

Designing Embedded Internet Devices

A Practical Guide to Hardware and Software Design Using the TINI Microcontroller



Dan Eisenreich | Brian DeMuth

CD-ROM Included

Contains Java code for all applications and a fully searchable eBook version of the book!

Copyrighted Material



Please note that the previous printing included a CD-ROM.

The material is now only available on the companion website:
<http://www.elsevierdirect.com/companions/9781878707987>

Designing Embedded Internet Devices

This page intentionally left blank

Designing Embedded Internet Devices

*by Dan Eisenreich
Brian DeMuth*



Newnes

An imprint of Elsevier Science

Amsterdam

Boston

London

New York

Oxford

Paris

San Diego

San Francisco

Singapore

Sydney

Tokyo

Newnes is an imprint of Elsevier Science.

Copyright © 2003, Elsevier Science (USA). All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.



Recognizing the importance of preserving what has been written, Elsevier Science prints its books on acid-free paper whenever possible.

Library of Congress Cataloging-in-Publication Data

ISBN: 1-878707-98-1

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library.

The publisher offers special discounts on bulk orders of this book.
For information, please contact:

Manager of Special Sales
Elsevier Science
200 Wheeler Road
Burlington, MA 01803
Tel: 781-313-4700
Fax: 781-313-4882

For information on all Newnes publications available, contact our World Wide Web home page at: <http://www.newnespress.com>

Transferred to Digital Printing in 2012

Printed in the United States of America

Acknowledgments

We would like to gratefully acknowledge

... the fine folks at Dallas Semiconductor for squeezing such a capable JVM on an 8-bit microcontroller.

... all of the hard work proofreading and organizing the draft for the book by the folks at Newnes (Kelly, Carol, Harry).

Dedication

Dan

To Keisha and Emily

Brian

To Linda, whose love and support made this possible.

This page intentionally left blank

Contents

What's on the CD-ROM?, xi

Chapter 1: Introduction	1
Why should you read this book?	1
What this book will do for you	2
How this book is laid out	2
How to get the most out of this book.....	3
What you should already know	4
Chapter 2: Computer Networks	5
Network Hardware	5
Network Addresses	13
Network Communication	17
Ports and Sockets	21
Summary	22
References	22
Chapter 3: Java Essentials for Embedded Networked Devices	25
The Java Development Kit	25
Serial Port Communications	30
Significant Topics for Review in the Java Language.....	37
Summary	92
References	92
Chapter 4: Overview of Embedded Networked Devices	93
Chapter 5: Getting Started with TINI	101
What Is TINI?	101
Getting Started	102
TINI libraries, utilities, TINI 1-wire libraries	107
1-Wire Libraries	117
Summary	126
References	126

Chapter 6: The TINI Hardware	127
What is TINI?	127
The Various Components of the TINI Stick	130
The Memory	139
The E10/E20 Socket Board	158
Other TINI socket accessories	173
Summary	175
References	176
Chapter 7: The TINI Software	177
JavaKit	177
The TINI Loader	180
The TINI Firmware	182
The TINI API3	185
Slush, the TINI Operating System Shell	190
Programming TINI	204
Other Tools to Make Life Simpler	232
Other TINI Software	241
Conclusion	243
References	243
Chapter 8: Enhancing TINI	245
Adding 512 kbytes on the SIMM	245
Adding 512 kbytes Flash Memory	251
Adding a SIMM Connector	253
Adding Memory-mapped Devices	255
Summary	294
References	294
Chapter 9: TINI Serial and Parallel I/O	295
Serial Ports	295
Parallel Ports	331
Summary	344
References	344
Chapter 10: 1-Wire Basics for TINI	345
What Is the 1-Wire Bus?	345
How the 1-Wire Bus Works	346
The 1-Wire Bus Protocol	349
1-Wire Bus Commands	353
Cyclic Redundancy Check (CRC)	359

1-Wire Device Example: the DS2405 Addressable Switch	362
1-Wire Device Example: the DS1920 Thermometer iButton	366
Connecting a PC to the 1-Wire Bus	370
The 1-Wire Java API	380
How TINI Communicates with the 1-Wire Bus	418
Summary	431
References	431
Chapter 11: The I²C Bus	433
What Is the I ² C bus?	433
The I ² C Bus in More Detail	434
How TINI Does I ² C	440
Summary	465
References	466
Chapter 12: Controller Area Network	467
What Is the CAN Bus?	467
The CAN Bus in More Detail	469
How TINI Does CAN	483
Summary	508
References	508
Chapter 13: Connecting TINI to an IP Network	511
The ipconfig Command	511
Using PPP	518
Summary	566
References	567
Chapter 14: A Few Final Thoughts	569
The Future of TINI	569
Connecting Your Device	570
What's Been Done with TINI	573
References	575
Index	577

This page intentionally left blank

What's on the (companion website)?

Included on the accompanying (companion website)

- A directory containing all of the example programs in the book organized by chapter.
- Appendix A, providing information on TINI components and pinout
- Appendix B, a listing of `ByteUtils.java`, which is used in a number of the example programs.
- Appendix C, a compilation of simple input/output circuits that can be easily connected to various types of I/O for sensing or controlling external devices.

