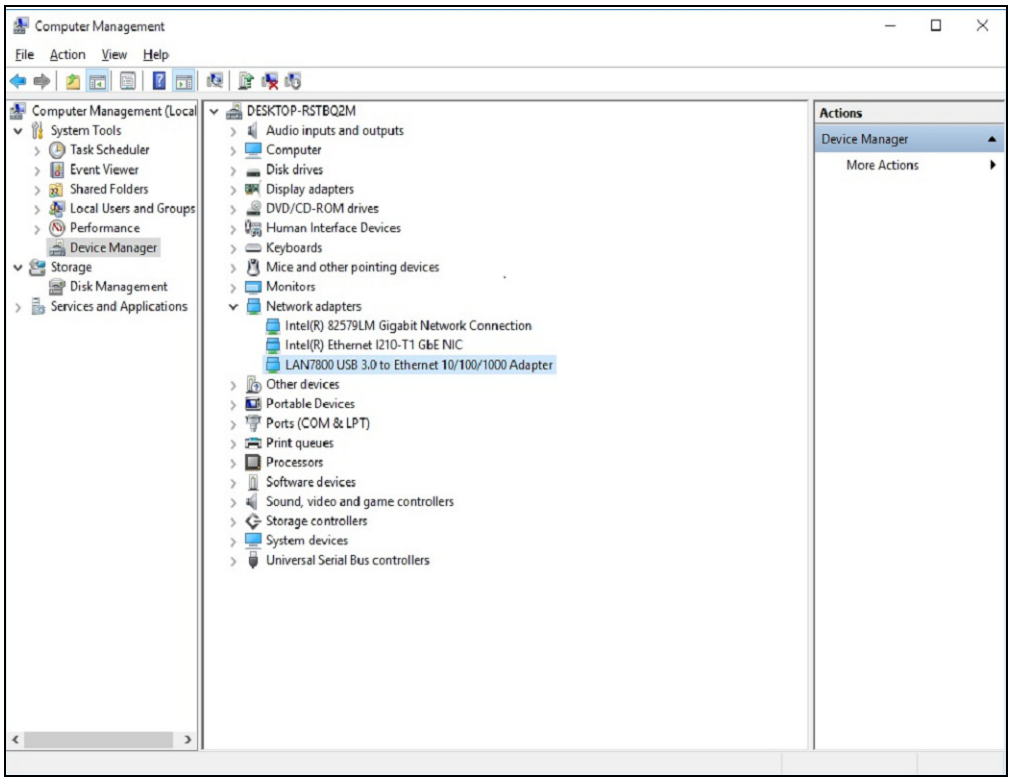
When finished, the installation progress window will change to indicate the driver has been installed. Click the "Close" button to complete the driver installation as displayed in Figure 5-13.

FIGURE 5-13: DEVICE DRIVER UPDATE COMPLETE SCREEN



The device driver is now listed in the Device Manager window, as displayed in Figure 5-14. The installation process is now complete.

FIGURE 5-14: DEVICE DRIVER INSTALLATION SUCCESS IN DEVICE MANAGER

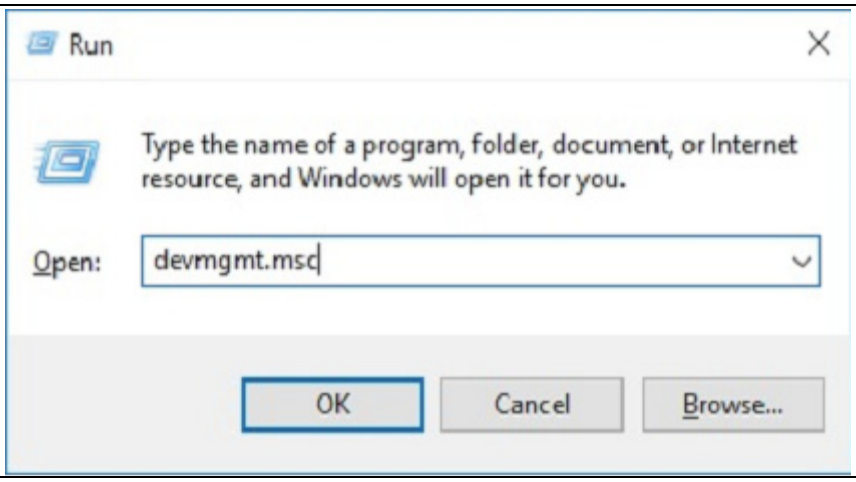


5.3 WINDOWS 10 32/64 BIT DRIVER UNINSTALLATION VIA DEVICE MANAGER

To uninstall the Windows 10 32/64-bit software:

1. From the Start menu, type `Run` in the search field.
   Results from the Programs list display.
2. Click **Run**.
   The Run dialog displays.
3. Type `devmgmt.msc`, as displayed in Figure 5-15.

FIGURE 5-15: RUN WINDOW



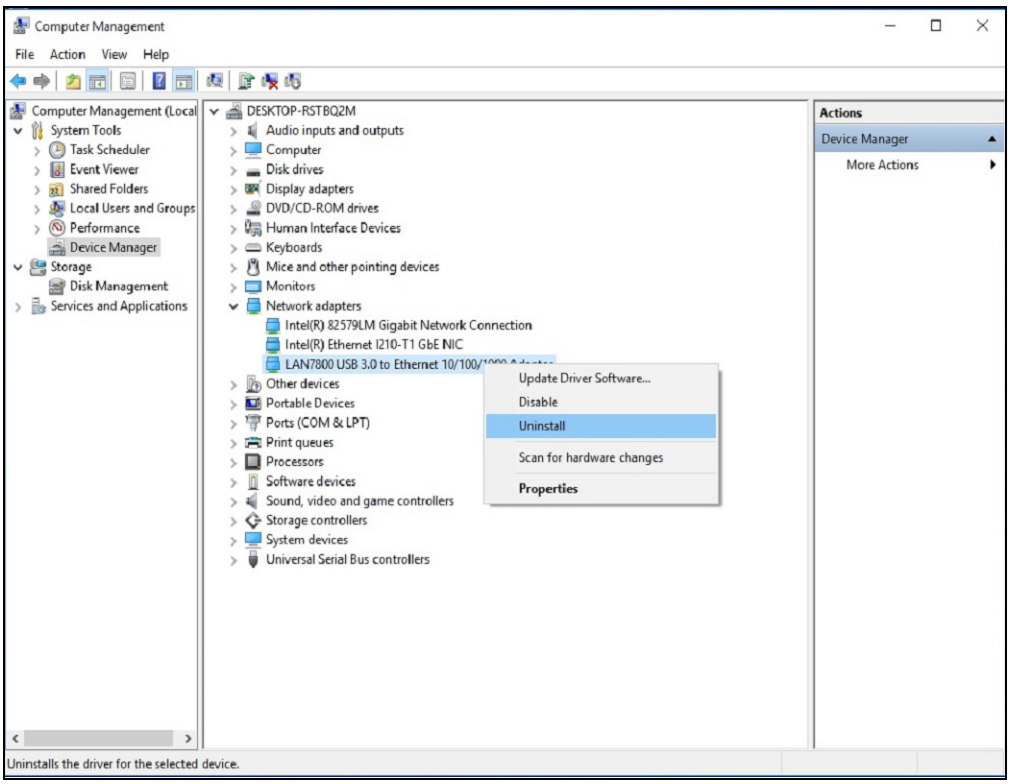
4. Click **OK**.

The Device Manager window displays.

5. Select the device from the “Network adapters” section.
6. Right-click the device.

   A context menu displays, as in Figure 5-16.

FIGURE 5-16: DEVICE MANAGER WINDOW

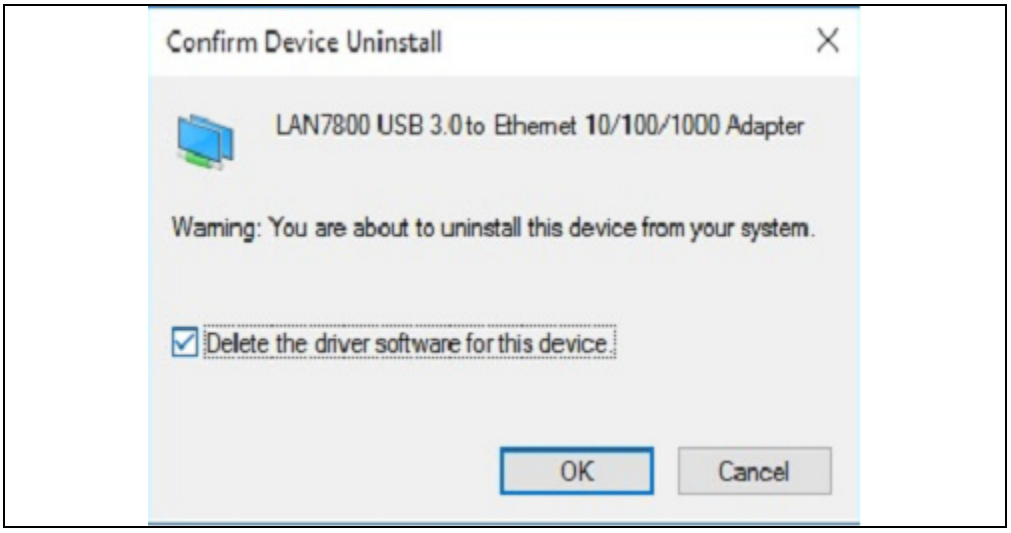


7. Click Uninstall.

   The Confirm Device Uninstall window displays

8. Select the “Delete the driver software for this device.” checkbox, as displayed in Figure 5-17.

FIGURE 5-17: CONFIRM DEVICE REMOVAL WINDOW



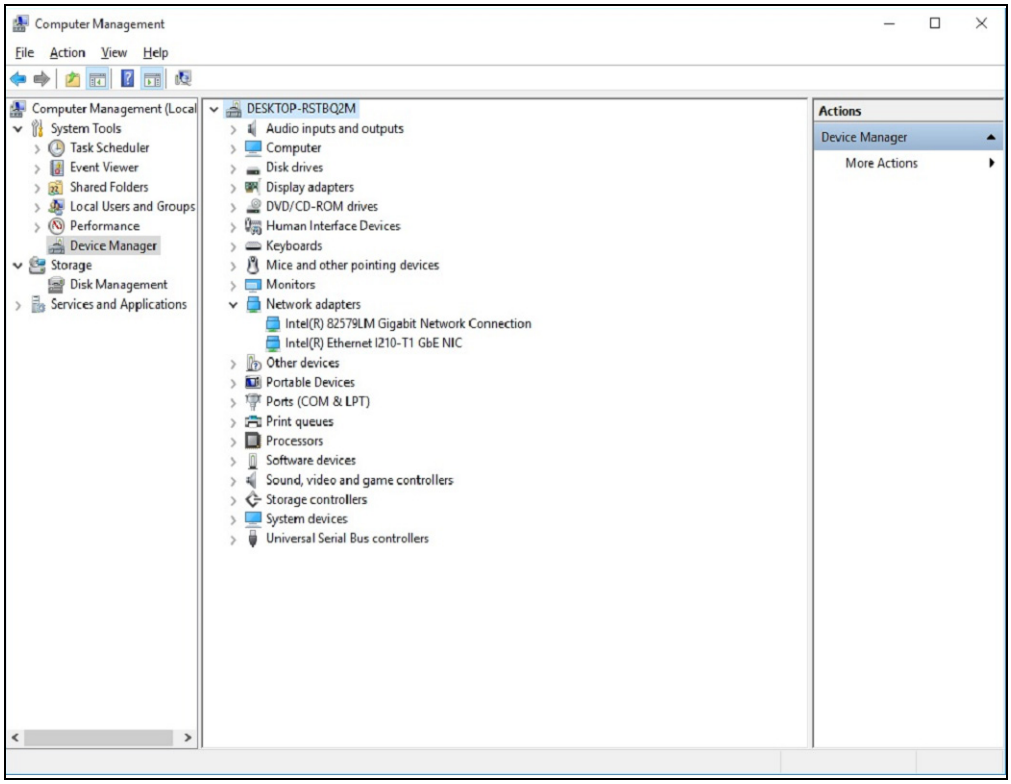
9. Click **OK**.

   The Confirm Device Uninstall window displays a progress indicator as the device drivers are being removed, as displayed in Figure 5-18.

FIGURE 5-18: DEVICE UNINSTALL PROGRESS WINDOW

The device driver is no longer listed in the Device Manager window, as displayed in Figure 5-19. The uninstallation process is now complete.

FIGURE 5-19: DEVICE REMOVED FROM DEVICE MANAGER



5.4 WINDOWS 10 32/64 BIT DRIVER UNINSTALLATION VIA CONTROL PANEL

This section details how to use the automated uninstaller using Windows 10's “Programs and Features”.

To uninstall the Windows 10 32/64-bit software:

1. From the Start menu, type `Run` in the search field.

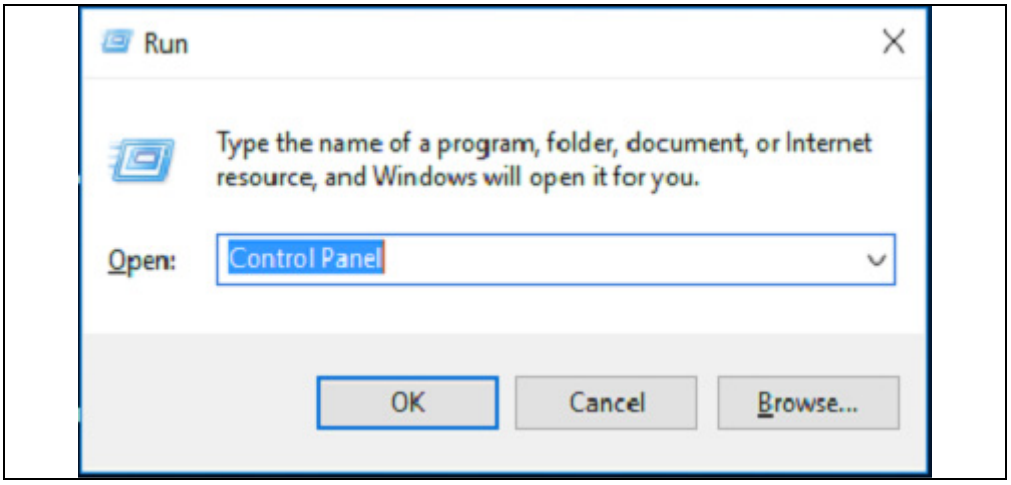
   Results from the Programs list display.

2. Click **Run**.

   The Run dialog displays.

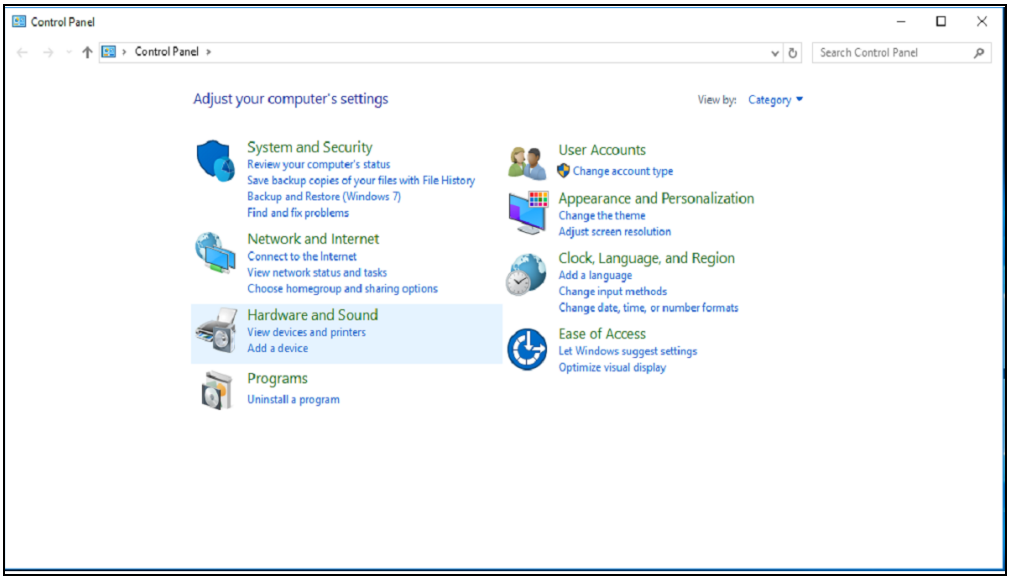
3. Type `control panel`, as displayed in Figure 5-20.

FIGURE 5-20: RUN WINDOW



The Control Panel window displays, as in Figure 5-21.

FIGURE 5-21: CONTROL PANEL WINDOW



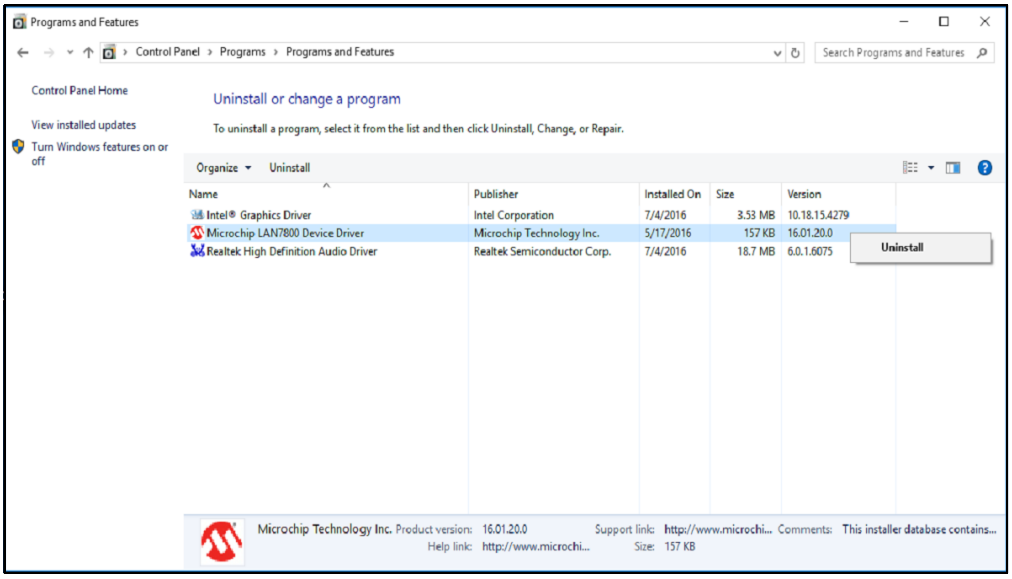
4. Click **Uninstall a program**.

   The Programs and Features window displays.

5. Right-click Microchip LAN7800 Device Driver.

   A context menu displays, as displayed in Figure 5-22.

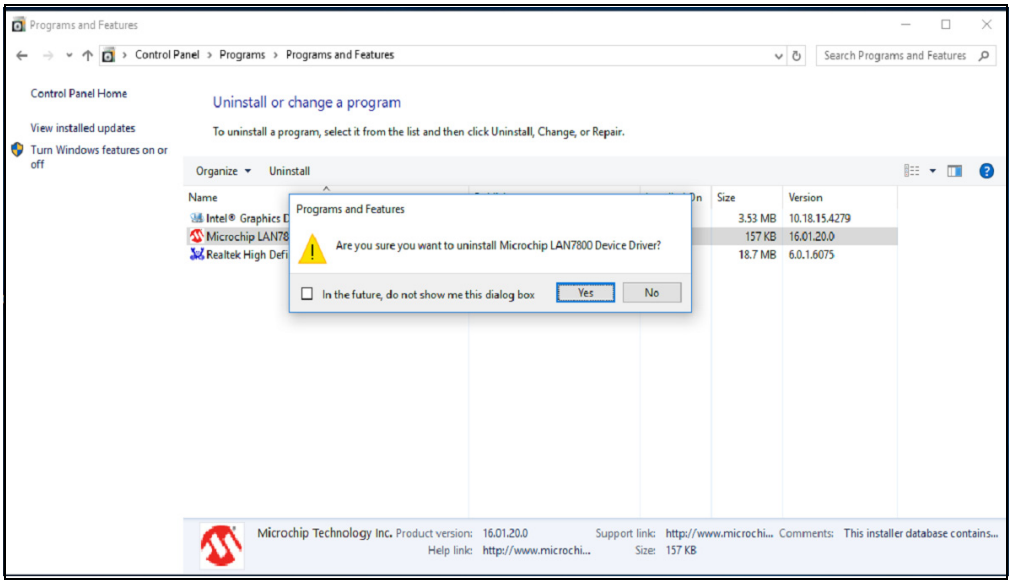
FIGURE 5-22: PROGRAMS AND FEATURES WINDOW



6. Click **Uninstall**.

   A confirmation dialog displays, as in Figure 5-23.

