THE EXPERT'S VOICE® IN .NET

Practical .NET 2.0 Networking Projects

Build home and office networking projects using GPS, RFID, infrared, fingerprint readers, and more

Wei-Meng Lee

Apress®

Practical .NET 2.0 Networking Projects

Wei-Meng Lee

Apress®

Practical .NET 2.0 Networking Projects

ISBN-13 (pbk): 978-1-59059-790-3

ISBN-10 (pbk): 1-59059-790-7

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Lead Editor: Ewan Buckingham
Technical Reviewer: Fabio Claudio Ferracchiati
Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Jason Gilmore, Jonathan Gennick, Jonathan Hassell, James Huddleston, Chris Mills, Matthew Moodie, Dominic Shakeshaft, Jim Sumser, Keir Thomas, Matt Wade
Project Manager: Beth Christmas
Copy Edit Manager: Nicole Flores
Copy Editor: Kim Wimpsett
Assistant Production Director: Kari Brooks-Copony
Production Editor: Katie Stence
Compositor and Artist: Diana Van Winkle, Van Winkle Design
Proofreader: Linda Seifert
Indexer: Broccoli Information Management, Inc.
Cover Designer: Kurt Krames
Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit http://www.springeronline.com.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit http://www.apress.com.

The information in this book is distributed on an "as is" basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at http://www.apress.com in the Source Code/Download section.

Contents at a Glance

About the Author ix
About the Technical Reviewer xi
Acknowledgments xiii
Introduction xv

CHAPTER 1 Sockets Programming 1

CHAPTER 2 Serial Communications 67

CHAPTER 3 Incorporating Fingerprint Recognition into Your .NET Application 123

CHAPTER 4 Infrared Programming 175

CHAPTER 5 Fun with RFID 201

CHAPTER 6 Interfacing with External Devices 241

INDEX 267

Contents

About the Author ix
About the Technical Reviewer xi
Acknowledgments xiii
Introduction xv

■CHAPTER 1 **Sockets Programming** 1

Introducing Sockets Programming 1
Creating Your Own Multiuser Chat Application 3
Using the TcpClient and TcpListener Classes for Network Communications 3
Building the Server 7
Building the Client 16
Testing the Chat Applications 24
Building an Advanced Multiuser Chat Application 25
Defining Your Own Communication Protocol 25
Protocol Description 26
Walking Through the Features 27
Building the Server. 29
Building the Client 45
Testing the Application 66
Summary 66

■CHAPTER 2 **Serial Communications** 67

Some Serial Communication Basics 68
Chatting Using Serial Ports 69
Hardware Needed 70
Building the Chat Application 72
Instantiating the SerialPort Class 73
Listing All the Available Serial Port Names 74
Opening a Serial Port 75
Disconnecting a Serial Port. 77
Sending Data Using the Serial Port 78
Receiving Data on the Serial Port 79
Testing the Application 80
Transmitting Unicode Characters 81
Connecting to Other Serial Devices. 84

Chatting Using Serial Ports on the Pocket PC 87
Hardware Needed 87
Building the Application 88
Coding the Application 89
Creating a Mapping Application Using a GPS Receiver and Microsoft Virtual Earth 93
Building the Application 97
Creating the HTML File Containing the Virtual Earth Map 98
Coding the Application 100
Displaying the Coordinates of the Map 103
Connecting to a GPS Receiver. 106
Plotting Saved Path 115
Summary 121

CHAPTER 3 Incorporating Fingerprint Recognition into Your .NET Application 123

Using the GrFinger SDK 124
Creating the Application 125
Coding the Application 128
Wiring Up All the Controls 130
Testing the Application 147
Summary 148

CHAPTER 4 Infrared Programming 175

Introducing IrDA 175
Creating Infrared Communications Between Windows Mobile Devices 176
What You Need 176
Creating the Project 177
Coding the Application 178
Receiving Messages 180
Displaying the Received Messages 183
Sending Messages 184
Compiling and Deploying the Application 188
Creating Infrared Communications on the Desktop 188
What You Need 189
Creating the Project 189
Importing the Namespaces 190
Declaring the Constants and Member Variables 191
Coding the Form1_Load() Event 191

Coding the ReceiveLoop() Subroutine 192
Coding the ReceiveMessage() Function 193
Coding the Delegate and the UpdateTextBox() and UpdateStatus() Subroutines 195
Coding the SendMessage() Subroutine 195
Coding the Send Button Control 198
Testing the Application 199
Summary 199