Each example from the book is in a separate folder that is named according to the program name and the corresponding listing number from each chapter, so the proper listing ought to be very easy to find.

To compile these listings using the supplied makefile (for linux) or `build.bat` (for Windows) you will need to set your environment variable as instructed in Chapter 3 of the book. Specifically, `TINI_HOME` must be set to point to the TINI API installation directory and `OW_HOME` must be set to point to the 1-wire API installation directory.

The `build.bat` and Makefiles included with the compile instructions for each listing assume that you are using API 1.02d or e.

On Windows/DOS, if you get an “Out of environment space” error, then you will need to increase the default environment space available for MS-DOS programs. To do this, add this line to your `config.sys` and then reboot your system:

```
SHELL=C:\COMMAND.COM C:\ /E:2048 /P
```

See <http://support.microsoft.com/default.aspx?scid=kb;EN-US;q230205> for more information.

This is how we are setting the `CLASSPATH` for Windows (95/98/2000/NT/XP):

```
SET CLASSPATH=c:\jdk1.3.1\lib\;c:\jdk1.3.1\lib\comm.jar;.
SET OW_HOME=c:\opt\lwire
SET CLASSPATH=%CLASSPATH%;%OW_HOME%\lib\OneWireAPI.jar
SET TINI_HOME=c:\opt\tini1.02d
SET CLASSPATH=%CLASSPATH%;%TINI_HOME%\bin\tini.jar;
    %TINI_HOME%\bin\tiniclasses.jar
```

This is how we are setting the CLASSPATH for Linux:

```
CLASSPATH=/usr/java/jdk1.3/lib/:  
    /usr/java/jdk1.3/commapi/comm.jar OW_HOME=/opt/onewire  
CLASSPATH=$CLASSPATH:$OW_HOME/lib/OneWireAPI.jar.  
TINI_HOME=/opt/tini  
CLASSPATH=$CLASSPATH:$TINI_HOME/bin/tini.jar:  
    $TINI_HOME/bin/tiniclasses.jar  
export TINI_HOME OW_HOME CLASSPATH
```

CHAPTER 1

Introduction

Why should you read this book?

The target audience of this book is anyone interested in merging practical electronic devices with the Internet: students, teachers, home automation enthusiasts, hobbyists, and small businesses. Computer programmers looking for a gentle introduction to the world of hardware will benefit, as will hardware designers looking to expand their skills into the realm of JAVA programming. College engineering and computer science departments will find in this book a wealth of possibilities for lab projects that expose students to cutting-edge technology with minimal expense. Why should you read this book? The best reason of all: fun. This book will provide anyone interested in tinkering with hardware on the net hours of fun. Another reason: the future. Even if you're not interested in making hardware, this book will give you a practical glimpse into what the real future and potential of the Internet is.

The first Internet wave connected people via computer, popularizing things such as email, search engines, and online shopping. The next wave is going to be Internet appliances: electronic devices connected to the Internet. Ever wondered how those "live internet cams" work? They're probably the most recognizable example of hardware connected to the web. In the past they were tremendously expensive and tended to be supported by engineering departments as an experimental thing. The future is going to see a tremendous expansion of low-cost, practical devices, connected to the Internet for home use.

A good example might be a future VCR. Have you ever found yourself at work, wishing you had set the VCR to program your favorite show? A VCR that was an Internet appliance would give you the capability of programming your VCR from a web page, wherever you may be. Another good example of an Internet appliance might be a piece of hardware that allows you to control your thermostat, your lights,

and your water heater from a web site. Cheap Internet appliance technology will make controlling devices in your home from a web page commonplace. Will everything in your house be “on the web” in the future? Probably not. But the growing number of people with Internet on their desk and the proliferation of cell phones and PDAs that can interact with web pages represents a growing market. That market is raising the attention of countless companies both large and small, looking to make products to put into every home. This book will examine commercial technology, discussing in great detail an inexpensive web-enabled microcontroller that can be used to connect a variety of devices to the web.

Why should you read this book? So when the next big Internet wave happens, you’ll be in front of it! (And, there is that fun thing, too.)

What this book will do for you

This book is a complete introduction to Internet-enabled devices. We provide all of the information you need to inexpensively build your own web-enabled hardware. Specifically, we’ll show you:

1. The basic terms and concepts required to understand the technology of web-enabled devices. This includes detailed sections on networking and Java programming.
2. A quick overview of commercially available, web-enabled, microcontrollers, comparing their price and availability.
3. A step-by-step examination of TINI¹, the Tiny InterNet Interface, a commercially available hardware/software package designed for use as a web interface to hardware. We’ll examine the hardware, software, and available enhancements in detail.

How this book is laid out

The information in this book is divided into 14 chapters.

Chapters 1 through 4 provide basic technical definitions with respect to networking and Internet clients and servers. We’re going to go into some detail on Java, the modern object-oriented, internet-ready, programming language rapidly becoming the language of choice for network applications. We’re not going to teach you Java, but we are going to cover some of the key features of the language that are very relevant to our topic, and are frequently not taught in the average Java course.