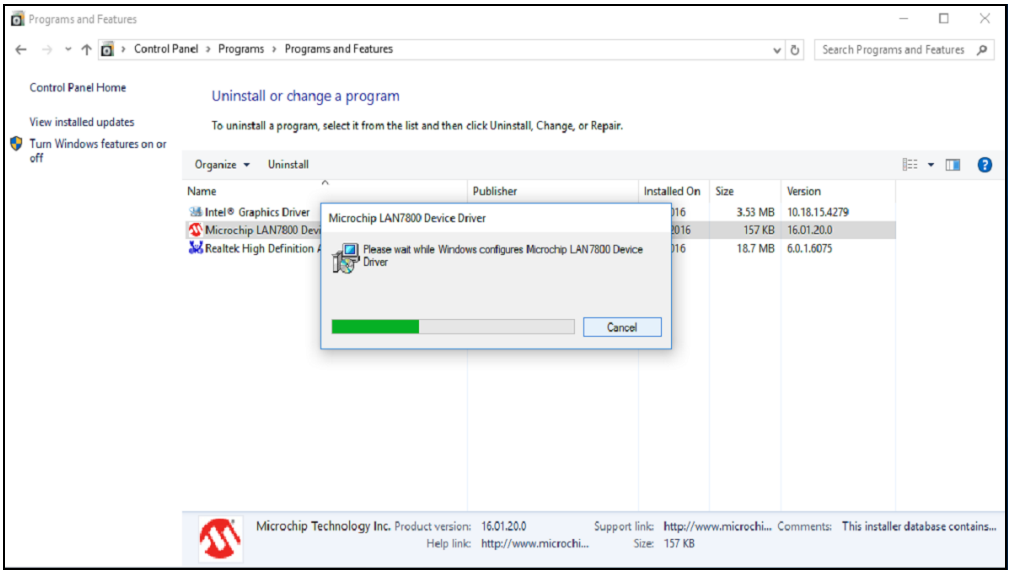
FIGURE 5-23: DRIVER UNINSTALL WINDOW



7. Click **Yes**.

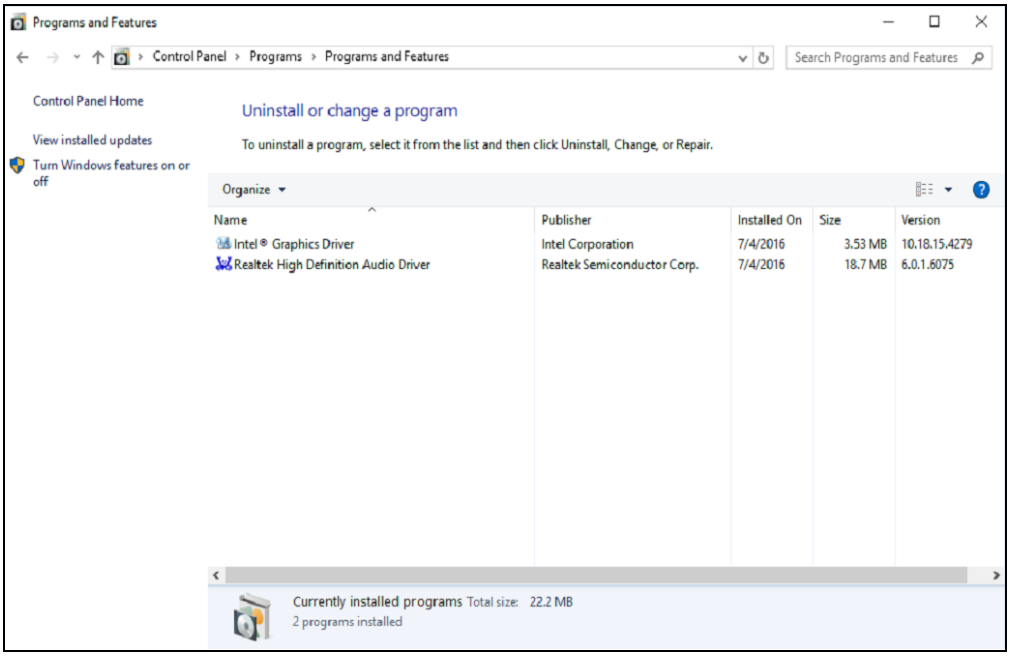
   The LAN7800 driver uninstalls, as displayed in Figure 5-24.

FIGURE 5-24: DRIVER UNINSTALL PROGRESS WINDOW



The LAN7800 driver Control Panel entry is removed, as displayed in Figure 5-25.

FIGURE 5-25: DRIVER REMOVED FROM CONTROL PANEL WINDOW



NOTES:

Chapter 6. MAC OS X Driver

6.1 MAC OS X DRIVER INSTALLATION

Copy the folder containing the distribution files to the desktop or any other convenient, known place within the directory structure. The folder contains the installer package and release notes file.

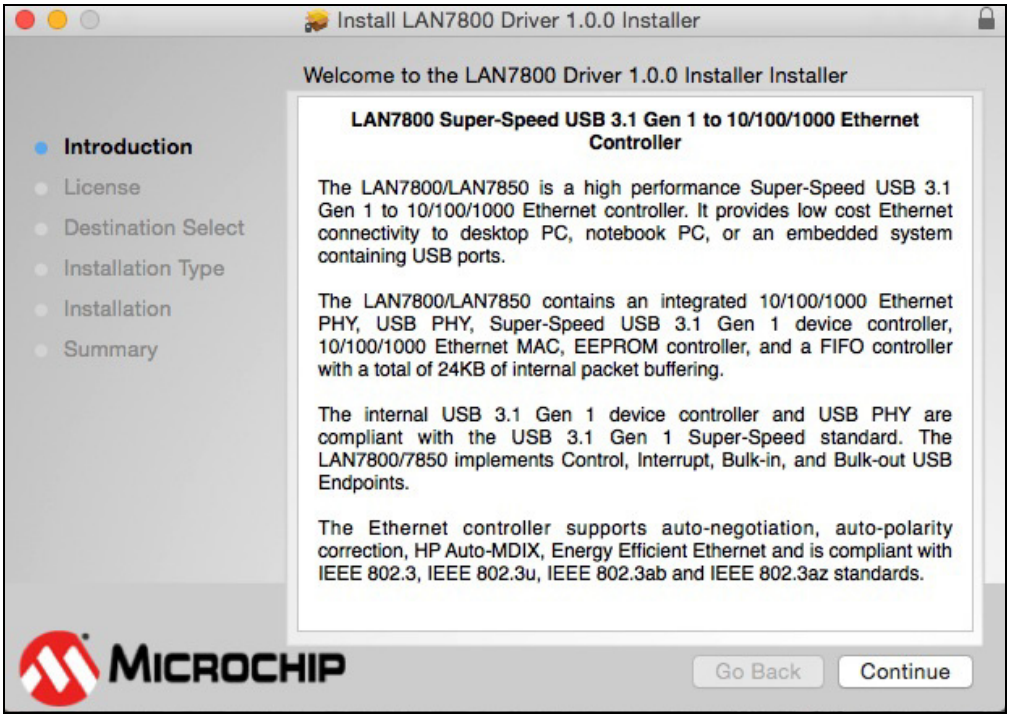
> **Note:** The device should not be plugged into the computer prior to installing the driver software.

To install:

1. Double click the installer icon.

   The introduction screen displays, as in Figure 6-1.

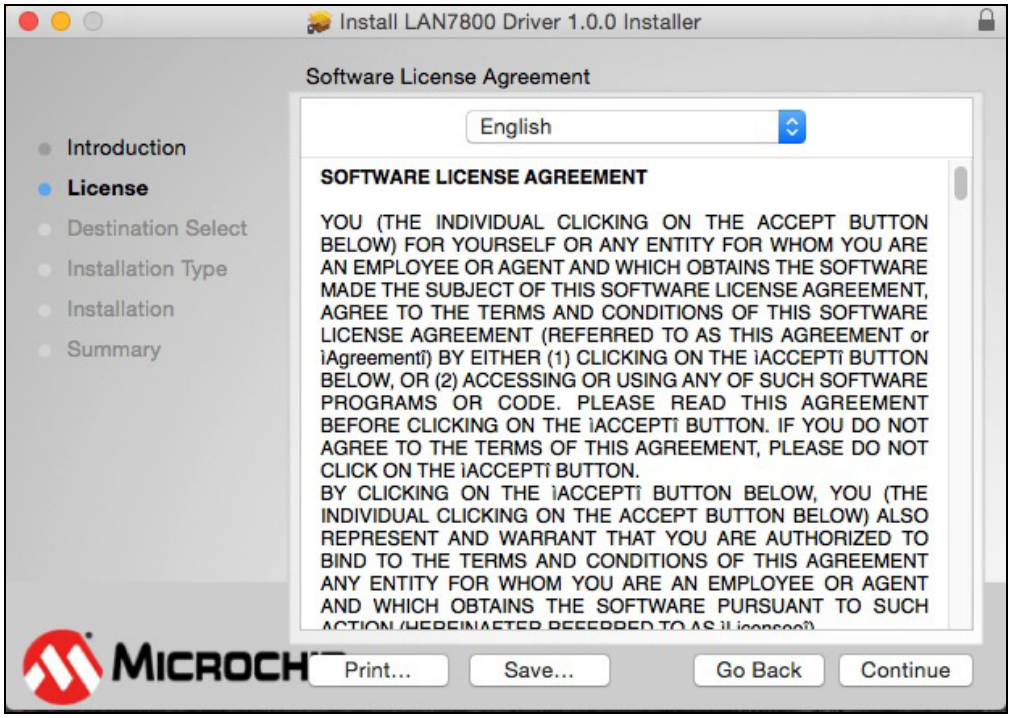
FIGURE 6-1: INTRODUCTION SCREEN



2. Click **Continue**.

   The License window displays, as in Figure 6-2.

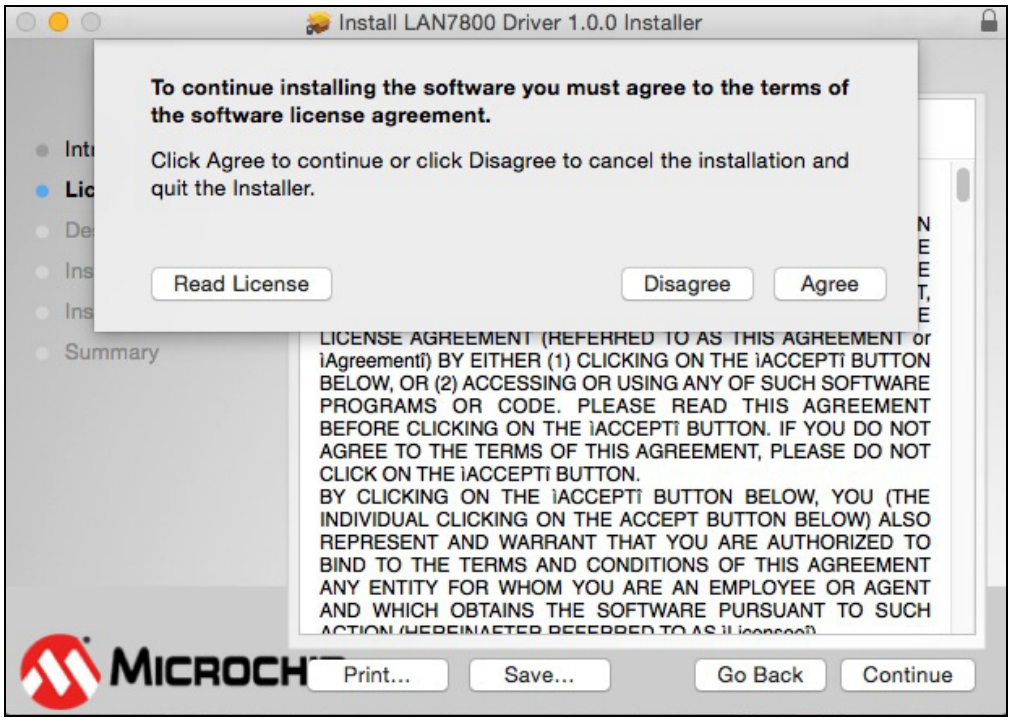
FIGURE 6-2: LICENSE WINDOW



3. Click Continue.

   A drop-down menu displays, as in Figure 6-3.

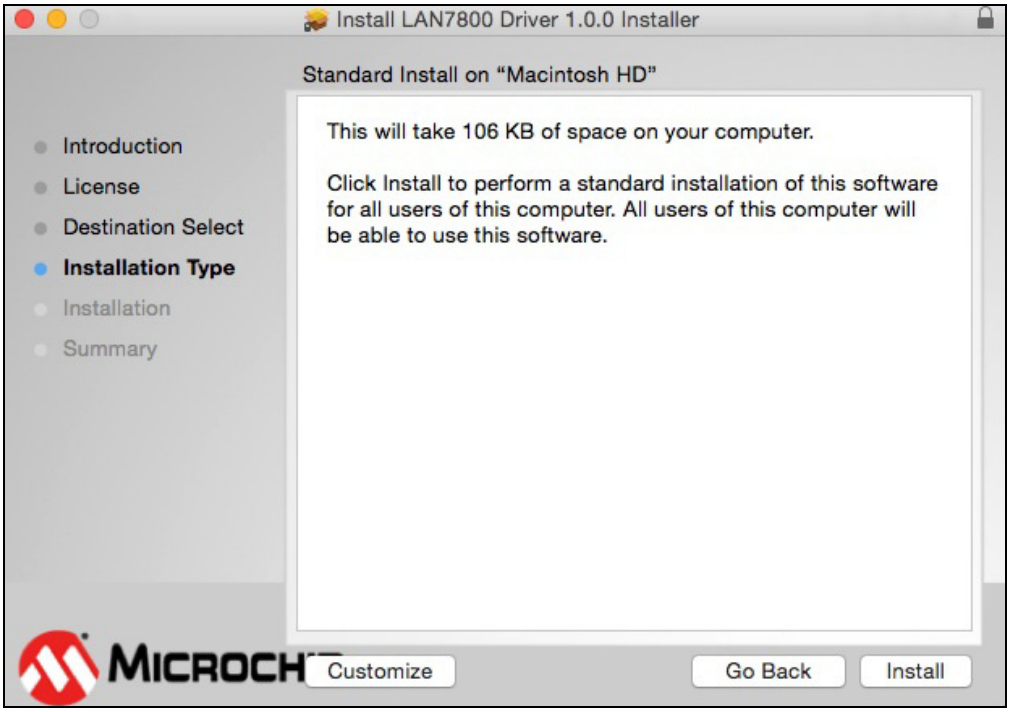
FIGURE 6-3: AGREE/DISAGREE DROP DOWN



4. Click **Agree**.

   The Installation Type window displays, as in Figure 6-4.

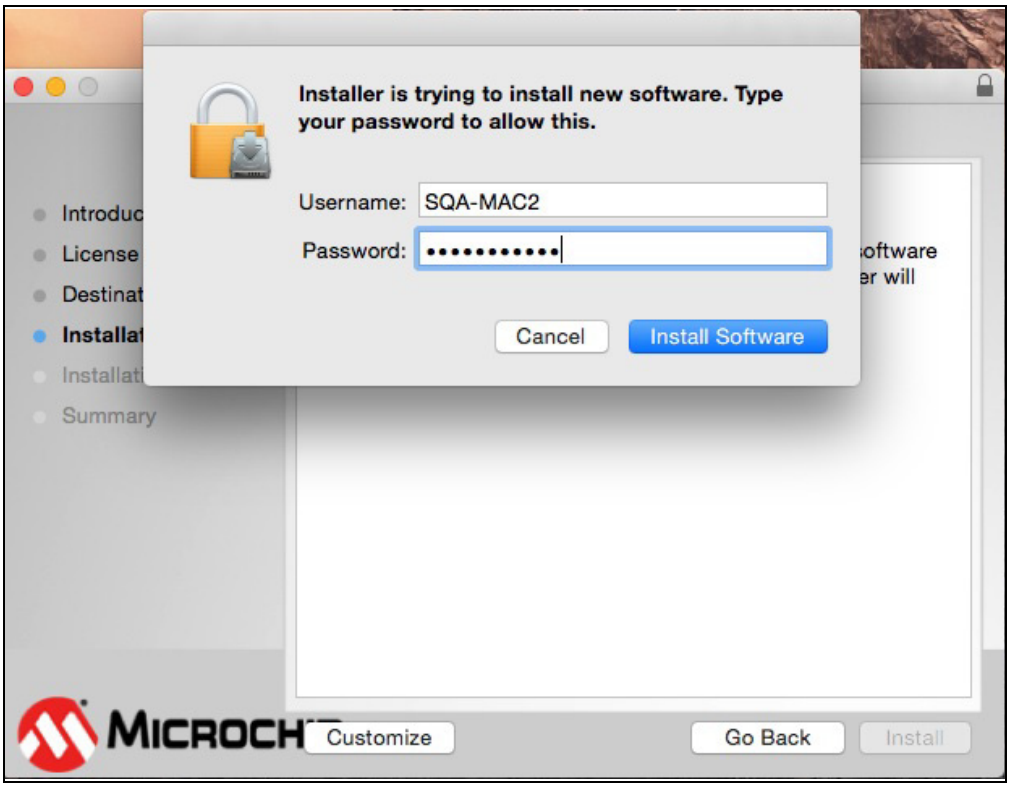
FIGURE 6-4: INSTALLATION TYPE WINDOW



5. Click **Install**.

   A drop-down requesting system administration rights displays, as in Figure 6-5.

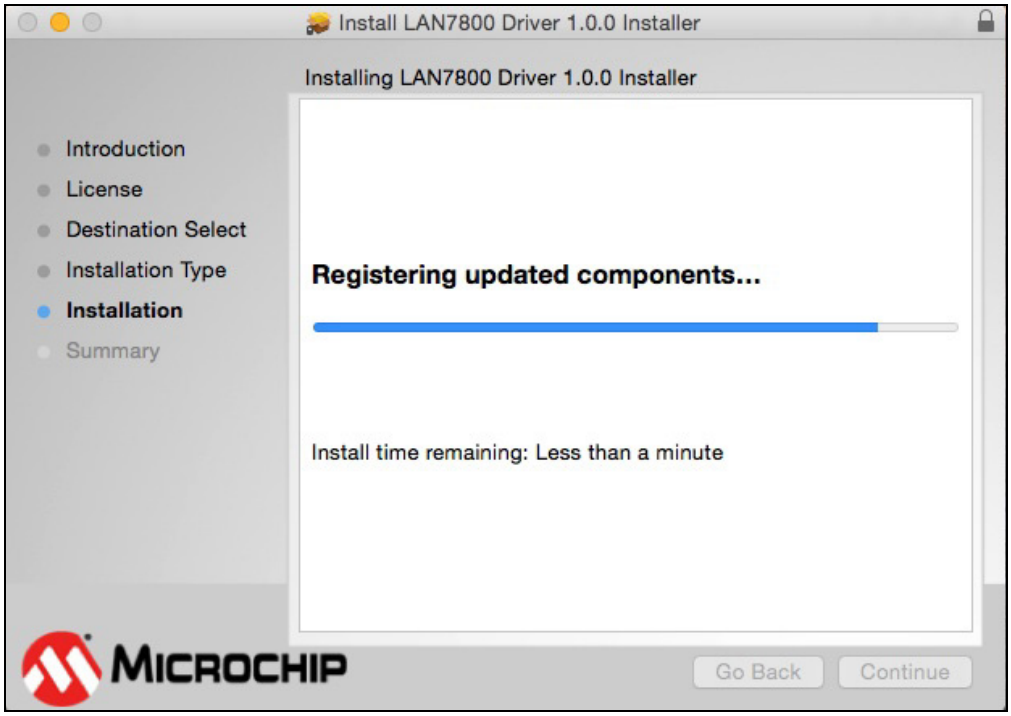
FIGURE 6-5: INSTALLER NAME/PASSWORD DROP DOWN



6. Enter the "Username" and "Password".

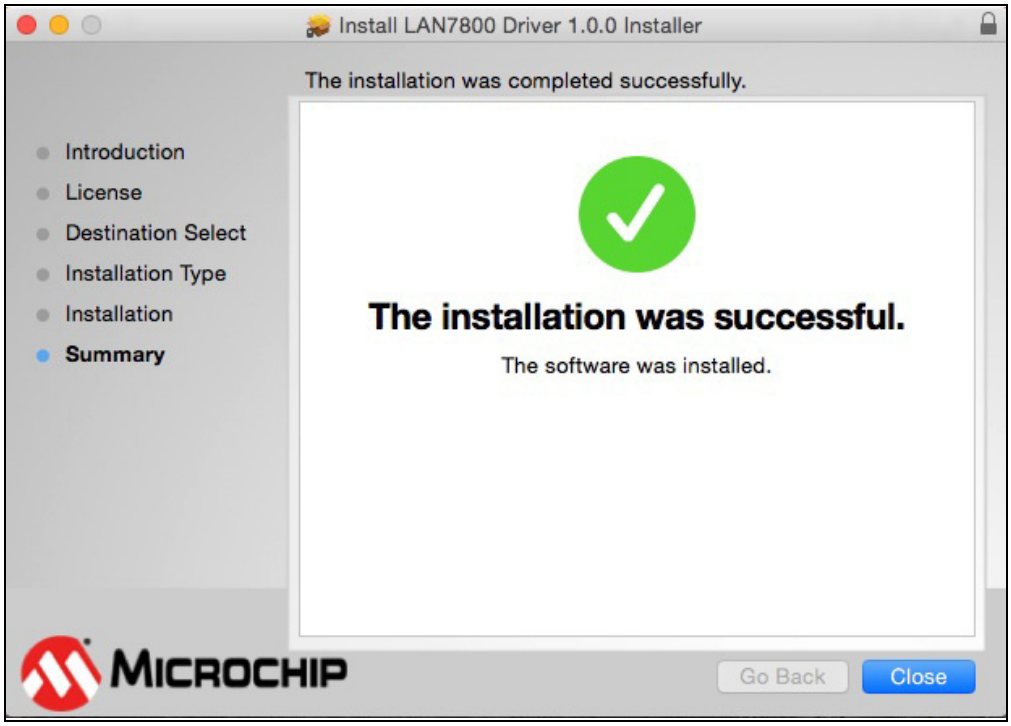
   The Installation window displays, as in Figure 6-6.