CHAPTER 5 **Fun with RFID** 201

Introducing RFID 201
Building an Attendance-Taking Application 203
RFID Reader #1: Parallax's RFID Reader Module 204
RFID Tags 204
Setting Up the Reader 205
Building the Application User Interface 207
Coding the Application 213
Testing the Application 227
RFID Reader #2: PhidgetRFID 227
RFID Tags 228
Building the Sample Application 229
PhidgetRFID APIs 232
Coding the Application 232
Testing the Application 238
Comparing the Two RFID Readers 239
Summary 239

CHAPTER 6 **Interfacing with External Devices** 241

Components Used 242
Sensor 242
Webcam 243
Connecting the Sensor to the PC 243
Connecting the PING Sensor 245
Programming the PING Sensor 246
Integrating with the PC 250
Programming the Webcam 255
Using the AVICap Class 256
Summary 265

INDEX 267

About the Author

WEI-MENG LEE is a technologist and founder of Developer Learning Solutions (`http://www.learn2develop.net`), a technology company that specializes in hands-on training in the latest Microsoft technologies. Wei-Meng speaks regularly at international conferences and has authored and coauthored numerous books about .NET, XML, and wireless technologies, including *ASP.NET 2.0: A Developer's Notebook* and *Visual Basic 2005 Jumpstart* (both from O'Reilly Media). He is also the coauthor of *Programming Sudoku* (from Apress). Find out about the latest books and articles by Wei-Meng at his blog: `http://weimenglee.blogspot.com/`.

About the Technical Reviewer

FABIO CLAUDIO FERRACCHIATI is a senior consultant and a senior analyst/developer using Microsoft technologies. He works for Brain Force (`http://www.brainforce.com`) in its Italian branch (`http://www.brainforce.it`). He is a Microsoft Certified Solution Developer for .NET, a Microsoft Certified Application Developer for .NET, and a Microsoft Certified Professional, as well as a prolific author and technical reviewer. Over the past ten years he has written articles for Italian and international magazines and coauthored more than ten books about a variety of computer topics. Visit his blog at `http://www.ferracchiati.com`.

Acknowledgments

Although the "Acknowledgments" section of a book is always placed at the front, it is always the last thing an author writes; and after spending several months working on the book together with different groups of people, most names are not even mentioned (or at least mentioned in passing). Yet, without the collective efforts of these people, the book would never have been possible.

This book is no exception.

Now that this book is done, I can finally look forward to seeing it on the shelves of bookshops. I want to take this opportunity to thank my editor, Ewan Buckingham, for his guidance and valuable suggestions for making this book a better read. Fabio Claudio Ferracchiati, the technical reviewer for this book, also deserves special mention because he painstakingly tested every project in this book and made several good suggestions for improving the quality of the code. Thank you, Ewan and Fabio. And this book would never be possible without the great patience of its project manager, Beth Christmas. Beth was extremely patient with me while I was juggling writing this book and working on my daytime projects. For this, I am very grateful, Beth! To the production crew, Kim Wimpsett and Katie Stence—thanks for the great job of polishing my work!

I also want to thank Rod Paddock (editor-in-chief at *CoDeMagazine*) for publishing my RFID chapter as an article in the Nov/Dec 2006 issue of *CoDeMagazine*. And I want to express my gratitude to Ryan Clarke (from Parallax) and Matt Trossen and Jennie Jetter (from Phidgets USA) for their help in getting me started with RFID. They have been very patient in guiding an hardware-idiot (that's me) and for this I am very grateful to them! Last but not least, I want to thank Lori Piquet, my editor at DevX.com. Lori has always been very open to my new article ideas, and her support has provided me with the avenues to try new project ideas. For this, I am indebted and very grateful to you, Lori.

Finally, thank *you* for picking up this book, and I hope you have a great time with the various projects discussed in this book.

Introduction

Practical .NET 2.0 Networking Projects demonstrates some of the key networking technologies that are being made easily accessible through the .NET Framework 2.0. It discusses communication between wired machines and between networks and mobile devices. The book teaches you about the technologies by walking you through sample projects in a straightforward and direct way.

This book contains six chapters, each covering a specific aspect of network programming. You'll use the various APIs within the .NET Framework as well as third-party SDKs to build a variety of cutting-edge networking applications that cover everything from Bluetooth and RFID communication to sockets programming and chat servers. You'll build working examples for each project, which you can also customize and use for your own purposes. The featured projects cover the following.