¹ TINI is a registered trademark of Dallas Semiconductor. TINI can be purchased on the iButton website, <http://www.ibutton.com/TINI/>

Chapters 5 through 8 discuss, in great detail, the hardware and software behind TINI, Dallas Semiconductor's commercially available, web-enabled microcontroller. This will include a high-level discussion of how the system works and an explanation of how to obtain and set up the hardware and software on your own Windows or Linux PC. Following that, we'll present an indepth discussion of the TINI hardware. We're going to explain what it is and how it works. Then, in a similar fashion, we're going to take apart the TINI software. Finally, with detailed technical discussions of the TINI hardware and software as background, we'll present detailed sections on how to upgrade the TINI hardware.

Chapters 9 through 12 consist of detailed technical discussions of the various I/O busses provided by TINI. These include 1-Wire, CAN, I²C, and standard serial and parallel ports. Every topic is profusely illustrated, and filled with examples.

Chapter 13 discusses how to connect TINI to a network, including how to attach a modem to TINI. This allows you to dial-in to TINI and have it act as a PPP server, or dial-out to an Internet service provider (ISP) allowing PPP clients to run on TINI.

Finally, Chapter 14 provides a summary.

How to get the most out of this book

Our book is not a novel, requiring cover-to-cover reading; rather, it's a combination technology intro, how-to guide, and reference manual. What one needs to read, and how to get the most out of that reading, depends on who you are and what your goals are.

1. For those who are simply curious about how web-enabled devices might be implemented, chapters 1, 2, 4, and 5 should be read. They will give the necessary background and technical depth. Chapters 10, 11, and 12 can be skimmed to provide additional information on the breadth of devices that can be connected to a web-enabled device.
2. For those who are unfamiliar with microcontrollers but plan on implementing a web-enabled device of some sort, chapters 1 through 7 should be read. There's enough information there to get you up and running. The remaining chapters can be skimmed to provide technical ideas, and serve as an excellent reference.
3. For those who are familiar with TINI, chapters 6 through 13 are an excellent reference.

What you should already know

While you don't have to be an engineer or computer scientist to have a lot of fun with this book, there are a few prerequisites. First, we assume that you understand basic electronics such as how to read circuit diagrams and how to use a soldering iron to build simple electronic circuits. Lastly, but most importantly, we assume basic knowledge of the Java programming language. We're going to provide detailed explanations of how to get and install the Java language for your Windows or Linux computer, as well as how to obtain the appropriate technology-specific software and class libraries. We do this to provide a solid foundation on which to build our later examples. We're going to explain in detail the Java examples we provide. But our book isn't going to teach you Java. If you're unfamiliar with Java, you may want to consider the following references.

1. Java 2 Platform, Standard Edition, v 1.3 API Specification.
<http://java.sun.com/j2se/1.3/docs/api/index.html>
2. Campione, Mary and Walrath, Kathy.
The Java Tutorial Second Edition: Object-Oriented Programming for the Internet,
<http://web2.java.sun.com/docs/books/tutorial/>.
Addison-Wesley, 1998.
3. Flanagan, David.
Java in a Nutshell: A Desktop Quick Reference,
O'Reilly & Associates, 1999.
4. Flanagan, David.
Java Examples in a Nutshell,
O'Reilly & Associates, 1999.
5. Wu, C. Thomas.
An Introduction to Object Oriented Programming with Java,
McGraw Hill College Div, 1998.

We hope you enjoy this book.

Computer Networks

Since this book is about designing network-enabled devices, it probably makes some sense to start with a little discussion about networks. This is not a substitute for a complete lesson in building computer networks, but just enough to establish some terms and common ground to get you through this book. Throughout this book we'll be talking primarily about 10base-T Ethernet networks using the TCP/IP protocols (there are many other ways to network computers together that are far beyond the scope of this book; see the reference section at the end of the chapter).

Network Hardware

So what is a computer network? A network is a collection (two or more) of computers that are connected together for sharing data and resources. Networks are commonly categorized by their geographical size. A **local area network** (a LAN) is a group of computers and associated devices that share a common communication line and are typically located within a small geographic area, such as in your home or in an office building, or even a single department in a company. A local area network may connect as few as two or three users, as in a home network, or as many as several hundred users, as in a business. A **metropolitan area network** (a MAN) is a bit more diverse, incorporating computers and LANs that are distributed over a few miles of area, as you would find between corporate buildings or on a college campus. A **wide area network** (a WAN) is a widely geographically dispersed computer network that is composed of multiple LANs and MANs and can be spread across a country or around the globe. The **Internet** (sometimes simply referred to as “the Net”) is a worldwide network of many computer networks (LANs, MANs, and WANs) that is accessible to hundreds of millions of people worldwide.

A really simple network

Connecting two computers together forms the simplest computer network, as shown in Figure 2-1. The computers, often referred to as **hosts** or **nodes** on a network, are connected to the network through a **network interface card (NIC)**. The network interface is a computer circuit board or card (or sometimes a single integrated circuit) that provides the logic for sending and receiving data from and to the host computer. There are a number of ways to connect these computers but by far the most common way is by using an Ethernet connection.