FIGURE 6-6: INSTALLATION WINDOW



Once the driver has been installed, the Summary window displays, as in Figure 6-7.

FIGURE 6-7: SUMMARY WINDOW INDICATING SUCCESSFUL INSTALLATION

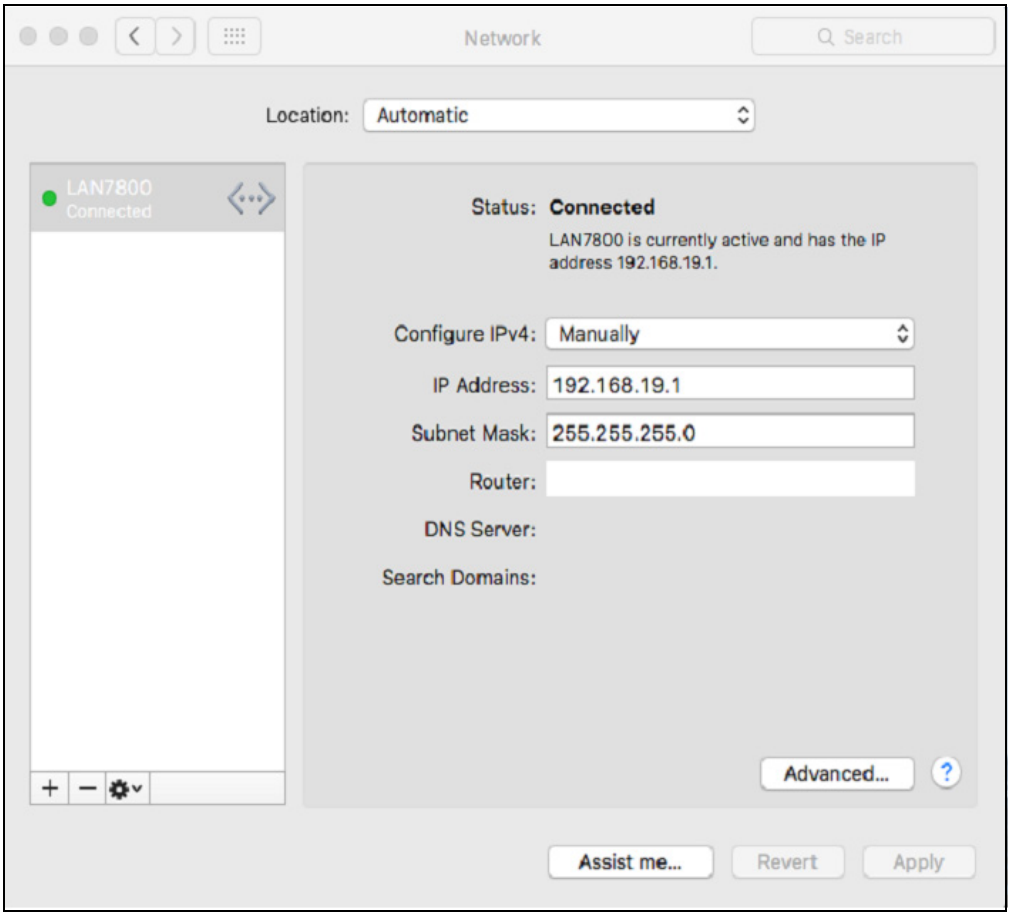


7. Click **Close**.
   The driver software installation is complete.
8. Plug the device into an available USB port.

The Network window displays.

9. Assign the IP address manually, as similar to details displayed in Figure 6-8.

FIGURE 6-8: DEVICE CONNECTED WITH MANUAL IP ADDRESS



6.2 MAC OS X DRIVER UNINSTALLATION

Note: Unplug the device before uninstalling. If the device cannot be removed from the system, use `sudokextunload/Library/Extensions/LAN7800.kext` command to force unload the driver.

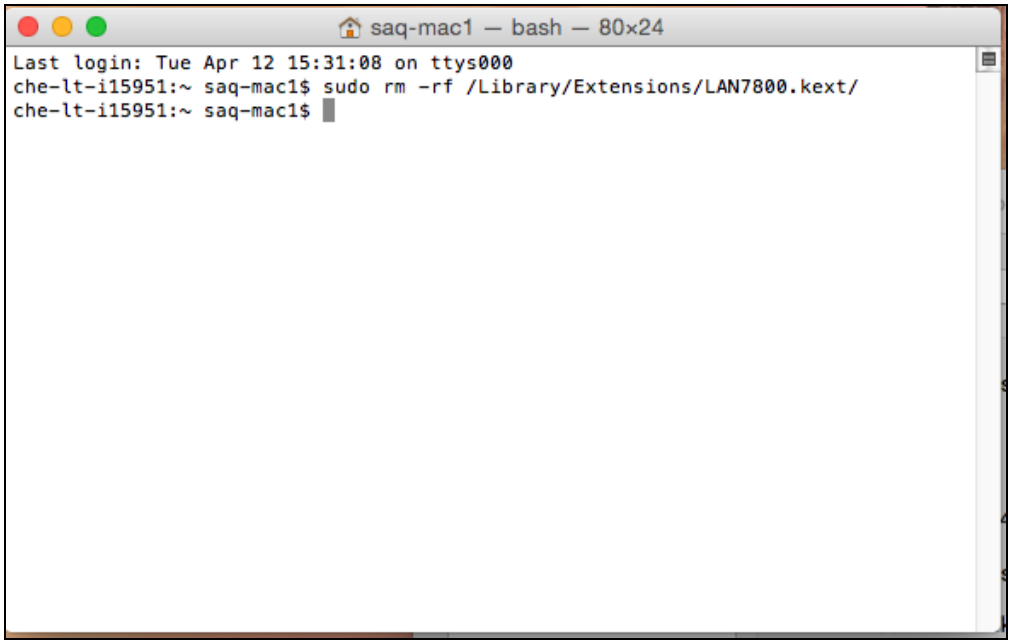
To uninstall:

1. Open a command terminal.
2. Enter the removal command below, as displayed in Figure 6-9.

   ```
   sudo rm -rf /Library/Extensions/LAN7800.kext/
   ```

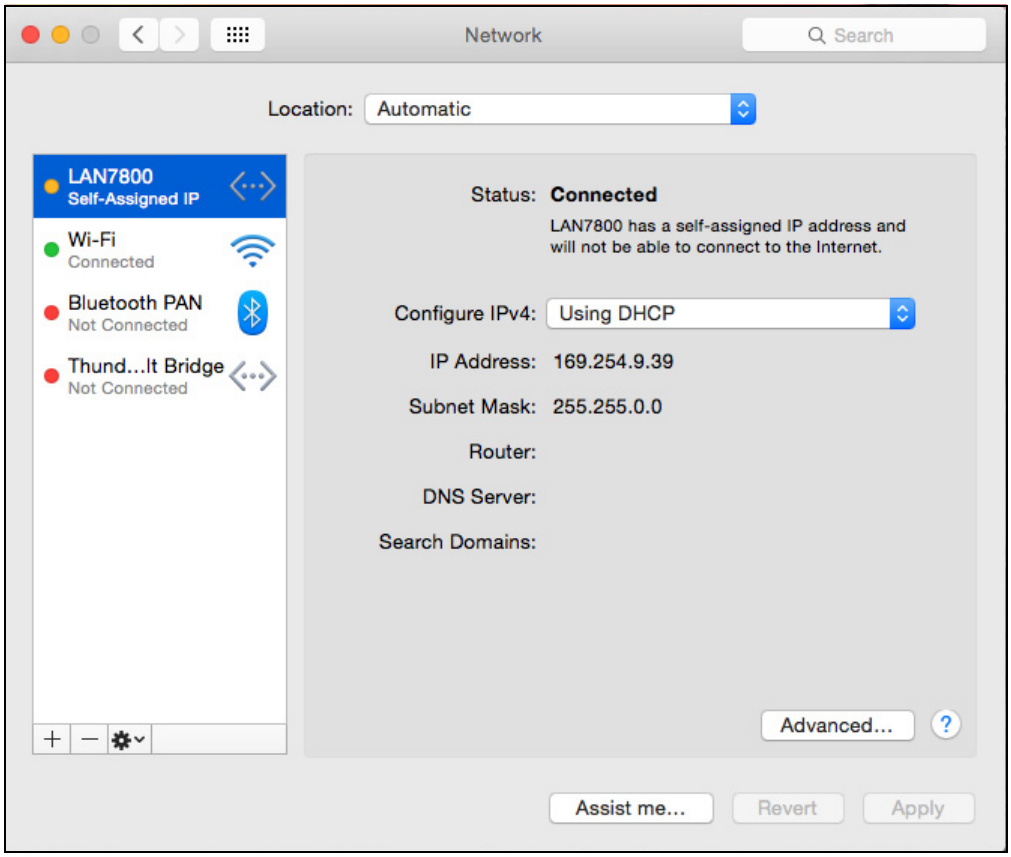
   This removes the device driver from the operating system.

FIGURE 6-9: DRIVER REMOVAL



The Network window displays, as in Figure 6-10.

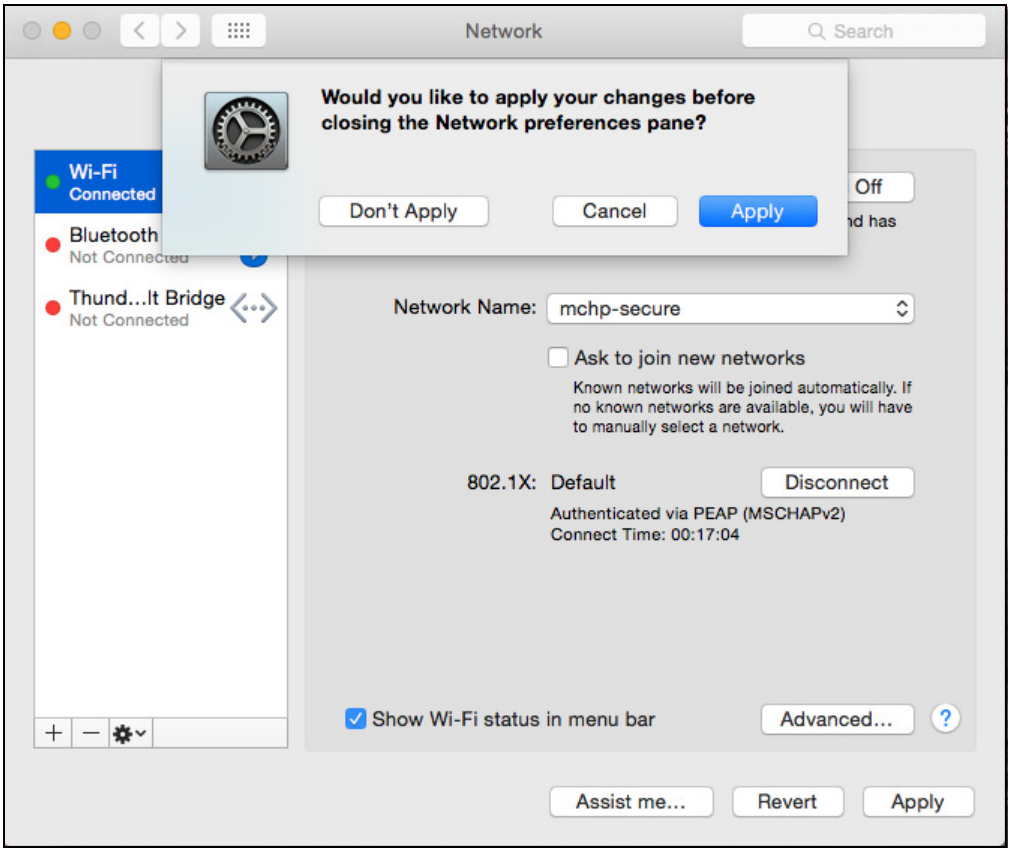
FIGURE 6-10: NETWORK PREFERENCES



3. Click the LAN7800 network entry to select and highlight it.
4. Click the "-" button.

   A prompt to confirm and apply the changes displays, as in Figure 6-11.

FIGURE 6-11: CONFIRM CHANGES



5. Click **Apply**.

   The device uninstallation is complete.

NOTES:

Chapter 7. Linux Driver

LAN78xx driver was included in kernel tree since kernel version 4.3. Any Linux distributions which use kernel 4.3 or later should have LAN78xx driver.

7.1 CHECKING THE KERNEL VERSION

After system boot up, open terminal program. Type the command `uname -r` to display the kernel version. If the kernel version is earlier than 4.3, update to latest Linux distributions or kernel version.

7.2 DRIVER LOCATION

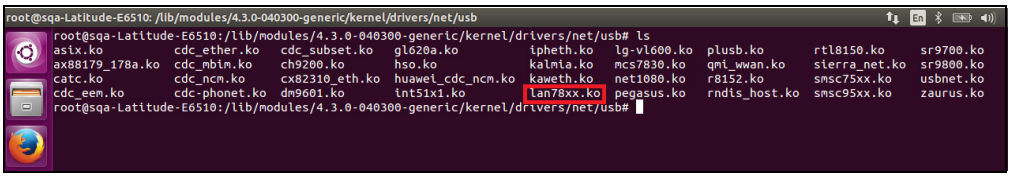
7.2.1 LAN78xx MAC Driver

Depending on kernel configuration, driver may be a built-in kernel image or exist as module. Most Linux distributions include LAN78xx driver as module.

The folder would be `/lib/modules/kernel_version/kernel/driver/net/usb` (where *`kernel_version`* is the kernel version, such as *`4.4.0-40-generic`*).

User can go to the folder by `/lib/modules/'uname -r'/kernel/drivers/net/usb` (*`'uname -r'`* command will be expanded to kernel version such as *`4.4.0-40-generic`*). `lan78xx.ko` is the driver for LAN78xx, as displayed in Figure 7-1.

FIGURE 7-1: LAN78XX MAC DRIVER



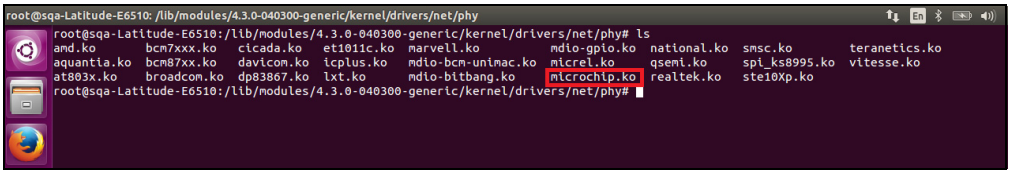
7.2.2 LAN78xx PHY driver

In addition to `lan78xx.ko`, LAN78xx requires phy driver which is located in `/lib/modules/'uname -r'/kernel/driver/phy`.

The file `microchip.ko` is required to operate LAN7800 & LAN7850, as displayed in Figure 7-2.

When drivers are installed correctly, those drivers will be loaded automatically in the event of USB connections.

FIGURE 7-2: LAN78XX PHY DRIVER

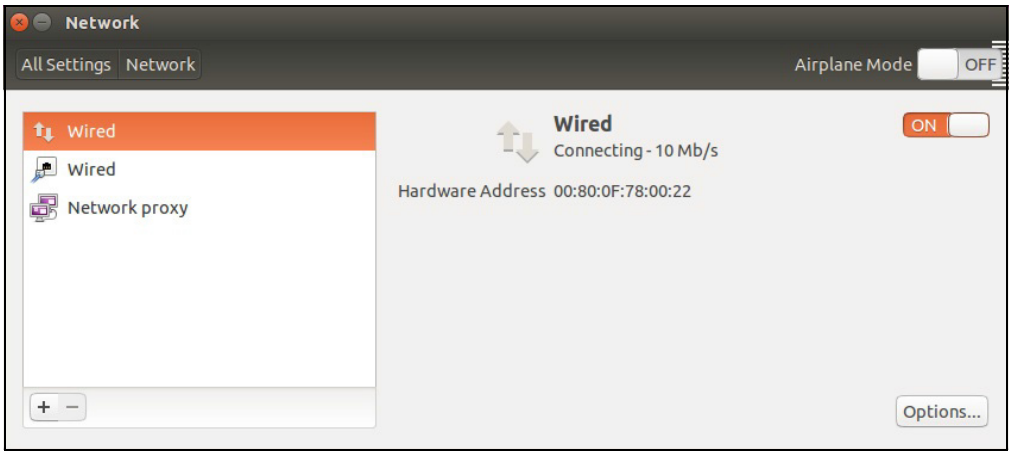


7.3 HARDWARE CONFIGURATION

LAN7800/LAN7800i device must now be plugged into an available USB port on the computer. Once the device is plugged in, follow the steps below:

1. Click *Settings>Network* on the task bar to bring up the Network Configuration window, as displayed in Figure 7-3.

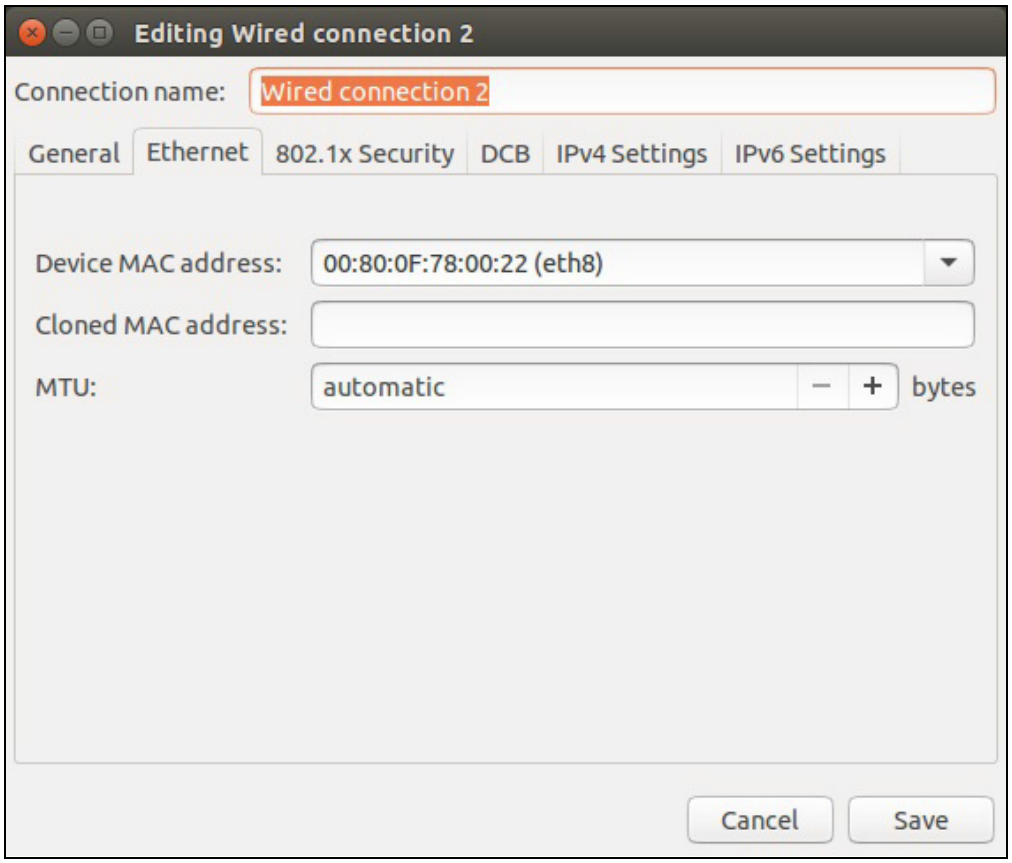
FIGURE 7-3: NETWORK CONFIGURATION WINDOW



2. Select the device and click **Options**.

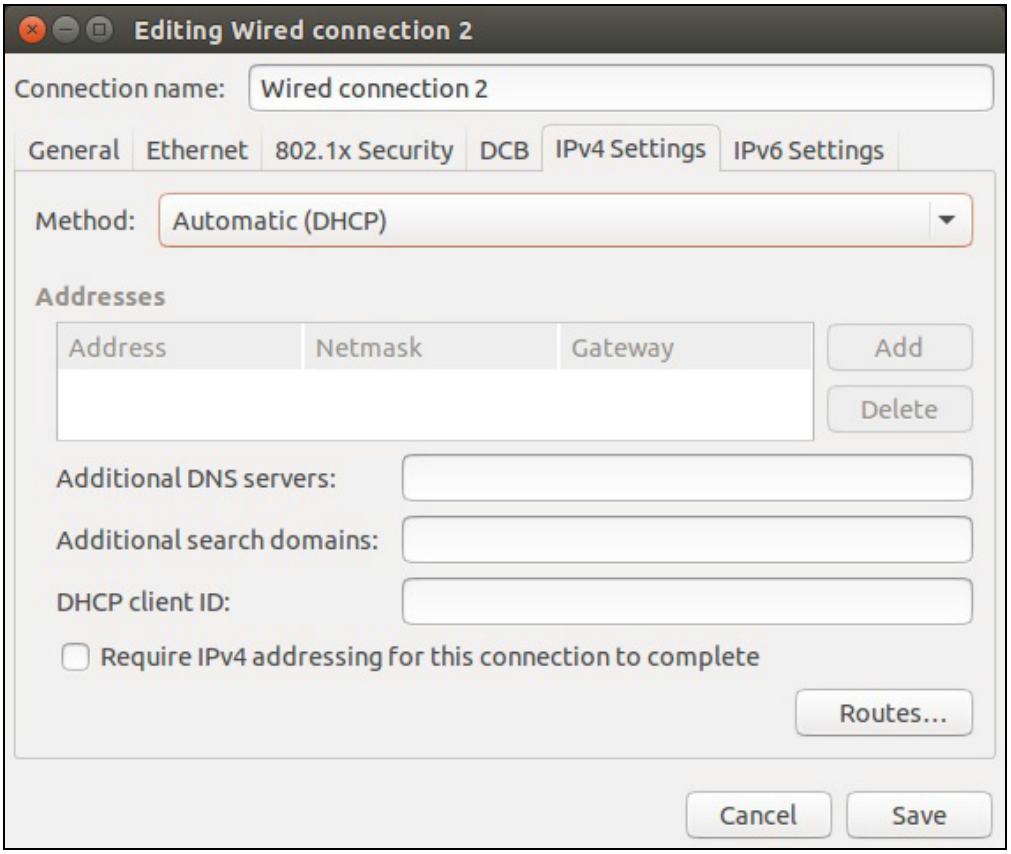
   The connection window displays, as in Figure 7-4.