Chapter 1: Sockets Programming

Writing networked applications is one of the most interesting aspects of programming. This is especially intriguing when you see your applications successfully communicating over the network. In this chapter, you will build a chat application that works similarly to Windows Live Messenger (or ICQ) using TCP/IP. Using the chat application, you will learn how network programming happens in .NET and the various challenges you'll encounter when building a multiuser chat application.

Chapter 2: Serial Communications

Serial communication is one of the oldest mechanisms for devices to communicate with each other. Starting with the IBM PC and compatible computers, almost all computers are equipped with one or more serial ports and one parallel port. As the name implies, a *serial* port sends and receives data serially, one bit at a time. In contrast, a *parallel* port sends and receives data eight bits at a time, using eight separate wires.

Despite the comparatively slower transfer speed of serial ports over parallel ports, serial communication remains a popular connectivity option for devices because of its simplicity and cost-effectiveness. Although consumer products today are using USB connections in place of serial connections, still a lot of devices use serial ports as their sole connections to the outside world.

In this chapter, you will learn how to communicate with other serial devices using the new `SerialPort` class available in the .NET Framework 2.0 and the .NET Compact Framework 2.0.

In particular, you will build three projects that illustrate how to use serial communications. The first project is a chat application that allows two computers (connected using either a serial cable or a Bluetooth connection) to communicate. And using the foundation of this application, you can extend it to communicate with other external serial devices such as cellular phones. You will learn how to use the AT commands to programmatically control your mobile phones through a serial Bluetooth connection. The second project is a Pocket PC chat application, which is similar to the first project. The third application shows how to communicate with a GPS receiver and then extract the useful data for displaying the current location on a map.

Chapter 3: Incorporating Fingerprint Recognition into Your .NET Application

Biometric recognition is one of the most reliable ways to confirm the identity of an individual. And by now, most people should be familiar with the Microsoft Fingerprint Reader. Using the Microsoft Fingerprint Reader, you can now log in to your computer by placing your finger on the reader. You can also use the application provided by the Fingerprint Reader to save your user IDs and passwords for websites that require them for authentication. You can then use your fingerprint as a key to retrieve the user IDs and passwords for logging into these sites securely. The Microsoft Fingerprint Reader removes the hassle of remembering different passwords for different sites.

In this chapter, I will show you how you can use the GrFinger Fingerprint SDK to integrate the Microsoft Fingerprint Reader into your .NET 2.0 Windows applications. In particular, you will build a visitor identification system whereby users visiting your office can register at the reception desk. Once a user is registered, the next time the user visits the office, he can simply scan his fingerprint, and the system will register his visit. Schools can also adapt this application for attendance-taking purposes, such as in big lecture theaters where attendance must be taken rapidly and efficiently.

Chapter 4: Infrared Programming

With all the buzz around WiFi, Bluetooth, and other wireless technologies, it's easy to overlook one of the simplest and most common forms of wireless communications—infrared. Anyone who has ever used a remote control has used it! Infrared uses the invisible spectrum of light just beyond red in the visible spectrum. You can use it in applications for short-range, point-to-point data transfer. Because it uses light, line-of-sight is a prerequisite for infrared. Despite this limitation, infrared is increasingly popular in devices such as digital cameras, PDAs, and notebook computers.

In this chapter, I will show you how to build an application that allows two devices (as well as computers) to communicate wirelessly using infrared. You can adapt the programming technique illustrated in this chapter for other programming tasks, such as writing wireless network games, and so on.

Chapter 5: Fun with Radio Frequency Identifications (RFID)

Radio frequency identification (RFID) is one of the buzzwords receiving a lot of coverage in the IT world lately. An RFID system is an identification system that uses radio waves to retrieve data from a device called a *tag* or a *transponder*. RFID is all around us in our daily lives—in the supermarkets, libraries, bookstores, and so on. RFID provides a quick and efficient way to collect information, such as stocktaking in a warehouse or tracking the whereabouts of items.

In this chapter, you will learn how to build a Windows application that incorporates RFID technology for data collection. You will use two RFID readers and understand their pros and cons.

Chapter 6: Interfacing with External Devices

Today, a webcam is a common peripheral that most people can easily afford; and it's used most often for video conferencing. But what can you do with your webcam besides video conferencing? For .NET developers, the answer is plenty; and you will be glad to know that integrating a webcam with a Windows application is not as difficult as you might imagine.

Besides integrating a webcam with your application, you can connect your Windows application to an external device such as a sensor to monitor the movements of the surroundings.

In this chapter, you will build a security system by interfacing a Windows application with an external sensor and a webcam so you can monitor for unwanted activities. You will be able detect the proximity of an intruder and use the webcam to record the intruder's movements.