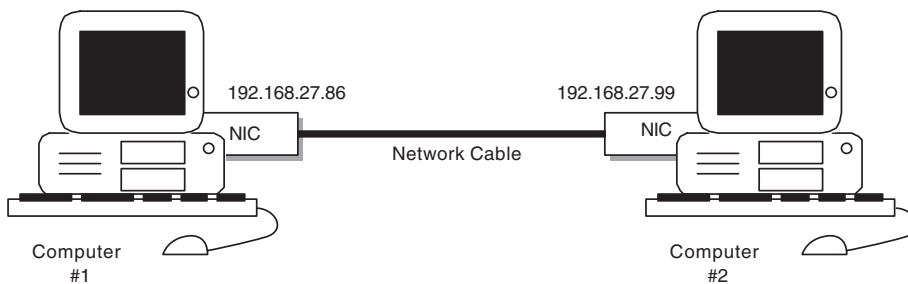


Figure 2-1: A really simple computer network

Ethernet is the most widely used local area network technology these days, particularly for home networks and small offices. It is fully described and specified in IEEE standard 802.3 (if knowing that sort of thing is important to you). The most commonly installed Ethernet system is called 10BASE-T and provides transmission speeds up to 10 Mbps. The designation **10BASE-T** is a shorthand identifier: The “10” in the media type designation refers to the transmission speed of 10 Mbps, the “BASE” refers to baseband signaling (not broadband), which simply means that only a single Ethernet signal set is carried on the cable, and the “T” represents twisted-pair. The reason we even mention this is because other types of Ethernet are becoming more popular like 100BASE-T (100 mbps) and 100BASE-F (“F” for fiber) but for this book we are only really concerned with 10Base-T and 100 Base-T (often printed as 10/100Base-T).

Another thing we need to note about this simple network of two computers is the cable directly connecting the two computers, commonly called a **crossover cable**. A crossover cable is a special type of Ethernet cable that is used to interconnect two computers by “crossing over” (reversing) their respective pin contacts for transmitted and received data. It allows the data transmitted by one computer to be connected to the receiving pin on the other and vice-versa. While this may seem like a trivial point, it is important to know that this is a crossover cable and what it does.

One last thing: each node in the network has a unique address, its network address. This is not a big deal with a simple network of two computers but as the networks expand and LANs are connected to other LANs and WANs, this is increasingly important. We will talk more about this address in a subsequent section.

A small office and home network

Connecting multiple computers together to form a home or small office network is almost as simple as the previous two-computer network, just involving a few more components. Refer to Figure 2-2 for a slightly more complex network configuration.

The first thing you will notice is the use of a **network hub**, usually simply called a

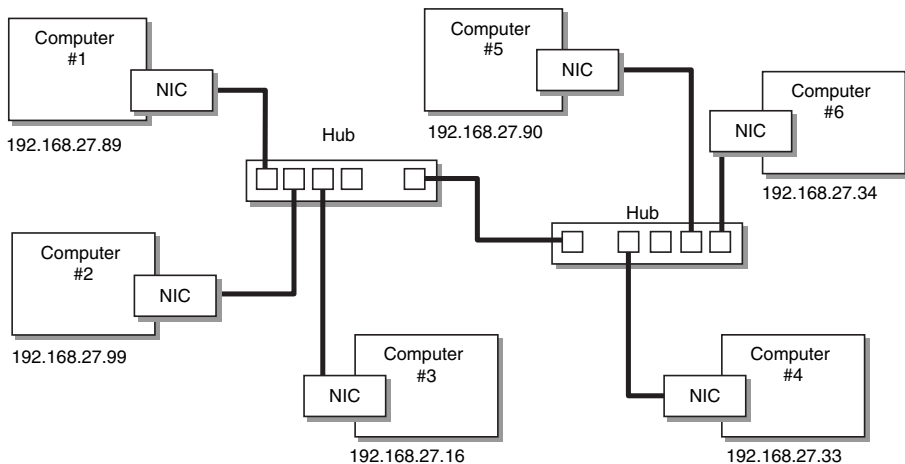


Figure 2-2: A complex network

hub. A hub is a connector where network data arrives from one or more directions and is forwarded out in all of the other directions. Each of the computers in our network is connected to the network through a hub. Internal to the hub, the transmit wires from one computer are connected to the receive wires of the others; in other words the hub also performs the crossover function that the crossover cable did in the two-computer network. So this network is connected with network cables that are “straight through.” The wires on pin 1 through 8 are connected to pins 1 through 8 on the opposite end, with no crossover. Hubs are connected to each other through an uplink port. The **uplink port** is similar to an ordinary port but the wires have not internally been reversed so it can be connected to another hub to increase the number of nodes in your network. If the hub does not have an uplink port then hubs can be connected to each other with a crossover cable.

Again, notice that each node in the network has a unique address, its network address. We will talk more about this address in a subsequent section.