FIGURE 7-4: DEVICE SELECTION



3. Select **Ipv4 Settings** tab as illustrated in Figure 7-5.

FIGURE 7-5: ADDRESS GENERATION



The new device must be configured to use DHCP or a fixed IP address, depending on the network configuration. Specification of this portion of the configuration is outside the scope of this document. For illustration purposes, automatic IP address generation is selected in Figure 7-6.

FIGURE 7-6: DEVICE CREATION

7.4 DRIVER UPDATE

Linux kernels and drivers get updated regularly, and the latest Linux kernel sources are located at https://www.kernel.org, while Linux kernel git is located at https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git.

To build a newer kernel, see references such as https://kernelnewbies.org/KernelBuild, or a Linux distribution reference such as https://wiki.ubuntu.com/Kernel/BuildYourOwnKernel.

> **Note:** When backporting the LAN78xx driver, remember that the Linux kernel is updated in various codes and data structures. As such, backporting may require more than just copying the newer version over the current one.

7.5 USB VENDOR ID & PRODUCT ID CHANGE

In case driver needs to be modified to support different USB Vendor ID & Product ID than default Microchip Vendor ID (0x0424) and LAN78XX Product ID (0x7800, 0x7850 and future products), please modify `product[]` indicated in `drivers/net/usb/lan78xx.c` as follows:

```
static const struct usb_device_id products[] = {
    {
    /* LAN7800 USB Gigabit Ethernet Device */
    USB_DEVICE(LAN78XX_USB_VENDOR_ID, LAN7800_USB_PRODUCT_ID),
    },
    {
    /* LAN7850 USB Gigabit Ethernet Device */
    USB_DEVICE(LAN78XX_USB_VENDOR_ID, LAN7850_USB_PRODUCT_ID),
    },
    {},
};
```

Customer can add their own Vendor & Product ID like sample below:

```
static const struct usb_device_id products[] = {
    {
    /* LAN7800 USB Gigabit Ethernet Device */
    USB_DEVICE(LAN78XX_USB_VENDOR_ID, LAN7800_USB_PRODUCT_ID),
    },
    {
    /* LAN7850 USB Gigabit Ethernet Device */
    USB_DEVICE(LAN78XX_USB_VENDOR_ID, LAN7850_USB_PRODUCT_ID),
    },
    {
    /* Sample of custom programmed Vendor & Product ID */
    USB_DEVICE(0x1234, 0x5678),
    },
    {},
};
```

> **Note:** Don't forget to configure EEPROM accordingly to be reflected into USB Device Descriptor fields.

Chapter 8. UEFI Driver

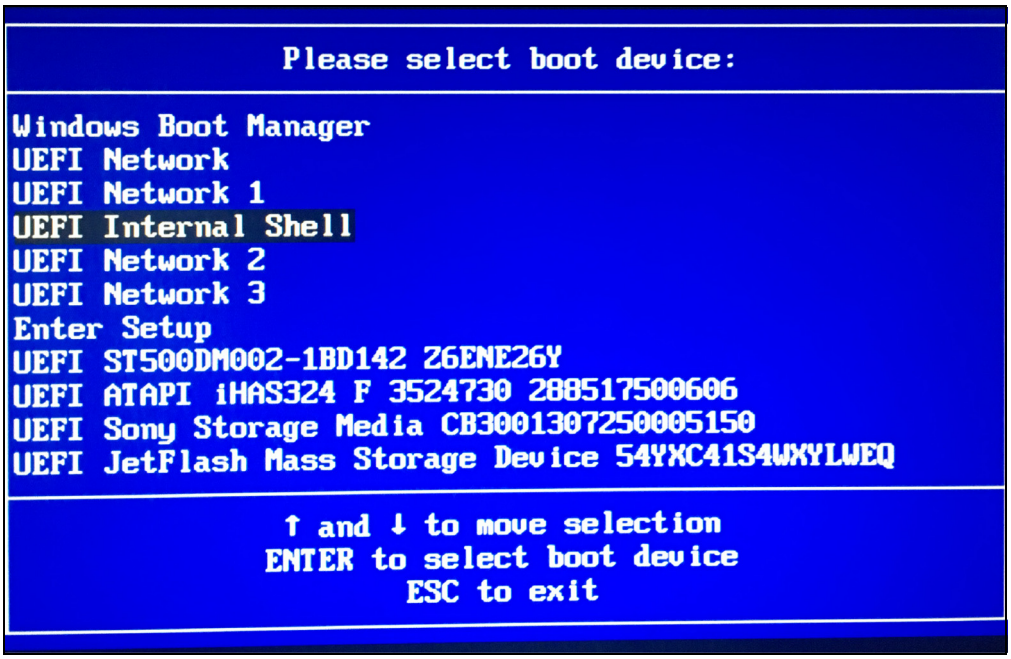
This chapter details the load and unload of the Unified Extensible Firmware Interface (UEFI) driver.

8.1 LOAD UEFI DRIVER

The UEFI LAN7800 software driver is distributed as a ZIP archive file containing all source files and release notes.

Copy the folder containing the distribution files to the desktop or any other convenient, known place within the directory structure.

FIGURE 8-1: UEFI INTERNAL SHELL



To install:

1. Connect the LAN78XX device.
2. Execute the command as displayed in the Figure 8-2.

   This loads the LAN78XX UEFI driver.

FIGURE 8-2: LOAD UEFI DRIVER

```
FS0:\LAN78xx_UEFI_Driver_v01\X64\> load Lan7800Driver.efi
Image 'FS0:\LAN78xx_UEFI_Driver_v01\X64\Lan7800Driver.efi' loaded at BADBA000 -
Success
FS0:\LAN78xx_UEFI_Driver_v01\X64\> _
```

3. Type the command as in Figure 8-3.

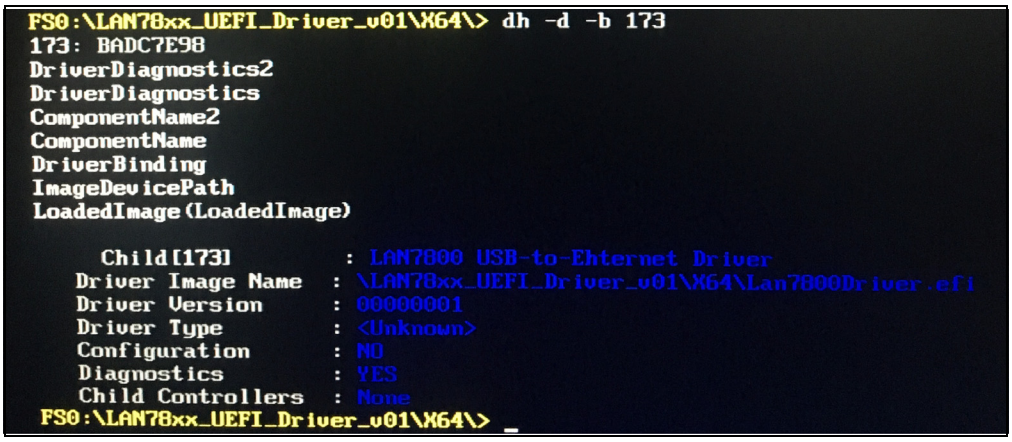
   This shows the installed drivers in the UEFI.

FIGURE 8-3: LAN78XX UEFI DRIVER

4. Execute the command as in Figure 8-4.

   This displays the installed LAN78XX driver details.

FIGURE 8-4: LAN78XX UEFI DRIVER DETAILS

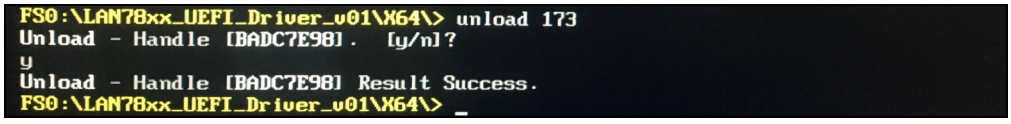


8.2 UNLOAD UEFI DRIVER

To uninstall:

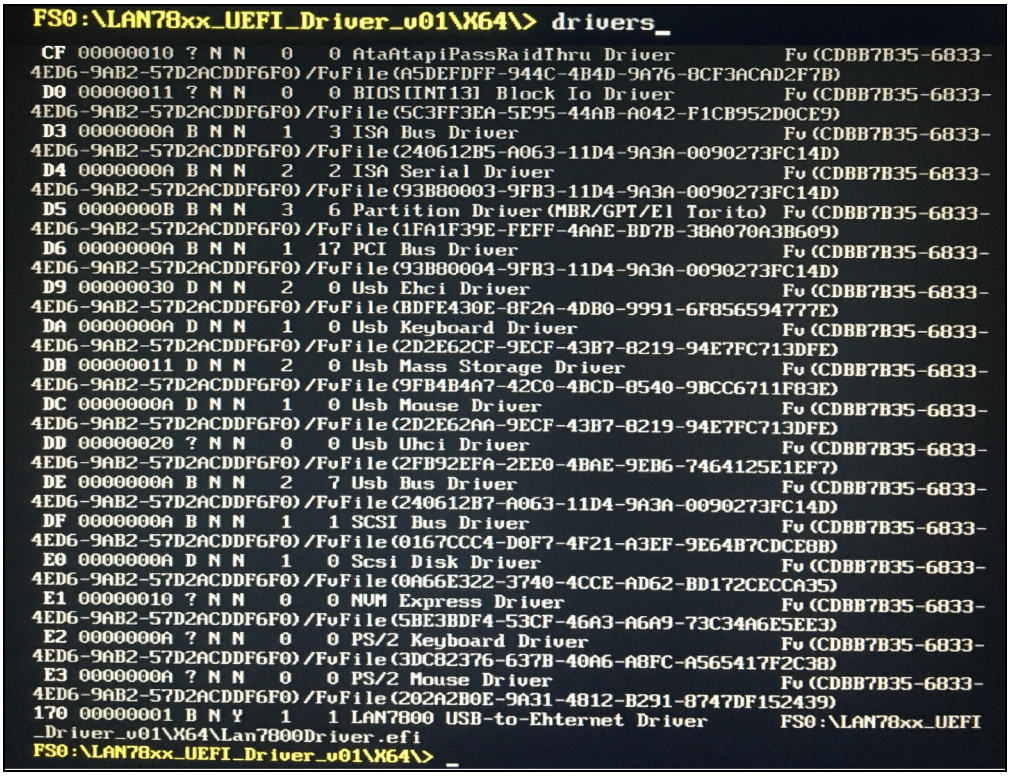
1. Unload LAN78XX UEFI driver using the command displayed in Figure 8-5.

FIGURE 8-5: UNINSTALL LAN78XX UEFI DRIVER



2. Check LAN78XX UEFI driver unloading as displayed in Figure 8-6

FIGURE 8-6: UNINSTALL LAN78XX UEFI DRIVERS



8.3 UEFI EEPROM TOOL

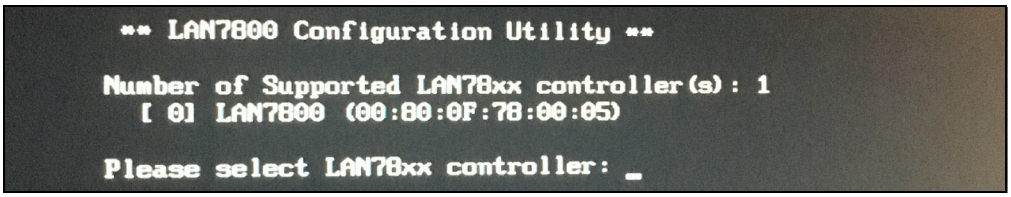
To read data from the EEPROM using UEFI:

1. After loading the LAN78XX UEFI driver, execute EEPROM utility driver as displayed in Figure 8-7.

FIGURE 8-7: LOAD UEFI EEPROM TOOL

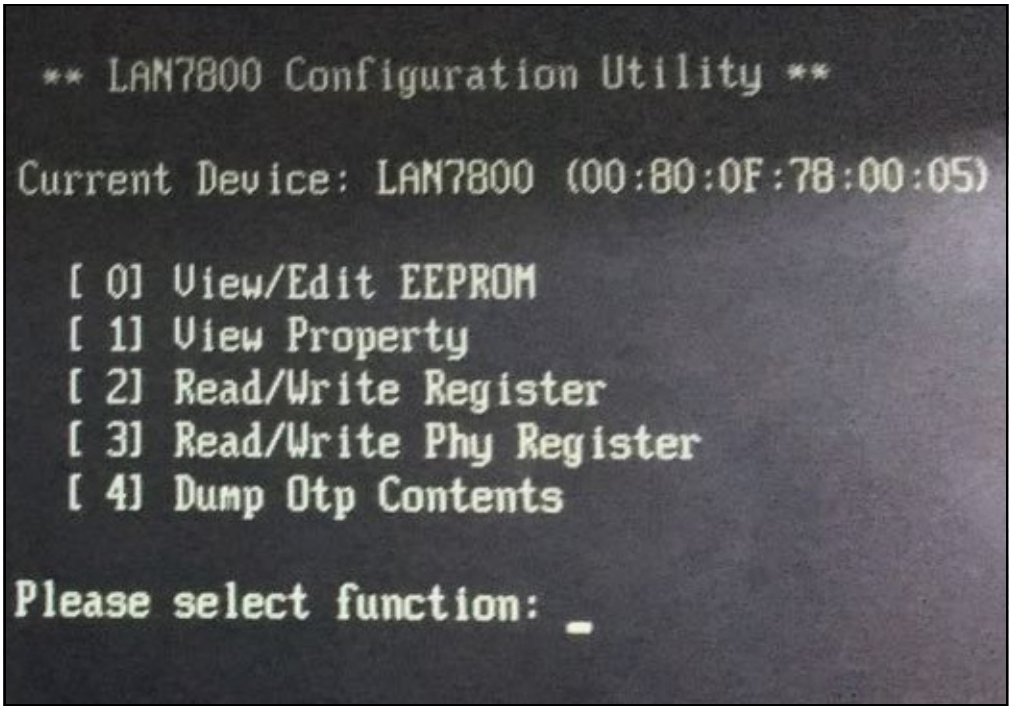
2. Select the LAN78XX controller, as displayed in Figure 8-8

FIGURE 8-8: LAN78XX CONTROLLER SELECTION



3. Select the various supported function in EEPROM tool, as displayed in Figure 8-9.

FIGURE 8-9: LAN78XX CONFIGURATION UTILITY SUPPORT FUNCTION



4. Display EEPROM content using the select function 0 as displayed in Figure 8-10, Figure 8-11, and Figure 8-12.

FIGURE 8-10: VIEW/EDIT EEPROM

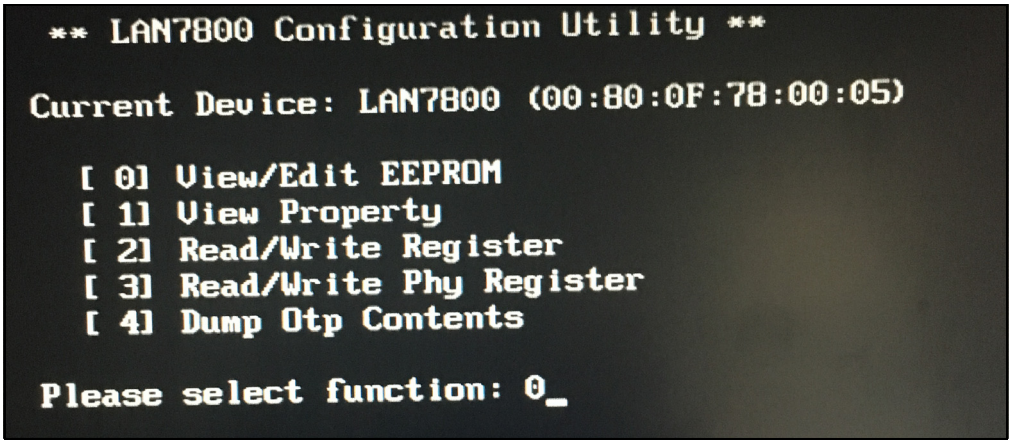


FIGURE 8-11: READING EEPROM

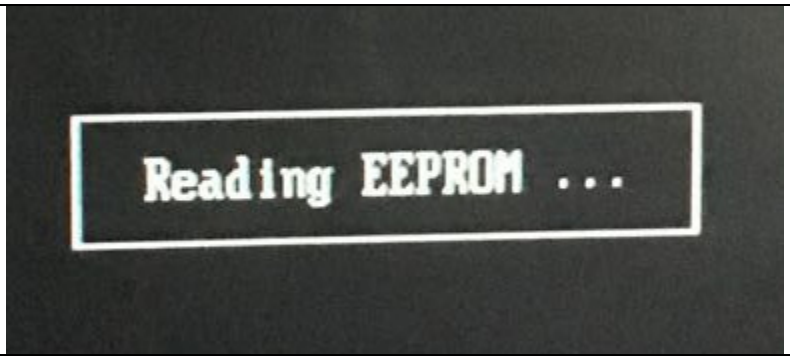
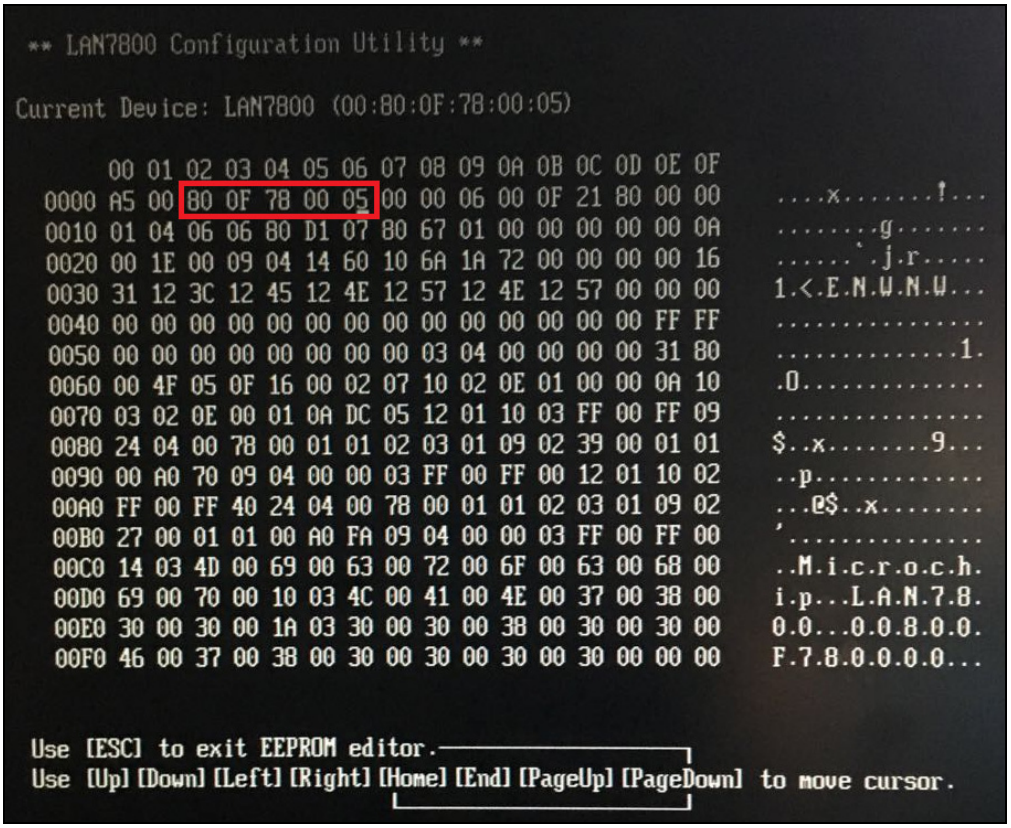


FIGURE 8-12: EEPROM CONTENT



5. Edit the EEPROM content using arrow keys.
6. Save and exit the EEPROM editor using <Esc> as displayed in Figure 8-13 Figure 8-14 Figure 8-15 Figure 8-16.

FIGURE 8-13: EEPROM CONTENT AFTER EDIT

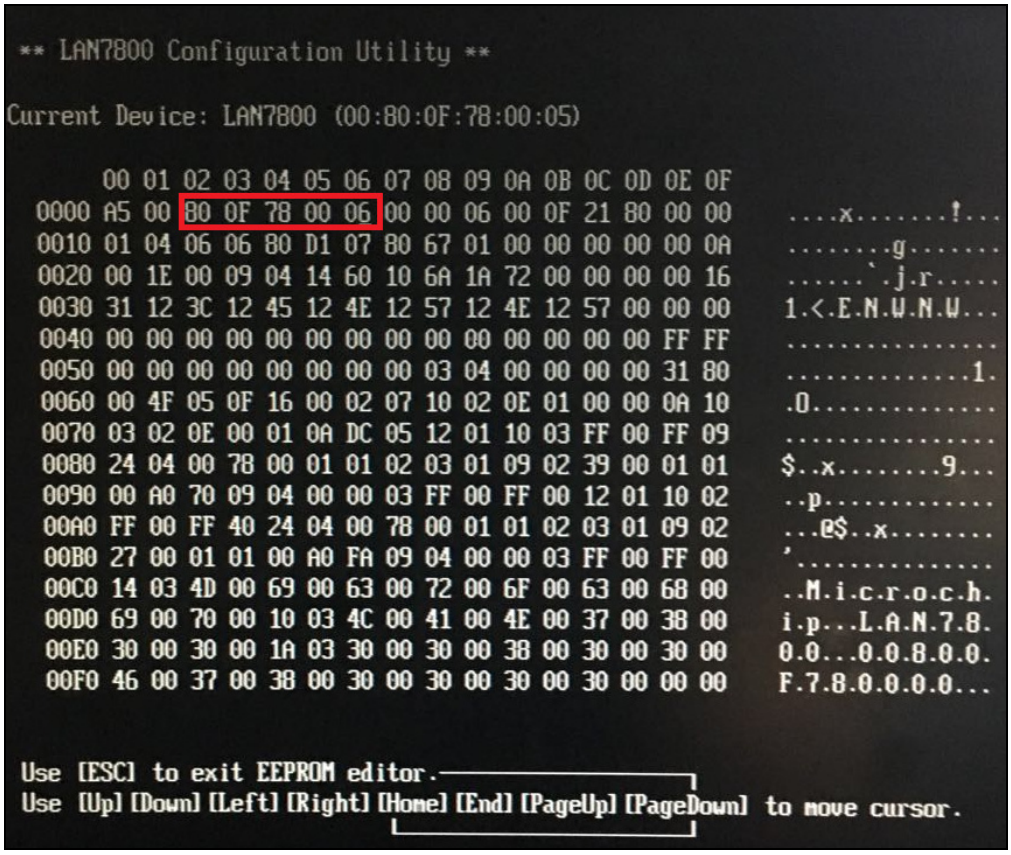


FIGURE 8-14: CONFIRMATION WINDOW FOR SAVE THE EEPROM CONTENT

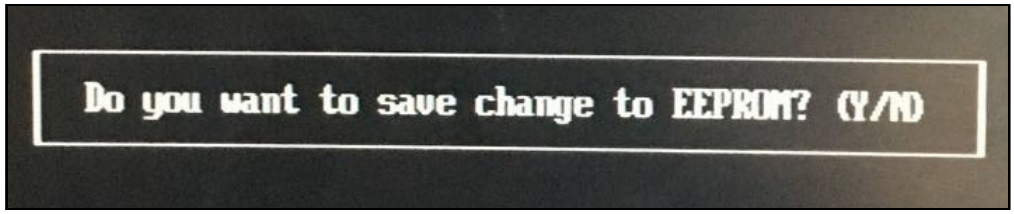


FIGURE 8-15: CONFIRMATION WINDOW FOR WRITE EEPROM

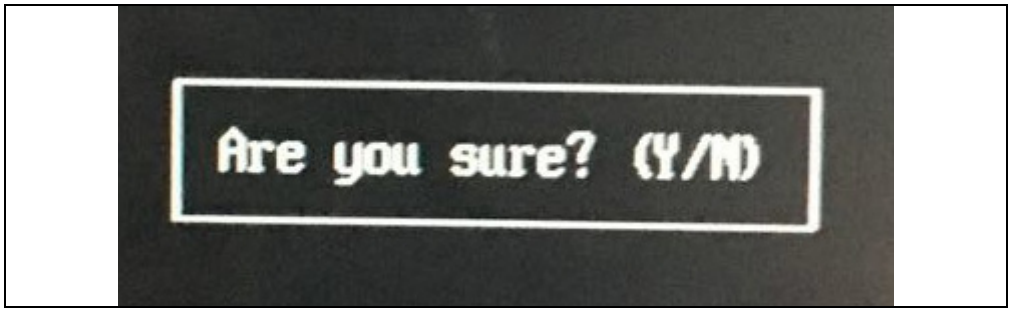
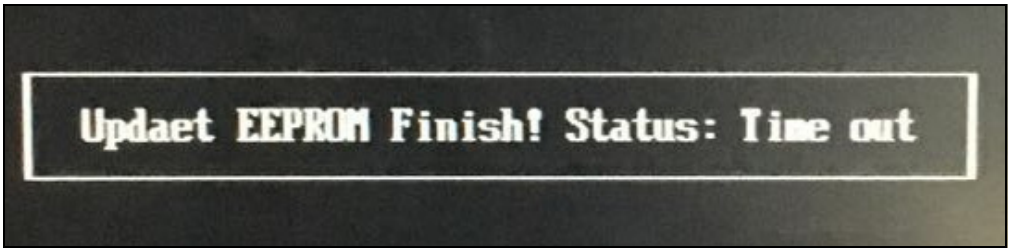
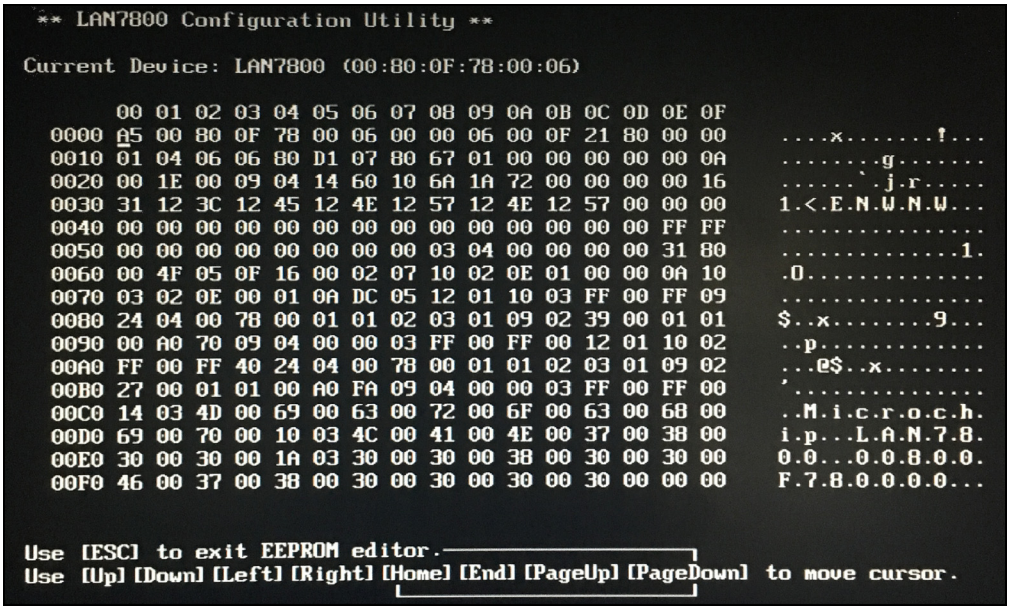


FIGURE 8-16: SUCCESS FULLY MODIFIED EEPROM CONTENT



7. After editing the EEPROM content, read and make sure EEPROM content is edited properly, as displayed in Figure 8-17.

FIGURE 8-17: EEPROM CONTENT AFTER EDITING SUCCESSFULLY



8. Display view property content using select function `1`, as displayed in the Figure 8-18 and Figure 8-19.

FIGURE 8-18: FUNCTION SELECTION WINDOW

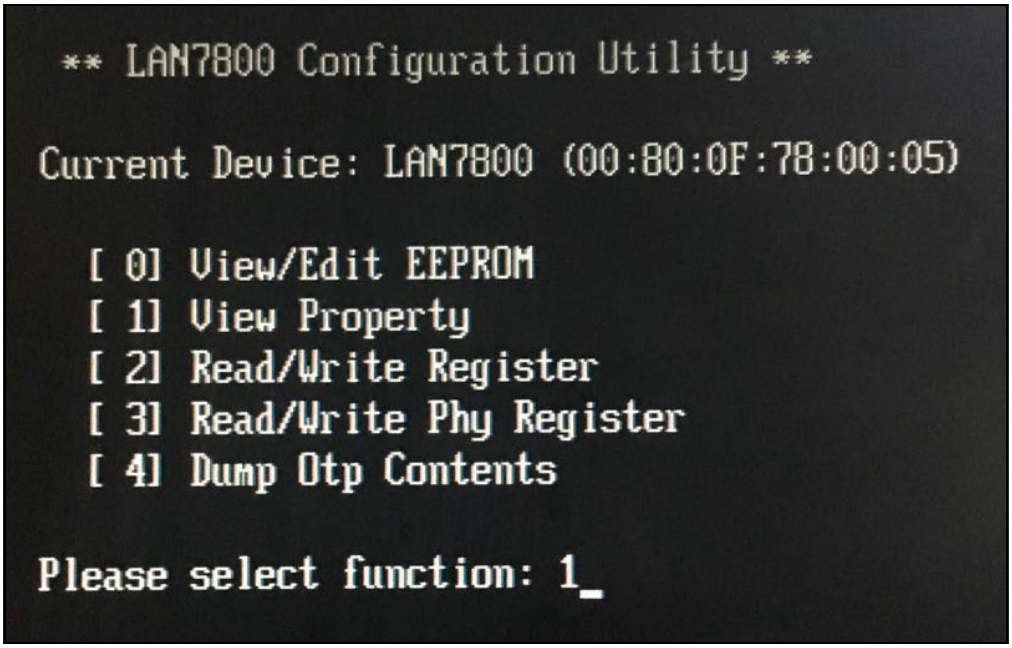
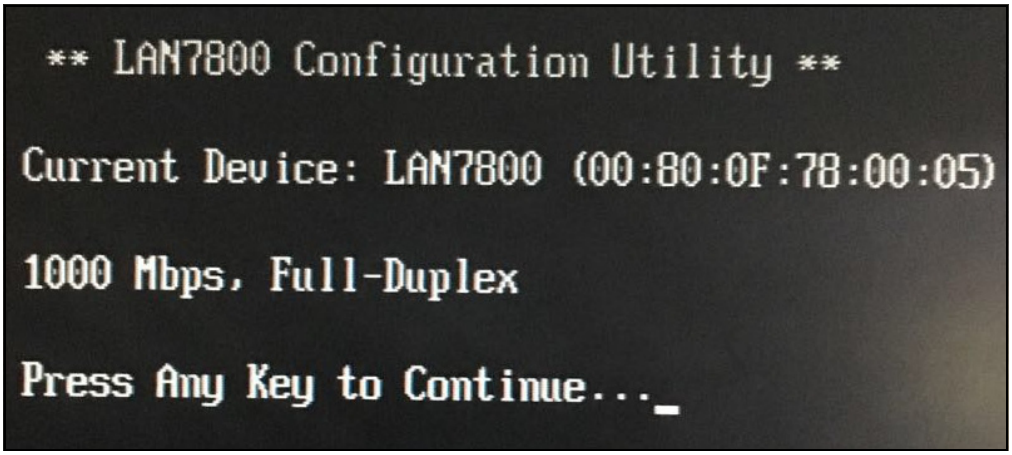


FIGURE 8-19: DEVICE PROPERTY WINDOW



9. Display Read or Write the register content using select function 2, as displayed in the Figure 8-20.
10. Display the value of given register address and also modify the register with new value as displayed in Figure 8-21.

FIGURE 8-20: FUNCTION SELECTION WINDOW

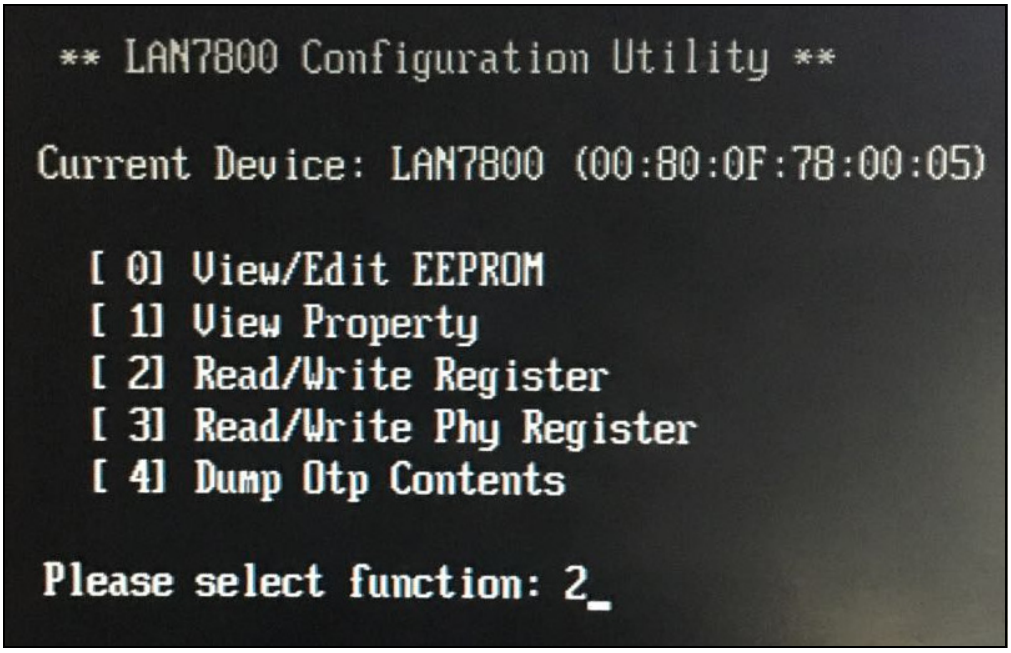
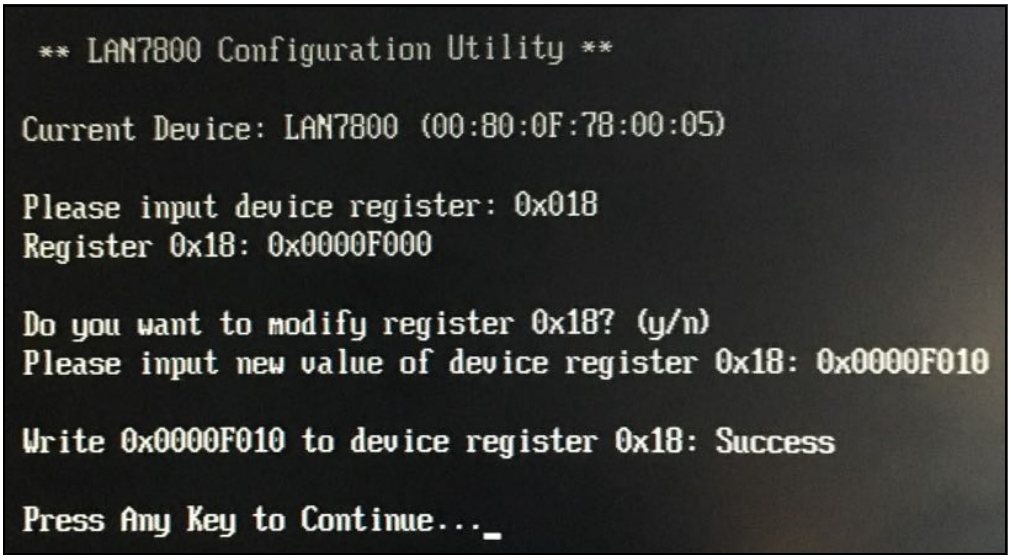
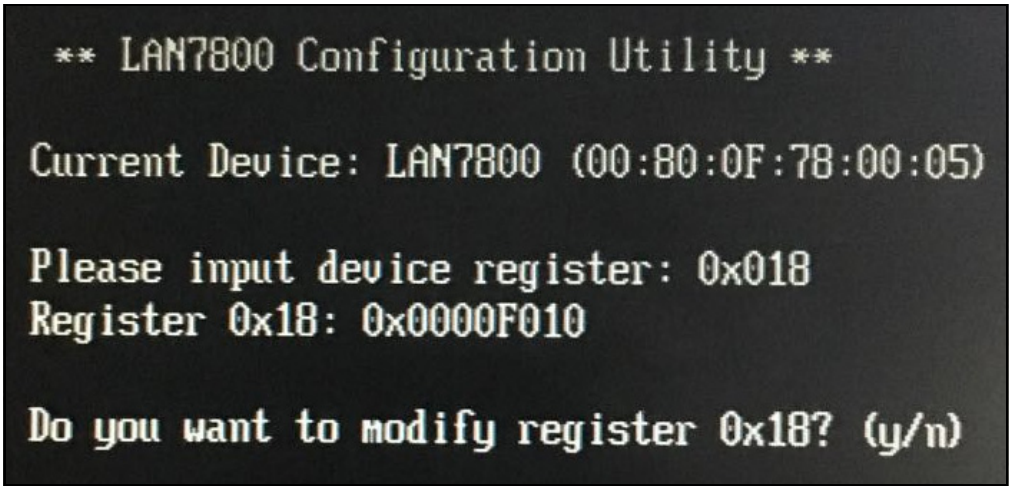


FIGURE 8-21: WRITE REGISTER WINDOW



11. Read and make sure the register value is modified properly, as displayed in Figure 8-22.

FIGURE 8-22: READ REGISTER WINDOW



12. Display Read or Write the Phy Register content using select function 3, as displayed in Figure 8-23.
13. Display the value of given Phy register address and modify the register with new value as in Figure 8-24