Connecting a LAN to the Internet (or a LAN, MAN, WAN)

Broadband connection

Now that we have our local area network as discussed above, how would we then connect this to other networks, either another LAN or a WAN or even the Internet? This is done with a wide variety of network devices. Some of these are:

- **Hub** – A hub is the generic name for a device where several nodes are connected to a network. A hub may be a repeater, switch, or a router. A hub may perform the function of several of these components.
- **Gateway** – A gateway is a generic name for a device that is used to connect multiple networks together. A gateway may be a repeater, bridge or router.
- **Repeater** – A repeater is a network hub that serves as a connection between 2 segments of a network. Repeaters only amplify all information they receive and pass it through to all of its ports.
- **Router** – A router is a smart hub, which can connect segments of different networks that use a different protocol.
- **Switch** – A switch is a network hub that acts like a repeater but instead of passing information on to all of its ports, it establishes a direct connection to the destination port.
- **Bridge** – A bridge is a device for connecting segments of a network or multiple networks that use the same protocol.

All of these devices are somewhat similar—in fact, the functions of several are often combined in one device. We will connect our LAN to the Internet with one of these generic network gateway devices. There are several types of gateways: repeaters, bridges, and routers. The technical distinction between these components is based on which layer of the Internet Protocol they operate on and that is really beyond the scope of this chapter. We will then connect our gateway to a modem. A modem (and there are many types of modems) can also act as a gateway from a LAN to a WAN but are usually separate devices. The modem modulates outgoing digital signals from a computer or other digital device to analog signals suitable for transmission on conventional telephone lines, cable TV coax, Digital Subscriber Line (DSL) or fiber and then demodulates the incoming analog signal and converts it to a digital signal. This is the type of gateway shown in the following network, Figure 2-3.

This LAN could also be connected to another LAN or WAN with a simple hub or gateway without the modem. But if we opt for the hub, then this would not keep this

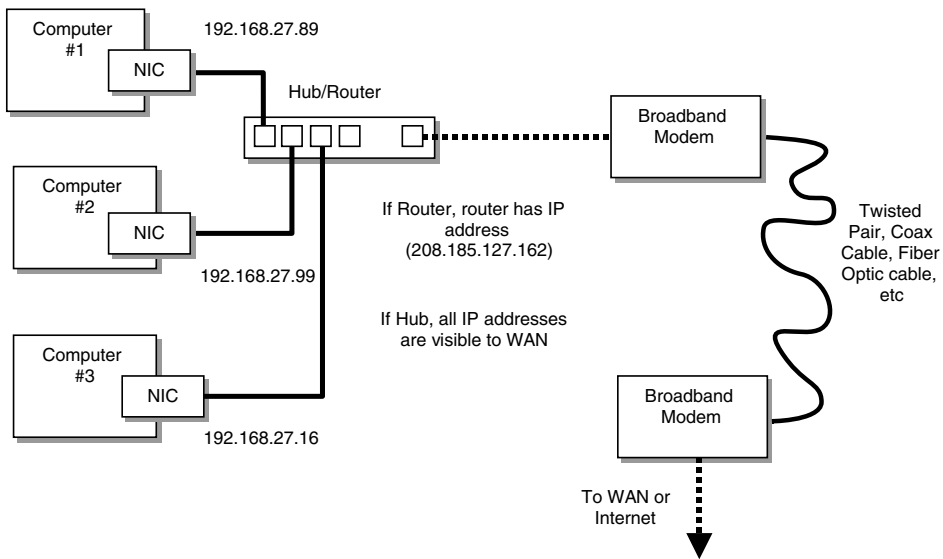


Figure 2-3: Broadband network

local area network as a separate network. This network would then be a part of the larger WAN. The hub passes all data from its input to all of the lines on the network. This is one of the reasons for creating smaller networks or sub-networks—to isolate data transmission between computers.

Alternate broadband connection

An alternate way to connect a simple network is to use a computer with two network interface cards as shown in Figure 2-4. In essence, this computer (and appropriate software) acts as the gateway and router for your local area network. It is important to note that one NIC has a network address that is known only to the WAN or Internet and the other NIC has a network address that is known only to the LAN. These addresses can be either static or dynamically allocated. This would allow you to use static IP addresses on your LAN and connect to your ISP with static or dynamic addresses. Windows 98 Second Edition directly supports multiple NICs and Internet sharing, and there are numerous software packages that can add this function to either Windows or Linux.

Dial up connection

A final alternative for connecting your LAN to a WAN or Internet is the use of a PPP connection through a dial-up modem. This is shown in Figure 2-5. **Point-to-Point Protocol (PPP)** is a protocol for communication between two computers using a serial interface, typically a personal computer connected by phone line to a service provider. But this can be as simple as a serial line with no modem. Essentially, it packages your computer's TCP/IP packets and forwards them to the server.