FIGURE 8-23: FUNCTION SELECTION WINDOW

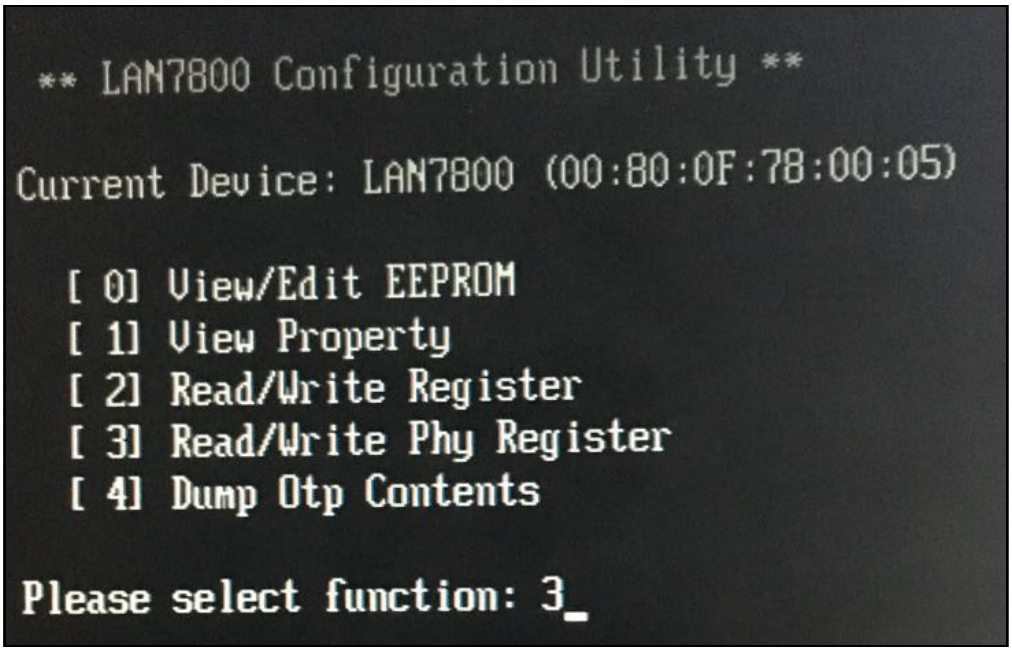
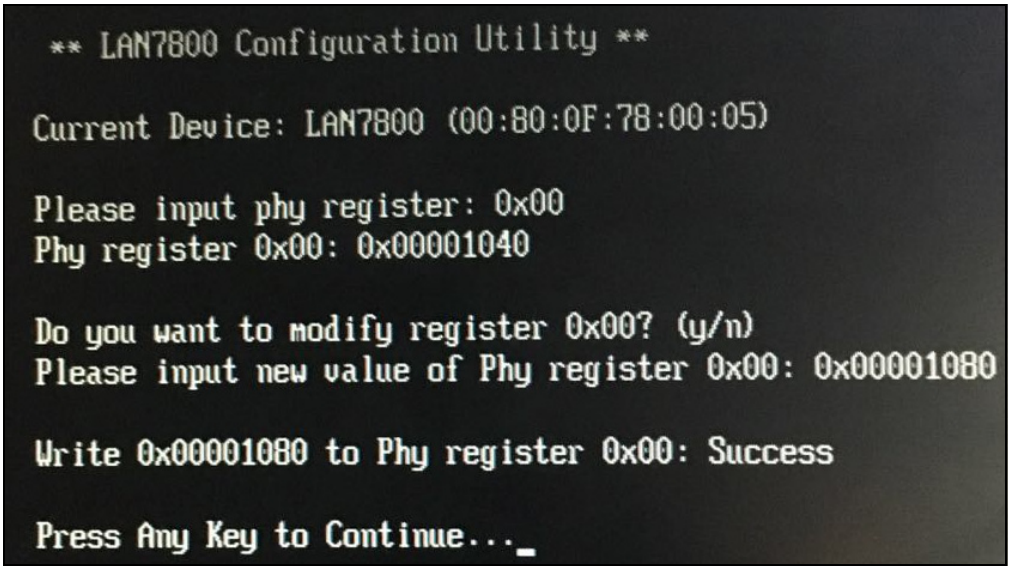
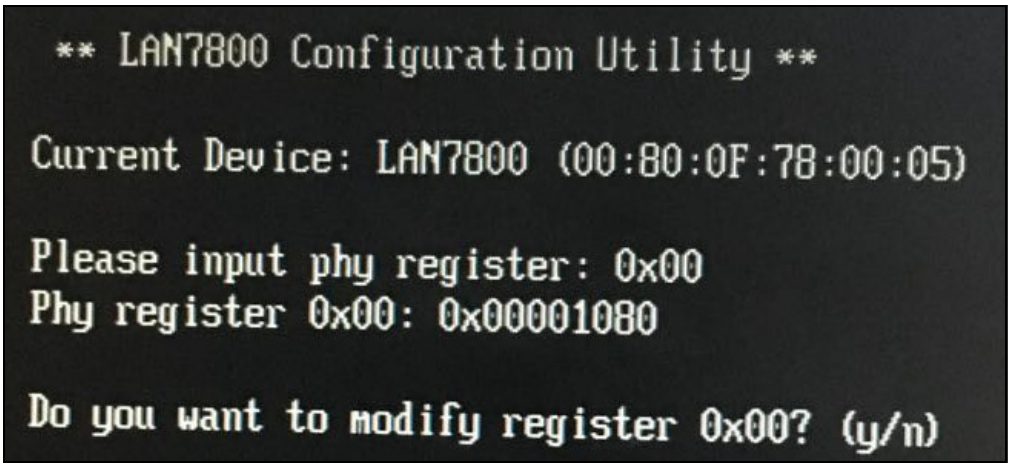


FIGURE 8-24: PHY REGISTER WINDOW



14. Read and make sure the register value is modified properly, as displayed in Figure 8-25.

FIGURE 8-25: READ PHY REGISTER



15. Display Dump OTP content using select function 4, as displayed in Figure 8-26 and Figure 8-27.

FIGURE 8-26: FUNCTION SELECTION WINDOW

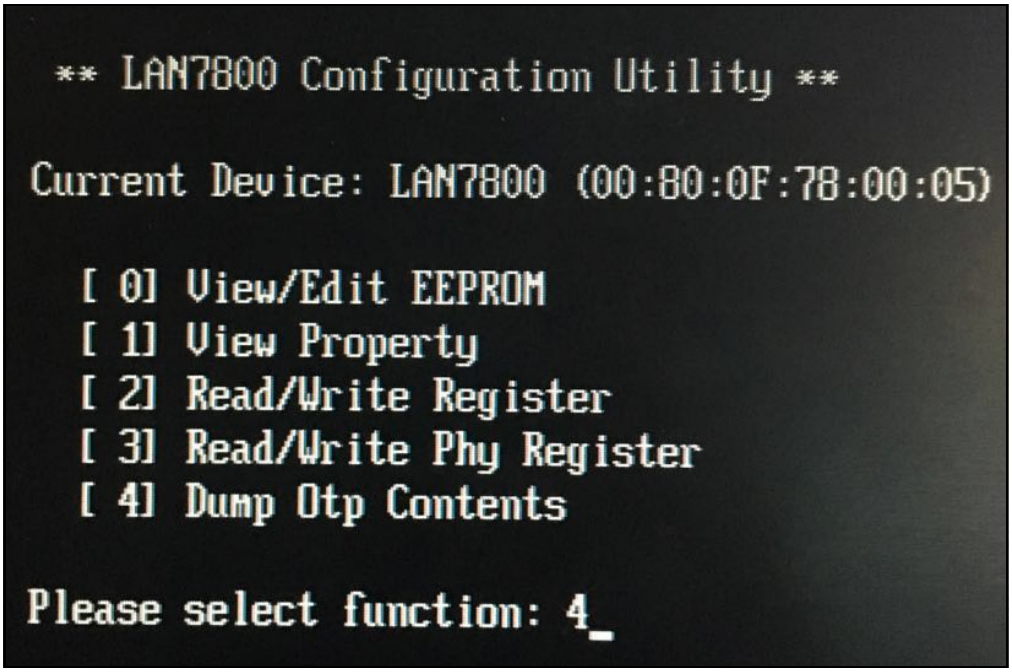
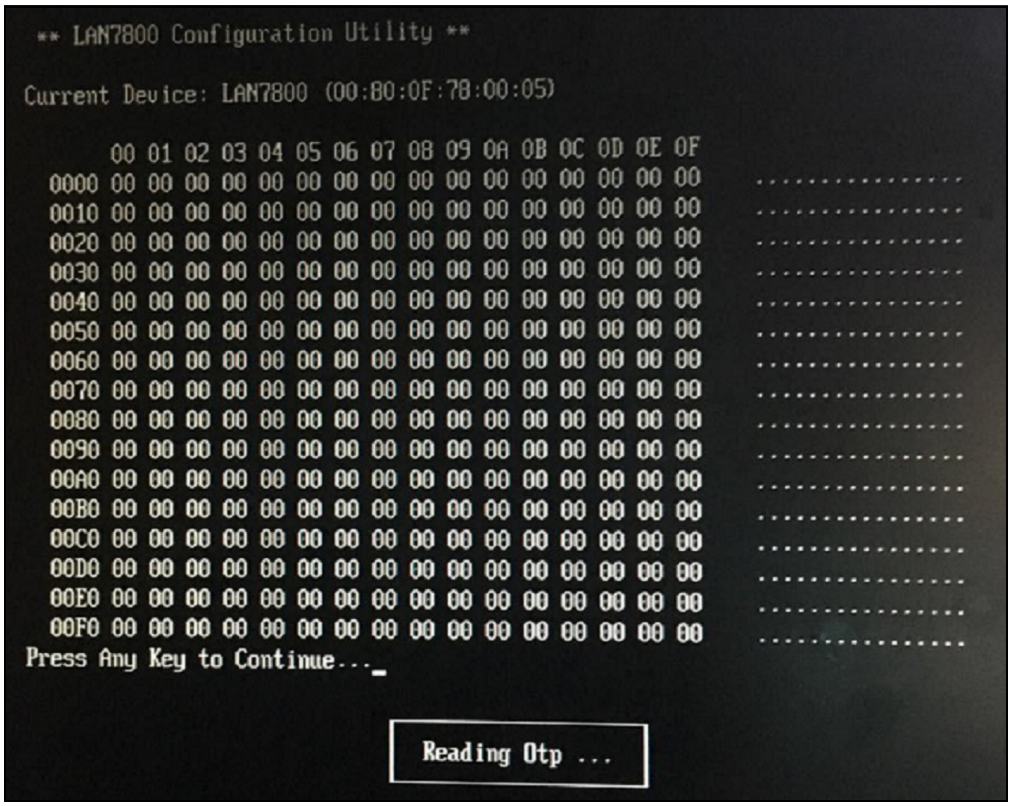


FIGURE 8-27: DUMP OTP



Chapter 9. Advanced Driver Parameters

The device drivers provide a set of advanced parameters that allow the device to be tailored to a specific application. The methods and parameters vary, depending on the operating system. This chapter provides an overview of how to access these parameters under various operating systems.

9.1 WINDOWS PARAMETERS

Windows parameters are accessible through the **Advanced** tab of the Device Properties window. Examples of these parameters in Windows Vista, 7, 8 and 10 are displayed in Figure 9-1, Figure 9-2, Figure 9-3, and Figure 9-4 respectively. These parameters may change between driver releases.

For additional details, refer to the release notes (`readme.txt`) of the particular release being used.

FIGURE 9-1: WINDOWS VISTA ADVANCED PARAMETERS

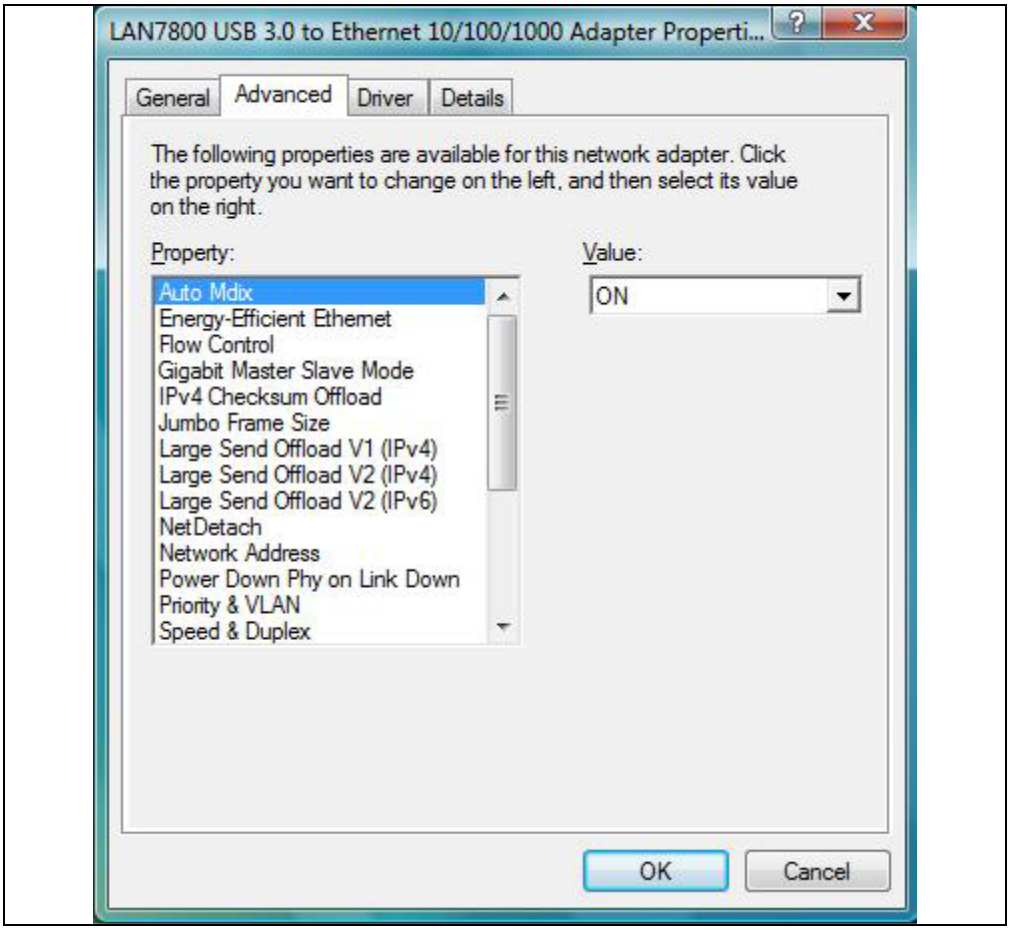


FIGURE 9-2: WINDOWS 7 ADVANCED PARAMETERS

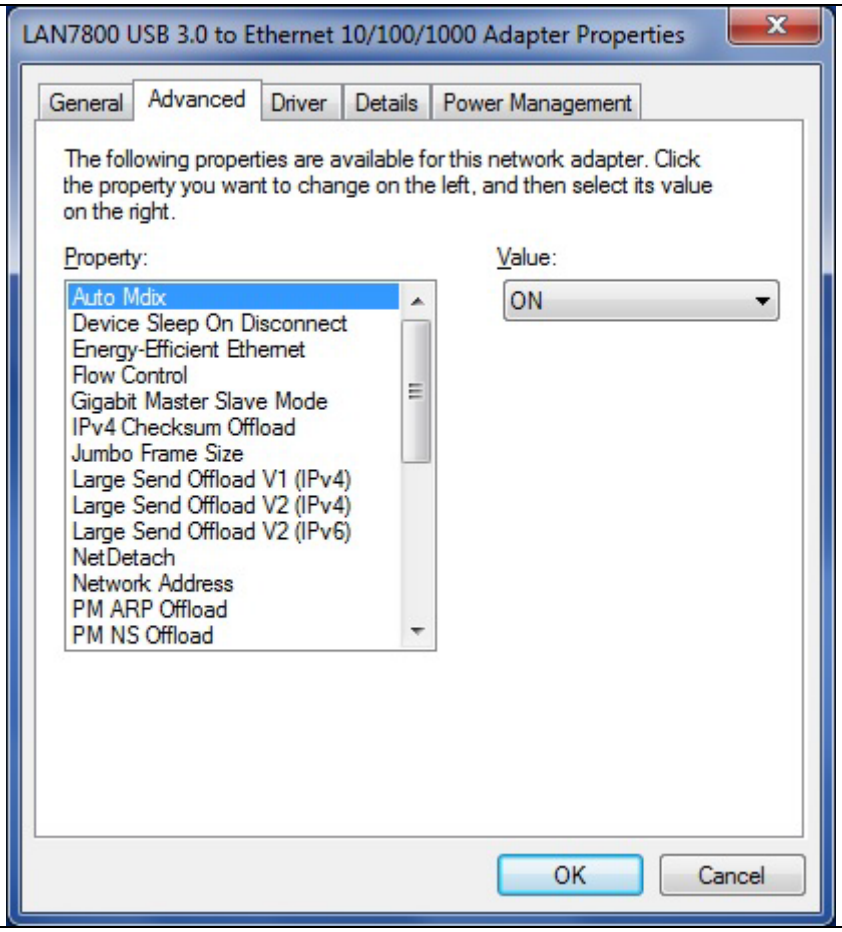


FIGURE 9-3: WINDOWS 8 ADVANCED PARAMETERS

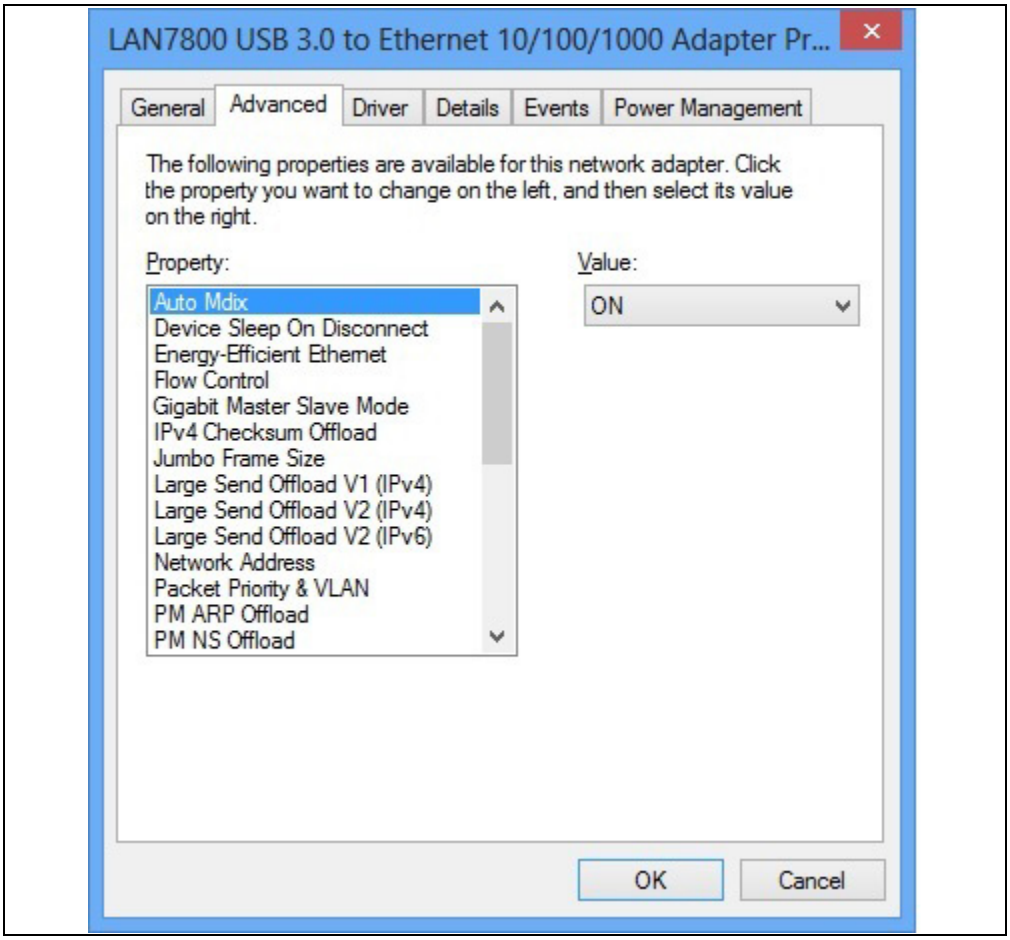
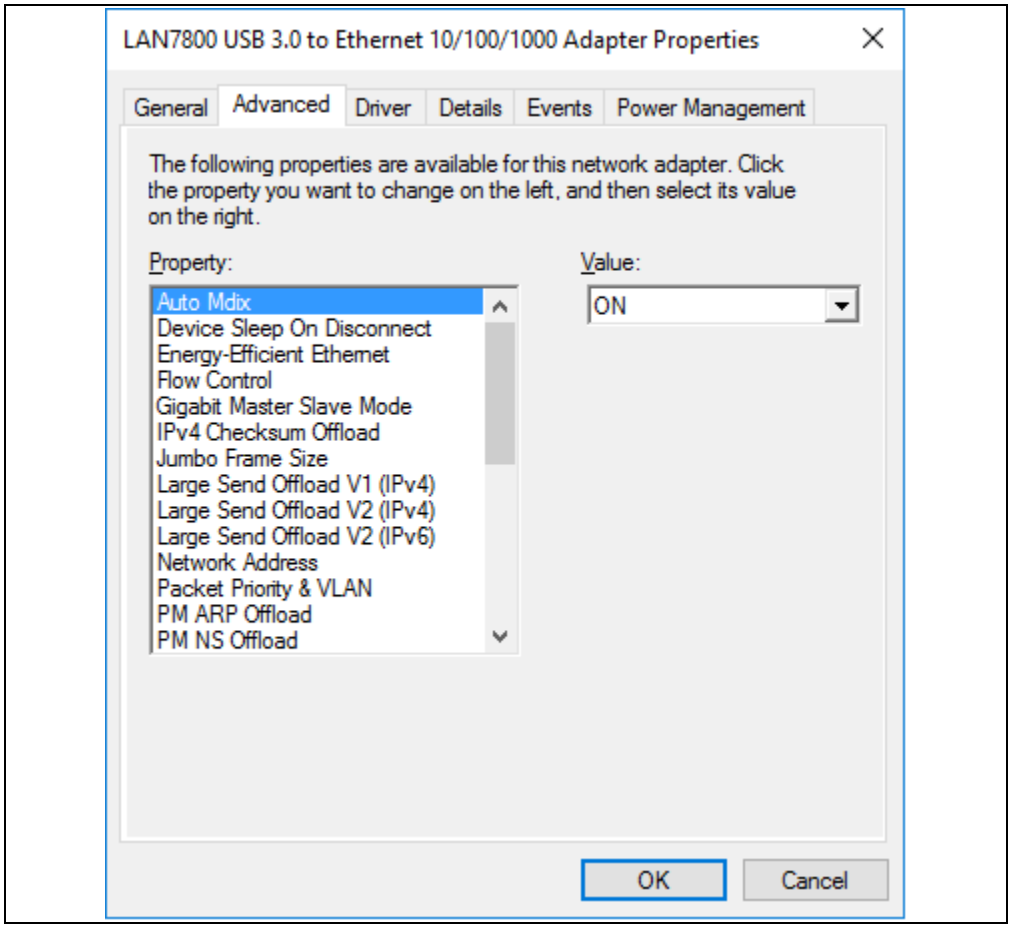


FIGURE 9-4: WINDOWS 10 ADVANCED PARAMETERS



9.2 LINUX PARAMETERS

Linux parameters may be modified via Runtime Standard (`ethtool`).