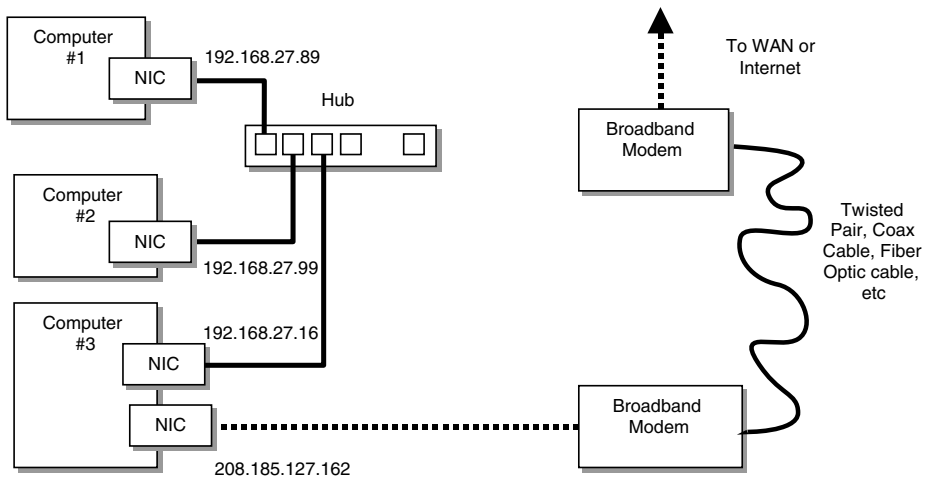


Figure 2-4: Alternate broadband network

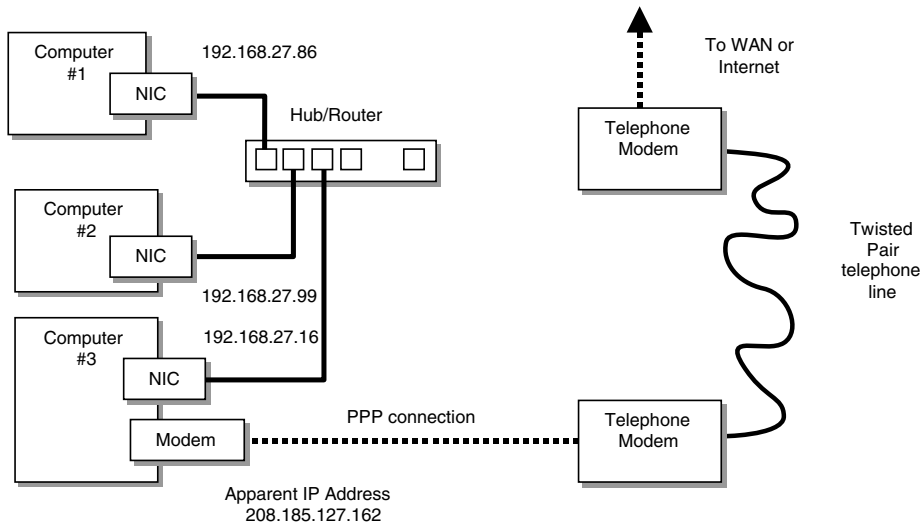


Figure 2-5: PPP network connection

The 10/100 Base-T cable

Ethernet uses a special type of cable called twisted pair. **Twisted pair** cable is made of four pairs (in this case) of insulated copper wires that are twisted around each other in pairs to reduce electromagnetic induction between pairs of wires. The combination of these four sets of twisted wires forms a network cable that is commonly referred to as **Category-5** or **CAT 5** cable. It derives that name from the American National Standards Institute/Electronic Industries Association (ANSI/EIA) Standard 568, in which this cable is specified. That same specification lists several standards that specify categories of twisted pair cabling systems (wires and connectors) in terms of the maximum data rates that they can reliably sustain. This specification describes the cable material as well as the types of connectors and junction blocks to be used in order to conform to a particular category. CAT5 cable is different from the wire you buy at your local hardware store for connecting your phone or computer modems to a wall jack, which is not twisted pair, but is a side-by-side wire also known as *silver satin*.

The two most popular specifications are CAT 3 and CAT 5. While the two cables may look identical, CAT 3 is intended for a lower data rate and can cause transmission errors if used for faster speeds. CAT 3 cabling is for signals that are 16 MHz or less and is suitable for 10base-T networks, while CAT 5 cable must pass a 100-MHz test to be suitable for 100base-T networks.

The connector on a 10/100 BASE-T cable is an RJ45, which is short for Registered Jack-45. This is an eight-pin connector that is commonly used for network cables, especially Ethernet. The RJ-45 connectors look very much like the modular connectors

Table 2-1: Network cable wire categories

Category	Maximum Data Rate	Typical Applications
CAT 1	< 1 Mbps	Voice telephone service Integrated Services Digital Network (ISDN)
CAT 2	4 Mbps	
CAT 3	16 Mbps	10Base-T Ethernet
CAT 4	20 Mbps	
CAT 5	100 Mbps	100Base-T Ethernet
CAT5e	100 Mbps	100Base-T Ethernet with improved transmission
CAT6	250 Mbps	Proposed Standard
CAT7	600 Mbps	Proposed Standard

on modern telephones (RJ-11), but they are somewhat wider as they have eight wires in them. There are many, many types of RJ style connectors and they vary widely.

Since we have talked about twisted pair cable and all that, perhaps it would make sense if we also discussed the detail of the Ethernet cable (the wire colors/order) and crossover cable so you can wire up your network. Essentially, the difference between 10Base-T and 100Base-T is the data rate. In Table 2-1 we can see that 10base-T needs at least a CAT3 cable to support the 10 mbps data rate error free and 100Base-T needs CAT5 wire. If you are wiring up a new network it is probably better to make it with CAT5 and wire it for 100 mbps, even if you are going to use it for 10Base-T communications. It will work just fine for 10 mbps and it's ready to upgrade to 100Base-T. It's also easier to find CAT5 cable and it is not noticeably more expensive. If you use CAT3 cable for a 10Base-T, when you upgrade you will need new wires and you run the risk of getting all of your 10-mbps wiring confused with 100-mbps wire—and you'll just have a great big mess on your hands.

Shown in Figure 2-6 is the wiring for a straight-through (not a crossover) cable that is more typical for network connections. The wire color order is the same for both ends of the cable.



Figure 2-6: 10/100Base-T Straight-through cable

Shown in Figure 2-7 is the wiring for the crossover cable that we used for our simple two-node network above. You will notice that the wire color order is not the same for both ends of this cable. You should carefully label or mark your crossover cables so that they are not easily confused with straight-through cables. I make mine from a different color cable so that they are immediately obvious.

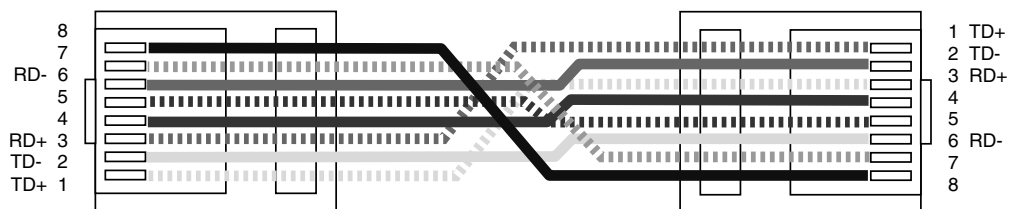


Figure 2-7: 10/100Base-T Crossover cable

Table 2-2 lists the wire colors and pin numbers for both straight-through and cross-over cables. You may find reference to that fact that only pins 1, 2, 3 and 6 are used in 10/100 Base-T network cables. This is generally true but there are some networks like 100Base-T4 that use the remaining wires, and faster networks like 1000Base-T also use all four pairs and are capable of supporting error-free transmission using CAT5e cable (“e” means “enhanced” for lower crosstalk between the conductors of the neighboring twisted pairs in the same cable). If you completely wire the connector, you will maximize the usefulness of your cables by allowing them to be used in many possible network configurations.

Table 2-2: 10/100 Base-T cable wire color code

Straight-through Cable (Both Ends)	Crossover Cable (One End)
1 – White/Orange	1 – White/Green
2 – Orange	2 – Green
3 – White/Green	3 – White/Orange
4 – Blue	4 – Blue
5 – White/Blue	5 – White/Blue
6 – Green	6 – Orange
7 – White/Brown	7 – White/Brown
8 – Brown	8 – Brown

Network Addresses

Each computer or node in a network really has two addresses. Actually, each interface in a computer or node has two addresses, as a single computer (or any network device) can have multiple network interfaces. These addresses are the **Internet Address** and the **Ethernet address**. The Ethernet address is a hardware address that identifies this specific network interface. The Internet address is a logical address that provides routing information so other computers on the network can find it.

Ethernet address

An Ethernet hardware address is also known as the Media Access Control (MAC) Address. This is a 12-digit hexadecimal (0-9, A-F) number that uniquely identifies your (and every other) Ethernet adapter. Manufacturers are assigned Ethernet address blocks to use, and they are used to identify each machine on the network. Ethernet addresses are usually shown as bytes separated by a colon or a dash, like this:

00:06:35:00:6B:BF or 00-06-35-00-6B-BF

The Ethernet address is always exactly 12 hexadecimal digits, so all leading zeroes are significant and must be entered. Since this address is a physical address it is

programmed into the network interface card and generally cannot be changed, while the Internet Address is a logical address and can be reassigned as needed. Also, there is no direct correlation between the Ethernet and Internet address and one cannot be calculated based on the other. Ethernet addresses are unique and never reused, unlike an Internet address that can be used on sub-networks and isolated networks. For the scope of this book it is not necessary to know more detail about the Ethernet address.

Internet addresses

The Internet Address is also called an **IP address** or **Host Address**. It is a logical address assigned to the network interface card in your computer. An IP address (where IP means Internet Protocol) is how one computer can find another computer on a network. Each node must know its own address on the network and that of any other computer with which it will communicate. The IP address is a 32-bit binary number that identifies each packet of information sent across the network. The IP address is usually expressed as four decimal numbers, each representing eight bits, separated by periods. This is sometimes known as the “dot address” or as “dotted quad notation.” For instance, in the example network shown in Figure 2-2, address 192.168.27.16 is the IP address of one of the machines. This is internally stored and used as the integer 3,232,242,448 which is 11000000.10101000.00011011.00010000. This can be represented in decimal numbers as shown:

1100 0000	.	1010 1000	.	0001 1011	.	0001 0000
192	.	168	.	27	.	16

Domain names

The numerical version of the IP address is usually represented by a name or series of names called the **domain name**—for instance, `www.someplace.com` or `ftp.filearchive.edu`, which is mapped into a static IP address using the **Domain Name System (DNS)**. The DNS is a hierarchical database used for translating the domain name to an IP address. When your computer needs to translate a domain name into a numerical IP address, it asks a domain name server to provide this information.