9.2.1 Ethtool

The device driver supports the standard Linux `ethtool` API. Refer to the Linux documentation for additional information.

Note:	The supported device driver parameters may change between driver releases. For additional details, please refer to the release notes (`readme.txt`) of the particular release being used.

Chapter 10. Legacy Pre-Execution Environment (PXE) Support

Currently, Microchip distributes a pre-execution environment (PXE) for LAN78xx driver through the Microchip website. However, as PXE gets updated often, it is recommended to check ipxe.org for the LAN78xx driver before downloading it from www.microchip.com.

NOTES:

Chapter 11. LAN78XX Command Line Utility

The 7800WinCmd a command line utility runs in a Windows Command prompt provide support for programming the EEPROM/OTP and testing basic LAN7800/LAN785 functionality. This chapter details the operation of the LAN78XX command line Utility.

Note:	Functionality and appearance may differ between releases. For the latest release information, please refer to the release notes located at the root of the LAN78XX command line Utility.
Note:	Protouch2 GUI is a Configuration and Programming tool for LAN products LAN78XX, USB253X/USB4604, USB57X4, USB5742 and other families of Microchip USB hub products. Latest Protouch2 tool can be downloaded from Microchip.com.

11.1 INSTALLATION

This section details the installation of the LAN78XX command line utility, the utility device driver, and includes setup and system requirements.

11.1.1 Setup and System Requirements

Table 11-1 details the LAN78XX command line utility setup and system requirements.

TABLE 11-1: SETUP AND SYSTEM REQUIREMENTS

Category	Requirements
PCs	• A 2 GHz Pentium 4 or higher PC is recommended for both the EEPROM programmer host system and the test partner system. • The test partner should support an IP/ICMP stack capable of replying to 65000 byte ICMP echo requests ("pings of 65000"). • Firewalls and any other existing security software must be set to enable "pings of 65000" in both EEPROM programmer host and test partner systems (if unsure on how to do the latter, simply turn firewalls off).
Supported OSs	• Windows XP x86 (32-bit) MUST be the OS running in the EEPROM programmer host system.
Environment	• The utility is designed to be used standalone with NO additional applications running on either the EEPROM programmer host or test partner.
Network	• End-to-end link between the device and the test partner should be on a closed network with no other connected hosts. • The test partner system (or switch) must be capable of (and configured to) autonegotiate advertising of all valid 10/100/1000 half/full duplex combinations. • The device's interface must be set to a static IP address that is on the same subnet as the test partner.

TABLE 11-1: SETUP AND SYSTEM REQUIREMENTS (CONTINUED)

Category	Requirements
Data	• Manual editing of the 78xxeep.ini file is not supported, and may lead to utility malfunction, corrupted EEPROM, or both.

11.1.2 Starting LAN78XX Utility and Contents

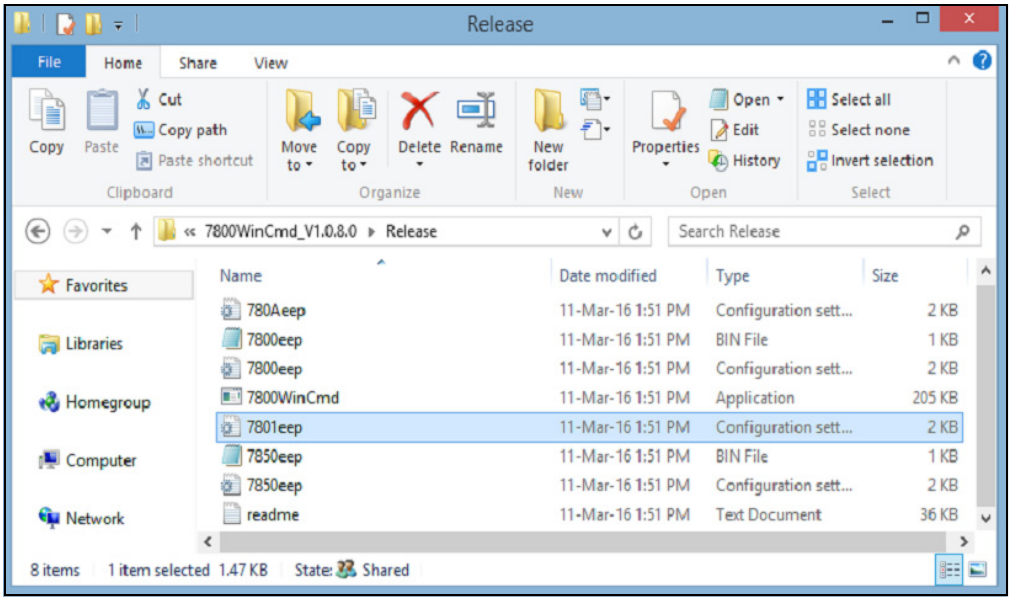
Open a Windows Command prompt with administrator privilege and run this utility with the Parameters.

The following files will be located in the chosen directory.

TABLE 11-2: FILES MANIFEST

File	Description
README.txt	A readme file detailing release notes and important information
LAN7800WinCmd.exe	The utility executable
7800eep.ini	Sample text initialization file for LAN7800 bus powered remote wake capable device
7800eep.bin	Sample binary file for LAN7800 bus powered remote wake capable device (matches .ini)
7850eep.ini	Sample text initialization file for LAN7800 bus powered remote wake capable device
7850eep.bin	Sample binary file for LAN7850 bus powered remote wake capable device (matches .ini)
7801eep.ini	Sample text initialization file for LAN7801 device (MCHP internal use only)
780Aeep.ini	Sample text initialization file for LAN780A device (MCHP internal use only)

FIGURE 11-1: LAN78XX COMMAND LINE UTILITY CONTENTS



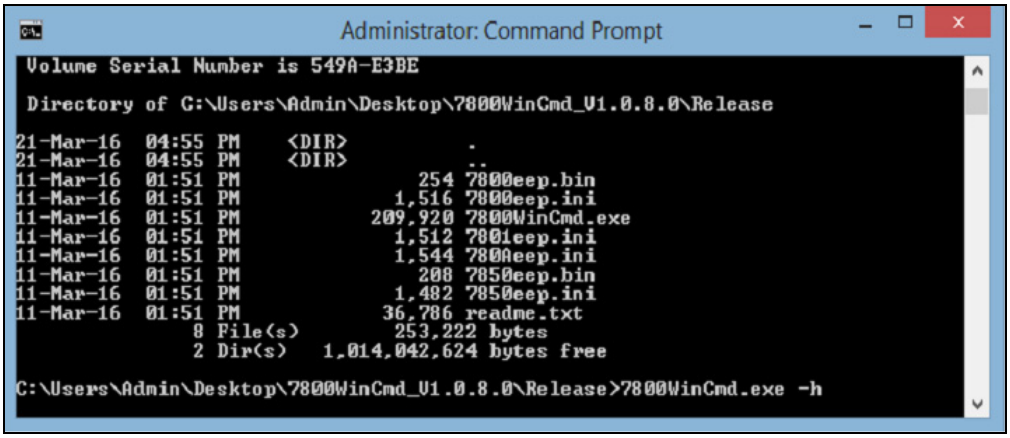
11.2 OPERATION

This section details the operation of the Windows Manufacturing Utility, including the following:

11.2.1 Starting the Utility

Open a Windows Command prompt with administrator privilege and run this utility with the Parameters.

FIGURE 11-2: LAUNCHING THE LAN78XX COMMAND LINE UTILITY



11.2.2 Using the Utility

11.2.2.1 UTILITY COMMAND LINE PARAMETERS DETAILS

```
7800WinCmd [-X <AdapterIndex>]
[(-b <bin_filename> -i <ini_filename>)(-w -r)]
[-M <mac_address> -S <serial_number> -m <mac_increment> -s
<serial_increment> -N ]
[-O] [-C <YES> ]
[-i <ini_filename> -z <target_bin_file> -d <device name>]
[-l <link mode>] [-p <Destination ip address>] [-L] [-h]
[-R <Register addr in hex>] [-W <Register addr in hex> -v <hex value>]
[-y <phy index in hex>]
[-Y <phy index in hex> -v <hex value>]
[-G (v b l k p o)]
```

FIGURE 11-3: EEPROM READ

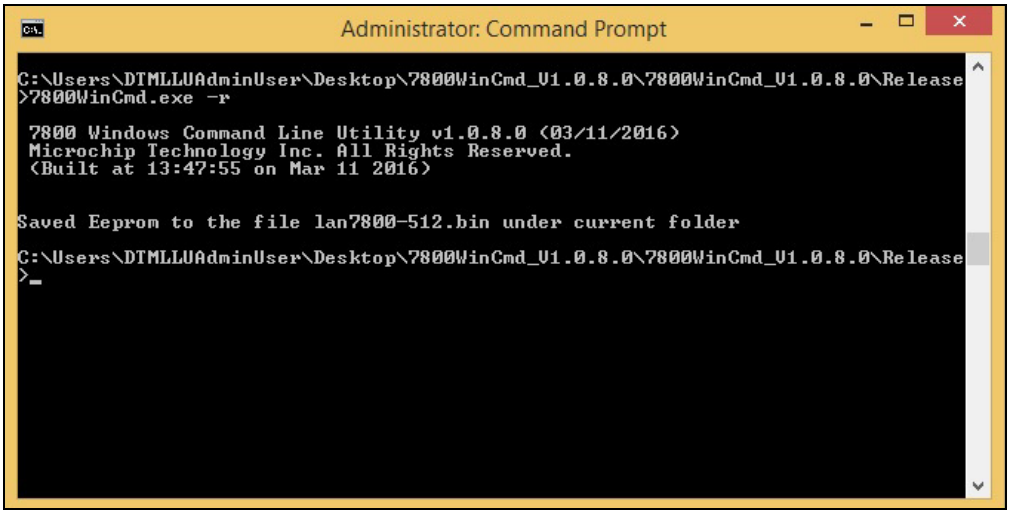
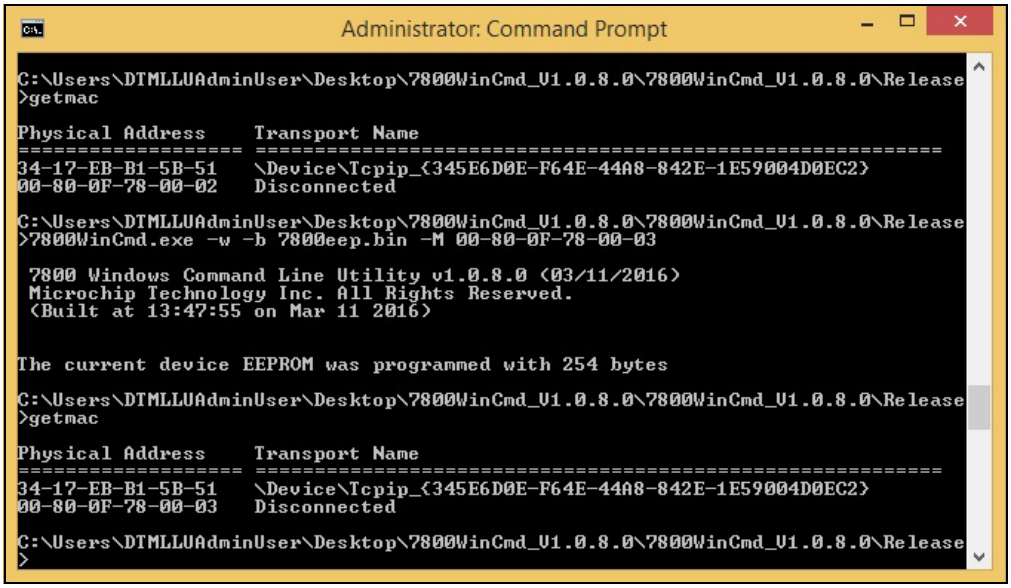


FIGURE 11-4: EEPROM WRITE MAC ADDRESS



The following fields are available:

TABLE 11-3: EEPROM COMMAND LINE PARAMETERS

Category	Descriptions
-X <AdapterIndex>	Support for multiple adapters. If more than one LAN7800 is plugged in, all commands can be directed to a specific LAN7800. The index is a zero based number associated with the adapter.
-b <bin_filename>	Uses the binary file <bin_filename> as "base" for the EEPROM contents. This option is mutually exclusive with the "-i" option, but either -b or -i must be present when operate write (-w) action. The Mac address and serial in the bin <bin_-filename> can be increased by using -N and -M, -S options.
-i <ini_filename>	Uses the options file <ini_filename> as the "base" for the EEPROM Contents. This option is mutually exclusive with the "-b" option, but either -b or -i must be present when operate write (-w) action. The Mac address and serial in the ini <ini_-filename> can be increased by using -N and -M, -S options.
-W	Write the EEPROM Contents specified in the file + overwrites to the device and verify them. The input file specified with the -b or -I option will be updated after the verification is complete according to the -N option.
-M <mac_address>	Overrides the mac address to the device EEPROM with the mac_address provided. mac_address format is ab:cd:ef:gh:ij:kl with a through l hex digits (For example, 00:80:0F:78:04:00). if -N is not specified, the binary/ini file will not be updated with the Mac Address, only the EEPROM content that written to the device get updated.
-S <serial_num>	Overrides the serial number to the device EEPROM with the serial_num provided. Serial_num format is a 16 characters. if -N is not specified, the binary/ini file will not be updated with the serial number, only the EEPROM content written to the device gets updated. If the supplied serial number is not con-tain all hex digits (For example: other than 000780404 format) then serial number cannot be incremented with -N option.

TABLE 11-3: EEPROM COMMAND LINE PARAMETERS (CONTINUED)

Category	Descriptions
-m <mac_increment>	This option increases the mac address from the ini/binary file for the next write. The maximum increment step is 255. Any value less than or equal to 0 will be overridden with 1. It must be used with -N option.
-s<serial_increment>	This option increases the Serial number from the ini/binary file for the next write. The maximum increment step is 255. Any value less than or equal to 0 will be overridden with 1. The value 0 means do not update serial number. If the value greater than 0 then -N option must be specified.
-N	Update ini or bin file with next mac address and serial number. Next mac address and serial number will be computed with -m <mac_increment> and -s <serial_increment>. If there is no -m <mac_increment> or -s <serial_increment> specified, the increment will be 1 if -N is specified. “-N” option is required when the “-m/-s” option is present.
-r	Read the current EEPROM Contents in the device and write to the 7800eep_256/7800eep_512.bin binary file (depending on the device) under the current folder.
-O	OTP Operation. -O option must accompany with -w (write) or -r (read) option. The -w option also include -b or -i option.
-C <YES>	Suppress the user confirmation before writing OTP. Must be “YES” (all upper case letters).
-z <target file>	Save the EEPROM/OTP file. Works with the -i <ini_filename> option. The ini file will be converted to binary file and saved in to target file. When -O option is specified binary file is for OTP. Without the -O option ini file is converted to EEPROM binary and saved to target file. To generate OTP image with signature 0xF7 (OTP occupies upper 256 bytes) use the [OtpConfig] and OtpSignature=0xF7 in the ini file.
-d <device name>	Specify the device name for which EEPROM image need to be generated. Works with the -z <target file> option. It should be either LAN7800 or LAN7850.
-p <Dest ip address>	Ping option. IP address must be specified when the “-p” option is present.
-l <link_mode>	The utility will attempt to auto negotiate using one of the following restricted negotiation capabilities “10hd”, “10fd”, “100hd”, “100fd” or “1000fd”.
-L	External loop back test. The -l option with “100fd” or “1000fd” need to be specified. If -l option is not specified assumes 1000fd. No need to specify IP address. Will ignore it.
-R <Register address>	Read Control and Status register. The Register address as hex value.
-W <Register address>	Write Control and Status register. Register address as hex value. The -v option should be used along with this option.
-y <PHY Register index>	Read PHY register. The PHY register index in hex.
-Y <PHY Register index >	Write PHY register. the PHY register index in hex. The -v option should be along with this option.
-G v	Get driver version

TABLE 11-3: EEPROM COMMAND LINE PARAMETERS (CONTINUED)

Category	Descriptions
-G b	Get adapter bits. Each bit represents an adapter. If 2 adapters are plugged in, you might see 0x03. That can be used to set the index with -X.
-G l	Get adapter location. For multiple adapters, this command helps identify the location of the adapter. An example is: "Port_#0006.Hub_#0007" Does not work on Windows XP. See -G p.
-G k	Get adapter registry key. For multiple adapters, this command returns the registry key for the adapter. An example is: "{4d36e972-e325-11ce-bfc1-08002be10318}\0015" If that is appended to: "HKLM\SYSTEM\CurrentControlSet\Control\Class\" You would have a full path to the registry key.
-G p	Get adapter port. For multiple adapters, this command returns the port number. On Windows XP -G l does not return the port number. This is an alternative.
-G o	Get adapter PDO name. For multiple adapters, this command returns the PDO name for the adapter. An example is: "\Device\USBPDO-9"
-h	Displays help menu

11.2.3 Exiting the utility

Close the command line window for exiting the utility.

11.3 INI FILE CONFIGURATIONS:

MAC Address [MacAddress]:

This is the 6-byte universally-unique Mac address the board will use. Bytes are separated by a colon.

CurrentMacAddress	Defines the first MAC address (it will be incremented after Manual/Auto burn for the next device)
MaximumMacAddress	Defines the last MAC address to be used before rolling over to 0.
MacAddressIncrement	Will be added to Current MAC Address after successful EEPROM burn cycle (max is 255).

Serial Number [SerialNumber]:

This is the (unique for this model of device) USB serial number. It is a 9-digit hexadecimal number.

CurrentSerialNumber	Defines the first Serial Number (it will be incremented after Manual/Auto burn for the next device)
MaximumSerialNumber	Defines the last serial number to be used before rolling over.
SerialNumberIncrement	Will be added to Current serial number after successful EEPROM burn cycle (max is 255).

Strings [String]:

The user can save 4 customized strings (Manufacturer, Product, Configuration and Interface) with different lengths in the EEPROM. Each string can be individually disabled or enabled.

ID [ID]:

VendorID	This is your company's USB Vendor Id. It is a 4-digit hexadecimal number
ProductID	This is your device's USB Product Id. It is a 4-digit hexadecimal number
BcdDevice	This is the release number you assign to the device. It is a 4-digit binary-coded decimal (BCD) number. For example revision 1.23 would be represented as bcdDevice=0123

Power [Power]:

These parameters affect values of the USB configuration descriptor. Improper setups which violate the USB 2.0/3.0 specification will not be allowed. An error message will be displayed if an illegal value has been entered.

SSMaxPower	Maximum Power Consumption in mA for USB 3.0 operation (Between 8mA and 896mA for BusPower; between 8mA and 100mA for SelfPower)
HSMaxPower	Maximum Power Consumption in mA for USB 2.0 operation (between 2mA and 500mA for BusPower; between 2mA and 100mA for SelfPower)

Configuration flags0 [ConfigFlags0]:

SWModeSuspendNSelect	This specifies the mode of operation during which SW_MODE pin will be asserted. (Values in decimal) 0 - SW_MODE will be asserts in SUSPEND2 1 - SW_MODE will be asserts in SUSPEND2, SUSPEND1 and NetDetach 2 - SW_MODE will be asserts in SUSPEND2, SUSPEND1, SUSPEND0 and NetDetach 3 - SW_MODE will be asserts in SUSPEND3, SUSPEND2, SUSPEND1, SUSPEND0 and NetDetach
SWModePolarity	Selects the polarity of the SW_MODE pin 0 - Active low 1 - Active high
RemoteWakeupEnable	Enables/disables the remote wakeup feature 0 - Device does not support the remote wakeup 1 - Device support the remote wakeup

SelfPower	Selects the power method 0 - Bus power 1 - Self power
ConfigFlags0Override	This is override key for [ConfigFlags0]. All the fields for the Configuration Flags 0 in the EEPROM can be changed. The value should be in Hex and must be in the format "0xFFFFFFFF". When this value is not present the default value is:0x4011-BA04 (excluding the above specified bits in the ini file).

LED Configuration 0 [LEDConfig0]:

LED3Enable	When set LED3 will be enabled in the phy
LED2Enable	When set LED2 will be enabled in the phy
LED1Enable	When set LED1 will be enabled in the phy
LED0Enable	When set LED0 will be enabled in the phy

LED Configuration 1 [LEDConfig1]:

LED1Function	The LED mode for LED 1. The value in decimal and in the range 0 - 15 (see LED mode functionality in Table 11-4).
LED0Function	The LED mode for LED 0. The value in decimal and in the range 0 - 15 (see LED mode functionality in Table 11-4).

LED Configuration 2 [LEDConfig2]:

LED3Function	The LED mode for LED 3. The value in decimal and in the range 0 - 15 (see LED mode functionality in Table 11-4).
LED2Function	The LED mode for LED 2. The value in decimal and in the range 0 - 15 (see LED mode functionality in Table 11-4).

TABLE 11-4: MODE FUNCTIONALITY

LED	Functionality
0	Link/Activity 1 - No link in any speed on any media interface. 0 - Valid link at any speed on any media interface. Blink or pulse-stretch - Valid link at any speed on any media interface with activity present.
1	Link1000/Activity 1 - No link in 1000BASE-T. 0 - Valid 1000BASE-T Blink or pulse-stretch - Valid 1000BASE-T link with activity present.
2	Link100/Activity 1 - No link in 100BASE-TX. 0 - Valid 100BASE-TX. Blink or pulse-stretch - Valid 100BASE-TX link with activity present.

TABLE 11-4: MODE FUNCTIONALITY (CONTINUED)

LED	Functionality
3	Link10/Activity 1 - No link in 10BASE-T. 0 - Valid 10BASE-T link. Blink or pulse-stretch - Valid 10BASE-T link with activity present.
4	Link100/1000/Activity 1 - No link in 100BASE-TX, or 1000BASE-T. 0 - Valid 100BASE-TX, or 1000BASE-T link. Blink or pulse-stretch - Valid 100BASE-TX, or 1000BASE-T link with activity present.
5	Link10/1000/Activity 1 - No link in 10BASE-T, or 1000BASE-T. 0 - Valid 10BASE-T, or 1000BASET-T link. Blink or pulse-stretch - Valid 10BASE-T, or 1000BASE-T link with activity present.
6	Link10/100/Activity 1 - No link in 10BASE-T, or 100BASE-TX. 0 - Valid 10BASE-T or 100BASE-TX, link. Blink or pulse-stretch - Valid 10BASE-T, or 100BASE-TX link with activity present.
7	Reserved
8	Duplex/Collision 1 - Link established in half-duplex mode, or no link established. 0 - Link established in full-duplex mode. Blink or pulse-stretch - Link established in half-duplex mode but collisions are present.
9	Collision 1 - No collision detected. Blink or pulse-stretch - Collision detected.
10	Activity 1 - No activity present. Blink or pulse-stretch - Activity present (becomes TX activity present if register bit 30.14 is set to 1)
11	Reserved
12	Auto-Negotiation Fault 1 - No auto-negotiation fault present. 0 - Auto-negotiation fault occurred.
13	Serial Mode Serial stream. Only relevant on PHY port 0 and reserved in others.
14	Force LED Off 1 - De-asserts the LED.
15	Force LED On 0 - Asserts the LED.

LED Behavior Configuration [LEDBehaviorConfig]:

LEDBehavior	Sets the LED Link/pulse behavior The value must be in Hex in the format “0xFFFF” (see LED behavior bit definitions in Table 11-5).

TABLE 11-5: MODE FUNCTIONALITY

BIT	Functionality
15:13	Reserved
12	LED pulsing enable 0 - Normal operation. 1 - LEDs pulse with a 5-kHz, programmable duty cycle when active.
11:10	LED blink/pulse-stretch rate 00 - 2.5-Hz blink rate / 400 ms pulse-stretch. 01 - 5-Hz blink rate / 200 ms pulse-stretch. 10 - 10-Hz blink rate / 100 ms pulse-stretch. 11 - 20-Hz blink rate / 50 ms pulse-stretch.
9	Reserved
8	LED3 pulse-stretch/blink select 1 - Pulse-stretch. 0 - Blink.
7	LED2 pulse-stretch/blink select 1 - Pulse-stretch. 0 - Blink.
6	LED1 pulse-stretch/blink select 1 - Pulse-stretch. 0 - Blink.
5	LED0 pulse-stretch/blink select 1 - Pulse-stretch. 0 - Blink.
4	Reserved
3	LED3 combine feature disable 0 - Combine enabled (link/activity, duplex/collision). 1 - Disable combination (link only, duplex only).
2	LED2 combine feature disable 0 - Combine enabled (link/activity, duplex/collision). 1 - Disable combination (link only, duplex only).
1	LED1 combine feature disable 0 - Combine enabled (link/activity, duplex/collision). 1 - Disable combination (link only, duplex only).
0	LED0 combine feature disable 0 - Combine enabled (link/activity, duplex/collision). 1 - Disable combination (link only, duplex only).

GPIO Configuration 0 [GPIOConfig]:

GPIOEnable	Enables the particular GPIO. The value must be in Hex in the format "0xFFF". Each bit specifies a GPIO with LSB first (GPIO0). 0 - Disable GPIO 1 - Enable GPIO
GPIOBuffer	Selects the output buffer for GPIO The value must be in Hex in the format "0xFFF". Each bit specifies a GPIO with LSB first (GPIO0). 0 - Open-drain 1 - push/pull
GPIODirection	Selects direction of the GPIO. The value must be in Hex in the format "0xFFF". Each bit specifies a GPIO with LSB first (GPIO0). 0 - Output 1 - input

GPIOData	Specifies what value needs to output when GPIO configured as output The value must be in Hex in the format “0xFFF”. Each bit specifies a GPIO with LSB first (GPIO0). 0 - low 1 - high
GPIOWake	Configures the GPIO wakeup functionality The value must be in Hex in the format “0xFFF”. Each bit specifies a GPIO with LSB first (GPIO0). 0 - The GPIO can not wakeup the device. 1 - The GPIO can trigger a wakeup event.
GPIOWakePolarity	Selects the GPIO polarity The value must be in Hex in the format “0xFFF”. Each bit specifies a GPIO with LSB first (GPIO0). 0 - Wakeup/interrupt is triggered when GPIO is driven low 1 - Wakeup/interrupt is triggered when GPIO is driven high

GPIO PME Flags 0 [GPIOPME0]:

GPIOPMEEnable	Setting this bit enables the assertion of the GPIO5 pin, as a result of a Wakeup (GPIO) pin, Magic Packet, or PHY Link Up. The host processor may use the GPIO5 pin to asynchronously wake up, in a manner analogous to a PCI PME pin. 0 - The device does not support GPIO PME signaling. 1 - The device supports GPIO PME signaling. Note: When this bit is 0, the remaining GPIO PME parameters are ignored.
GPIOPMEConfiguration	This bit selects whether the GPIO PME is signalled on the GPIO pin as a level or a pulse. 0 - GPIO PME is signaled via a level. 1 - GPIO PME is signaled via a pulse
GPIOPMELength	When GPIOPMEConfiguration is set to 1 (pulse),this bit determines the duration of the pulse. 0 - GPIO PME pulse length is 1.5 mS. 1 - GPIO PME pulse length is 150 mS.
GPIOPMEPolarity	Specifies the level of the signal or the polarity of the pulse used for GPIO PME signaling. 0 - GPIO PME signaling polarity is low. 1 - GPIO PME signaling polarity is high.
GPIOBufferType	This bit selects the output buffer type for GPIO. 0 - Open drain driver / open source 1 - Push-Pull driver **Note:** Buffer Type = 0, Polarity = 0 implies Open Drain. Buffer Type = 0, Polarity = 1 implies Open Source.
GPIOPMEWOLSelect	Selects WOL wakeup events or Link up wakeup events. 0 - WOL event wakeup supported. 1 - PHY linkup wakeup supported. **Note:** If WOL is selected, the PME Magic Packet Enable and PME Perfect DA Enable bits determine the WOL event(s) that will cause a wakeup.
PMEMagicPacketEnable	When GPIOPMEWOLSelect set to 0 (WOL), this flag enables/disables Magic Packet Detection and wakeup. 0 - Magic Packet event wakeup disabled. 1 - Magic Packet event wakeup enabled.

PMEPerfectDAEnable	When GPIOPMEWOLSelect set to 0 (WOL), this flag enables/disables Perfect DA detection and wakeup. 0 - Perfect DA event wakeup disabled. 1 - Perfect DA event wakeup enabled.

GPIO PME Flags 1 [GPIOPME1]:

GPIOConnectEnable	Enables the assertion of the CONNECT pin, when the USB device controller is ready to attach to the host. Applicable only to HSIC part LAN7850.
GPIOConnectBufferType	Selects output buffer type for CONNECT pin. 0 - Open drain driver / open source 1 - Push-Pull driver
GPIOConnectPolarity	Selects the polarity of CONNECT signal 0 - GPIO CONNECT signaling polarity is low. 1 - GPIO CONNECT signaling polarity is high.
PMEWUFFEnable	Enables the wakeup frame for PME mode. 0 - Wakeup Frame detection disabled. 1 - Wakeup Frame detection enabled.

PME Wakeup filter Configuration [PMEWUFConfiguration]:

This key is used in the EEPROM only when GPIOPMEEnable and PMEWUFFEnable is enabled.

FilterAddressType	This specifies the destination address type of the pattern. 0 - Pattern applies only to unicast frames. 1 - Pattern applies to all frames. 2 - Pattern applies only to multicast frames.
FilterPatternOffset	Pattern starting offset. Specifies the offset of the first byte in the frame on which CRC checking begins for Wakeup Frame recognition. Offset 0 is the first byte of the incoming frame's destination address. The value in decimal and range 0 - 255.

FilterMaskPattern	This is string of packet pattern in the Ethernet frame order being destination address first. The pattern can be of maximum 128 bytes of the ethernet packet offset between 0 - 384 bytes. The Destination address offset is offset 0. The pattern should be in hex and in the form “xx xx xx xx ...”. For example: To match ARP packets the pattern is “08 06”. The FilterPatternOffset is 12.
FilterCRC	This specifies the expected 16-bit CRC value for the filter that should be obtained by using the pattern offset and the byte mask programmed for the filter. This value is compared against the CRC calculated on the incoming frame, and a match indicates the reception of a Wakeup Frame. Value is hex and should be in the format “0xFFFF”. **Note:** If the 'FilterCRC' key is present in the ini file then the supplied CRC value is used in configuring the filter. If the 'FilterCRC' is not present then the CRC is calculated and programmed in the filter.

Interrupt Polling interval [InterruptPollInterval]:

This section specifies the interrupt endpoint polling interval values.

FSPollInterval	Full speed interrupt polling interval. The values are in decimal and in the range 0 - 255. The default value is 1.
HSPollInterval	High speed interrupt polling interval. The values are in decimal and in the range 0 - 16. The default value is 4.
SSPollInterval	Super speed interrupt polling interval. The values are in decimal and in the range 0 - 16. The default value is 16.

Configuration Flags1 [ConfigFlags1]:

This section is to configure Configuration Flags1 in the EEPROM. If this section is empty then tool generates the default values (default: 0x00013380). If user want to override the default value the below key can be used.

ConfigFlags1Override	Overrides the default values for Configuration Flags1. The value must be in Hex in the format “0x00000000”.

Configuration Flags2 [ConfigFlags2]:

This section is to configure Configuration Flags2 in the EEPROM. If this section is empty then tool generates the default values (default: 0x0). If user want to override the default value the below key can be used.

ConfigFlags2Override	Overrides the default values for Configuration Flags2. The value must be in Hex in the format “0x00000000”.

Configuration Flags3 [ConfigFlags3]:

This section is to configure Configuration Flags3 in the EEPROM. If this section is empty then tool generates the default values (default: 0x001E000A). If user want to override the default value the below key can be used.

ConfigFlags3Override	Overrides the default values for Configuration Flags3. The value must be in Hex in the format "0x00000000".

Configuration Flags4 [ConfigFlags4]:

This section is to configure Configuration Flags4 in the EEPROM. If this section is empty then tool generates the default values (default: 0x0). If user want to override the default value the below key can be used.

ConfigFlags4Override	Overrides the default values for Configuration Flags4. The value must be in Hex in the format "0x00000000".

Configuration Flags5 [ConfigFlags5]:

This section is to configure Configuration Flags4 in the EEPROM. If this section is empty then tool generates the default values (default: 0x4F008031). If user want to override the default value the below key can be used.

ConfigFlags5Override	Overrides the default values for Configuration Flags5. The value must be in Hex in the format "0x00000000".

OTP Configuration [OtpConfig]:

This section is to provide the OTP signatures. This section can be used to force the OTP signature (therefore offset) while writing the OTP or generating OTP image with -z option.

If this section is not present then the OTP signature will be selected based on the OTP space availibility.

OtpSignature	Forces the this signature while writing OTP. This signature is validated with respect to availability of the OTP space. The valid values are 0xF3 and 0xF7. 0xF3: The OTP content (except signature byte) starts at offset 0x1. 0xF7: The OTP content (except signature byte) starts at offset 0x100. [0xDEAD : Writes 0xFF to signature. THIS MAKES OTP DEAD!. THE OTP CAN NOT BE USED ANY MORE. NEED TO USE EXTERNAL EEPROM IF YOU WANT TO CHANGE ANY DEVICE PARAMETERS]

LTM BELT configuration [LTMBELTConfig]:

This section is to configure the LTM BELT parameters. The below parameters are available. Without this section or any of the parameter is empty then the tool generated default value of 0 (default: 0).

Please refer to respective register descriptor in the data sheet.

LTMBELTIdle0	This key configures the value for LTM_BELT_IDLE0 register in the EEPROM. The value must be in Hex in the format "0x00000000".

LTMBELTIdle1	This key configures the value for LTM_BELT_IDLE1 register in the EEPROM. The value must be in Hex in the format “0x00000000”.
LTMBELTActive0	This key configures the value for LTM_BELT_ACT0 register in the EEPROM. The value must be in Hex in the format “0x00000000”.
LTMBELTActive1	This key configures the value for LTM_BELT_ACT1 register in the EEPROM. The value must be in Hex in the format “0x00000000”.
LTMBELTInActive0	This key configures the value for LTM_INACTIVE0 register in the EEPROM. The value must be in Hex in the format “0x00000000”.
LTMBELTInActive1	This key configures the value for LTM_INACTIVE1 register in the EEPROM. The value must be in Hex in the format “0x00000000”.

USB Common test Bus In configurations [CommonTestBusConfig]:

This section configures the common test Bus In parameters. Please refer to respective register description of the data sheet.

CommonTestBusIn	This key configures the value for COM_TEST_IN register in the EEPROM. The value must be in Hex in the format “0x00000000”.

USB PLL and AFE Test Configure [USBTestConfig]:

This section configures the USB 3 and USB to PLI and AFE test parameters. Please refer to respective register description in the data sheet.

USB3PllTestBusIn	This key configures the value for USB3_PLL_TEST_IN register in the EEPROM. The value must be in Hex in the format “0x0000”.
USB2AFETestBusIn	This key configures the value for USB2_AFE_TEST_IN field of USB2_TEST register in the EEPROM. The value must be in Hex in the format “0x00”.
USB3AFETestBusIn0	This key configures the value for USB3_AFE_TEST_IN_0 register in the EEPROM. The value must be in Hex in the format “0x00000000”
USB3AFETestBusIn1	This key configures the value for USB3_AFE_TEST_IN_1 register in the EEPROM. The value must be in Hex in the format “0x00000000”.

USB 2.0 Extension Descriptor Configuration [USB2ExtensionConfig]:

This section allows to change bmAttributes (4 bytes) section of the USB 2.0 Extension Descriptor. It is recommended to use the default values.

USB2ExtAttributes	Configures the bmAttributes value in the USB 2.0 Extension Descriptor. The value must be in hex in the format “0x00000000”. **Note:** LPM bit should match the value mentioned in the ConfigFlags0Override.

Super-Speed USB Device Capabilities Descriptor Configuration [USB3ExtensionConfig]:

This section allows below section of the Super-Speed USB Device Capabilities Descriptor. Note that the bmAttributes (LTM setting) is taken from ConfigFlags0Override.

SpeedsSupported	This key configures the wSpeedsSupported field in the Super-Speed USB Device Capabilities Descriptor. The value must be in Hex in the format "0x0". Without this key the default value is 0xE. The bit definitions are as below. Bit 0 - If set Device supports Low Speed USB. Bit 1 - If set Device supports Full Speed USB. Bit 2 - If set Device supports High Speed USB. Bit 3 - If set Device supports Super Speed USB.
FunctionalitySupport	This key configures the bFunctionalitySupport field in the Super-Speed USB Device Capabilities Descriptor. The value must be in Hex in the format "0x0" Any one of the value must be specified. 0 - Low Speed 1 - Full Speed 2 - High Speed 3 - Super speed (5 Gbps)
U1DevExitLatency	This key configured bU1DevExitLat field in the Super-Speed USB Device Capabilities Descriptor. The value must be in Hex in the format "0x0". The below values are valid. The default value of this parameter is 0xA. 0x0 - 0 0x1 - Less than 1 µS 0x2 - Less than 2 µS 0x3 - Less than 3 µS ..... = ... 0x9 - Less than 9 µS 0xA - Less than 10 µS
U2DevExitLatency	This key configured wU2DevExitLat field in the Super-Speed USB Device Capabilities Descriptor. The value must be in Hex in the format "0x000". The default value of this parameter is 0x5DC. 0x000 - 0 0x001 - Less than 1 µS 0x002 - Less than 2 µS 0x003 - Less than 3 µS .....= ..... 0x7FF - Less than 2047 µS

Appendix A. Customer Requirements

This appendix details various customer requirements for device utilization, which include the following:

- **Section A.1 "MAC Address"**
- **Section A.2 "USB Vendor ID and Logo"**
- **Section A.3 "Serial Number"**
- **Section A.4 "WHQL Logo"**

A.1 MAC ADDRESS

If an organization manufactures or plans to manufacture products using ISO/IEC 8802 standards, it must obtain an Organizationally Unique Identifier (OUI) from the Institute of Electrical and Electronics Engineers, Inc. (IEEE). The three-octet OUI can be used to generate Universal LAN MAC addresses and Protocol Identifiers, per the ANSI/IEEE 802 standard, for use in Local and Metropolitan Area Network applications.

The IEEE has been designated by the ISO Council to act as the single, world-wide registration authority for the implementation of International Standards in the ISO/IEC 8802 series. For further details contact:

IEEE Registration Authority
IEEE Standards Department
445 Hoes Lane
Piscataway NJ 08854

Phone: (732) 465-6481
Fax: (732) 562-1571
E-mail: IEEE Registration Authority
Web: http://standards.ieee.org/regauth/oui/index.shtml

A.2 USB VENDOR ID AND LOGO

Obtaining a USB vendor ID is recommended for all applications. For information on obtaining a Vendor ID, refer to the following USB organization website link:

http://www.usb.org/developers/vendor/

Note:	The USB-IF logos may be used only in conjunction with products that have passed USB-IF compliance testing and are currently on the integrators list. This requires that the company be assigned a USB Vendor ID number.

Although Microchip silicon has passed all compliance tests, final stand alone products must also go through the compliance testing process. The company must have a unique Vendor ID to satisfy the USB-IF requirements. However, if the company decides not to use the USB logo for standalone devices or if the product is a soldered down device, the company may use the Microchip VID and PID.

A.3 SERIAL NUMBER

In many cases it is desirable to associate software configuration items (for example, manually assigned IP addresses, wake up configurations, etc.) with a particular device. This allows the device to be moved from one USB port to another and/or be used simultaneously with more than one Microchip device in the same system (connected to different USB ports). Environments such as Windows use the serial number string in order to uniquely identify a device in these conditions. Thus, a unique serial number string is required for Windows to avoid reinstallation of the driver each time the device is plugged into a different port.

Therefore, if the customer wants to use the Microchip VID and PID, Microchip requires the serial number to be equal to the MAC Address.

A.4 WHQL LOGO

Refer to the following frequently asked questions regarding Windows Hardware Quality Labs (WHQL) logo usage:

Q: Some of our customers will have a different product name on their boxes (sales/marketing perspective), but don't care that Windows UI will show Microchip's company and product name. They will use our PnP ID (USB-IF assigned Vendor ID and Microchip assigned Product ID). Therefore, the standard driver and INF files can be used with no further modifications. What is the standard procedure in this case?

A: If these customers will display the WHQL logo on the package or device, they must either obtain WHQL logo compliance themselves, or complete a reseller agreement with Microchip. By completing the reseller agreement, they will have signed all the legal agreements regulating the use of the WHQL logo. No WLK testing or upload of logs will be required.

Note:	Reseller agreements can only be made with customers that have a Windows Quality Online Services (Winqual) account.

Q: Some of our customers are willing to use our unmodified driver binary, but want to use their Plug and Play (PnP) ID (customer's USB-IF assigned Vendor ID and customer's assigned Product ID) and/or reflect their Company and Product name in Windows' UI. Because of this, these customers need to modify the INF. What is the standard procedure in this case?

A: If a change is made to an INF that is already part of a logo-ed submission, the customer can utilize the Microsoft Acceptable Device and Driver Update Policy (Policy-0015). This policy, referred to commonly as Driver Update Acceptable (DUA), allows for certain changes in the INF and driver package without retesting. However, a submission must be made. Once the submission is complete, the submission number with the modified INF may be shipped.

Note:	Submissions can only be made by customers that have a Winqual account.

NOTES:

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-3326-8000
Fax: 86-21-3326-8021

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

France - Saint Cloud
Tel: 33-1-30-60-70-00

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-67-3636

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7289-7561

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820