Network classes

The original Internet Protocol defines IP addresses in five major classes of address structure, Classes A through E. This has been named **classful routing** (probably in hindsight considering some of the improvements made to IP addressing that we will get to in a minute). Each of these classes allocates one portion of the 32-bit Internet address format to a network address and the remaining portion to the specific host machines within the network specified by the address. Class E is reserved for experimental use. Class D addresses are used for multicasting. Multicasting is data transmission between a single sender and multiple receivers on a network.

Class A networks use the 8 leftmost bits (the leftmost of the dotted quads) to designate the network number. The leftmost bit of these 8 bits is always 0, so Class A IP addresses range from 0.x.x.x to 127.x.x.x, except that address 0.x.x.x and 127.x.x.x are reserved for special use so this means there are 126 possible Class A networks. The rest of the dotted quads refer to the specific hosts or nodes on a large network. Since there are 24 bits in the remaining address, this means there can be 2^{24} possible hosts in each class A network, except that the all 0's case is reserved and means "this network" and the all 1's case is used for broadcasting, which leaves 16,777,214 possible hosts. So, a portion of the IP address represents the network number or address and a portion represents the local machine address.

Class B networks use the 16 leftmost bits (the leftmost two dotted quads) to designate the network number. The leftmost two of these 16 bits are always 10 so Class B addresses range from 128.0.x.x to 191.255.x.x, which means there are 16,384 possible Class B networks. The remaining two dotted quads (16 bits) refer to specific hosts or nodes on these networks. These 16 bits means there are 2^{16} possible hosts in each Class B network, except that the all 0's case is reserved and means "this network" and the all 1's case is used for broadcasting, which leaves 65,534 possible hosts. Using the above example, here's how the IP address is divided:

<u>155.185</u>	.	<u>127.162</u>
Network		Host
Address		Address

Class C networks use the 24 leftmost bits (the leftmost three dotted quads) to designate the network number. The leftmost three of these 24 bits are always 110 so Class C addresses range from 192.0.0.x to 223.255.255.x, which means there are 2,097,152 possible Class C networks. The remaining 8 bits (the right most dotted quad) refers to the specific hosts or nodes on each of these networks. These 8 bits means that there are 2^8 possible hosts in each class C network or 254 possible hosts (256 minus the all 0's and all 1's case as in the Class A and B networks). In the class C network the addresses from 192.168.0.0 to 192.168.255.0 are reserved for networks not directly connected to the Internet.

If you are really curious, you can look up the addresses in the different classes and see who they are assigned to on the "IP Network Index"¹ web page.

Not considering all of the reserved addresses in class A, B and C, there are 2^{23} or 4,294,967,296 possible IP addresses. While this might seem like a lot of addresses, many were not being used by the Class A address owners and with the rapid growth of broadband users and dedicated network devices, it turns out this is not enough addresses to support the future growth of the Internet. To overcome this limited

¹ IP Network Index, <http://ipindex.dragonstar.net/>

address space, a number of fixes have been devised, such as Subnetworking and Classless Inter-Domain Routing.

Subnetworks

A **subnetwork** is a logically separate portion of a larger network. It is a way of taking a single network address and splitting it so that a single network address can be used on several local networks. While not 100% technically accurate, it is kind of like splitting a class A or B network into smaller networks (or subnetworks). If you wanted to add a subnet to the sample address above, then some portion of the host address could be used for a subnet address. A company with a Class B address who needed more than 254 host machines, but far fewer than the 65,533 host addresses possible, would essentially be “wasting” most of the block of addresses allocated. That company could use several subnets and the remaining could be allocated to another company. A class A network address could be subnetted by allocating the second and maybe even the third dotted quad for a subnet address. A class B network address could be subnetted by allocating the third dotted quad for the subnet address as shown in this example:

<u>155.185</u>	.	<u>127</u>	.	<u>162</u>
Network		Subnet		Host
Address		Address		Address

To determine which part of the IP address was the host address and which was the subnet address, the IP address was paired with a subnet mask. This mask was used to separate the extended-network prefix (the network address and the subnet address) from the host address (logically ANDing the address and the subnet mask returns just the extended network prefix). With this technique, each of the quads of the subnet mask was either all binary 1's or all 0's (255 or 0).

IP address	155.185.127.162	10011011.10111001.11111111.10100010	
Subnet mask	255.255.255.0	11111111.11111111.11111111.00000000	
		Extended Network Prefix	Host
			Number

Classless Inter-Domain Routing (CIDR) was developed to effectively solve some of the problems with classful routing and some of the limitations of subnetting and extended network prefix by providing a new and more flexible way to specify network addresses in routers.

CIDR uses a variable-length subnet mask that does not necessarily have to be divided on any of the whole byte boundaries of the dotted quads. CIDR essentially eliminates classful routing (class A, B, D networks) by allowing the subnet mask to be any size. With CIDR a network address might look like this: