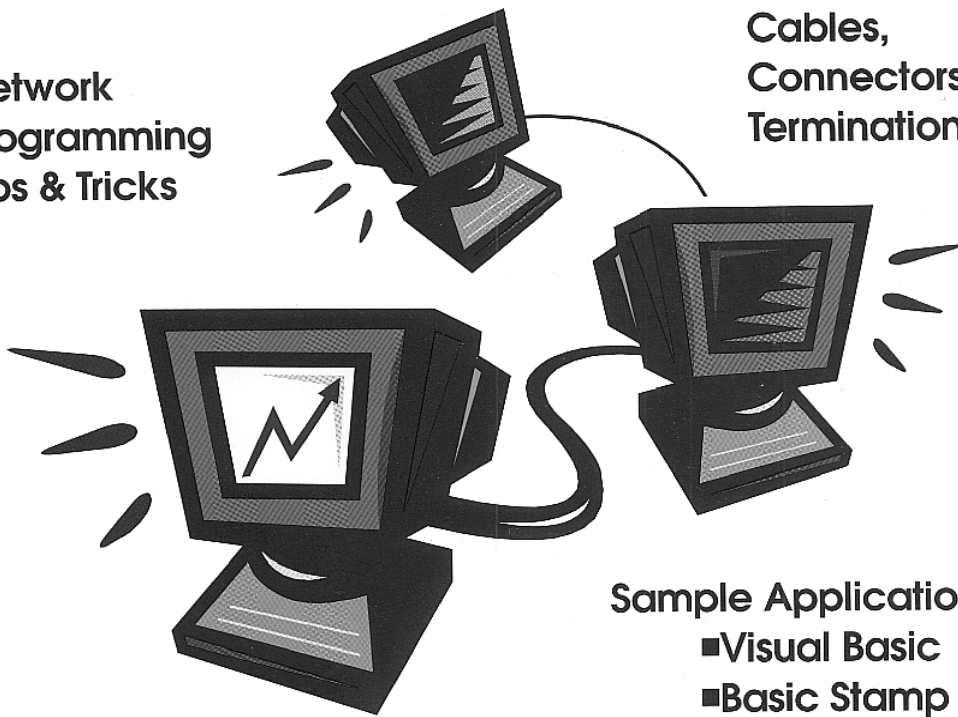


Jameco Part number 154771

# Serial Port Complete

*Programming and Circuits for  
RS-232 and RS-485  
Links and Networks*

Network  
Programming  
Tips & Tricks



Cables,  
Connectors, &  
Terminations

Sample Applications for

- Visual Basic
- Basic Stamp
- 8052-Basic



**INCLUDES  
DISK!**

## Jan Axelson

author of the best-selling  
*Parallel Port Complete*

## Table of Contents

	<b>Introduction</b>	<b>ix</b>
<b>1</b>	<b>Options and Choices</b>	<b>1</b>
	<b>The Computers</b>	<b>1</b>
	<b>The Programming</b>	<b>3</b>
	Languages and Operating Systems	
	Message Properties	
	<b>The Link</b>	<b>5</b>
	<b>Applications</b>	<b>7</b>
<b>2</b>	<b>Formats and Protocols</b>	<b>11</b>
	<b>Sending Serial Data</b>	<b>11</b>
	Synchronous Format	
	Asynchronous Format	
	System Support	
	<b>Transmitting a Byte</b>	<b>15</b>
	The Bit Format	
	Autodetecting the Bit Rate	
	<b>Data Formats</b>	<b>18</b>
	Binary Data	

	Text Data	
	<b>Preventing Missed Data 20</b>	
	Handshaking	
	Buffers	
	Polling and Interrupts	
	Acknowledgements	
	Error-checking	
<b>3</b>	<b>The PC's Serial Port from the Connector In Port Architecture 25</b>	<b>25</b>
	The UART	
	Enhancements	
	<b>Port Resources 28</b>	
	Finding Ports	
	Port Information in the Registry	
	<b>Configuring 31</b>	
	New Systems	
	Adding a Port	
	Using Older Hardware	
	Internal vs. External Devices	
	IRQ Conflicts	
	Solutions for Multiple Ports	
	<b>Inside the UART 40</b>	
	Interrupt Sources	
	Control Registers	
	New Functions	
<b>4</b>	<b>PC Programming Using MSComm 45</b>	<b>45</b>
	Properties	
	Text and Binary Transfers	
	Polled Communications	
	Using OnComm	
	Handshaking Options	
	<b>A Template Application 55</b>	
	The Main Form	
	General Routines	
	Selecting a Port	
	Saving Data	
	<b>Other Ways to Access Serial Ports 80</b>	
	API Functions	
	Direct Port Access	

Jameco Part number 154771

	<b>Using Older Basics 87</b>	
	Visual Basic Versions	
	Accessing Ports under DOS	
<b>5</b>	<b>Microcontroller Serial Ports 91</b>	
	<b>The 8051 Family 91</b>	
	The Serial Port	
	Interfacing Options	
	<b>The 8052-Basic 93</b>	
	Communications Abilities	
	Processing Received Data	
	Custom Communications	
	<b>The Basic Stamp 102</b>	
	A Firmware UART	
	Signal Levels	
	Stamp-to-Stamp Links	
	<b>Adding a Hardware Serial Port 113</b>	
	Options	
	An SPI/Microwire UART	
<b>6</b>	<b>Linking Two Devices with RS-232 117</b>	
	<b>About RS-232 118</b>	
	Features	
	Signals	
	Voltages	
	Timing Limits	
	<b>Converting between 5V Logic and RS-232 125</b>	
	The MAX232	
	Other Interface Chips	
	Short-range Circuits	
	<b>Port-powered Circuits 133</b>	
	<b>Alternate Interfaces 135</b>	
	Direct Connection	
	Other Unbalanced Interfaces	
<b>7</b>	<b>Connectors and Cables for RS-232 139</b>	
	<b>Connectors 139</b>	
	25-pin Shells	
	9-pin Shells	
	The Alt A Connector	
	Modular Connectors	
	Adapters	

	<b>Cables 146</b>	
	Length Limits	
	How Many Wires?	
	<b>Isolated Links 148</b>	
	Ways to Achieve Isolation	
	About Grounds	
	Power Supply Grounds	
	Optoisolating	
	Surge Protection	
	<b>Troubleshooting Tools 155</b>	
<b>8</b>	<b>RS-232 Applications</b>	<b>157</b>
	<b>Linking Two Computers 157</b>	
	A 2-PC Link	
	Selecting a Remote CPU	
	<b>PC-to-Basic Stamp Link 173</b>	
	Exchanging Data	
	Ensuring that the Stamp Sees Incoming Data	
	<b>PC-to-8052-Basic Link 177</b>	
	Ensuring that the 8052-Basic Sees Incoming Data	
	Exchanging Data	
	<b>Simple I/O 179</b>	
	Accessing the Signals	
	Connecting to a Stand-alone UART	
	Controlling Synchronous Interfaces	
	<b>Operating System Tools 181</b>	
	Direct Cable Connection	
	DOS Interlink and Intersvr	
<b>9</b>	<b>Links and Networks with RS-485</b>	<b>185</b>
	<b>About RS-485 185</b>	
	Balanced and Unbalanced Lines	
	Voltage Requirements	
	Current Requirements	
	Speed	
	<b>Adding an RS-485 Port 194</b>	
	PC Expansion Cards	
	Converter Chips	
	Converting TTL	
	Converting RS-232	
	Short Links between Different Interfaces	

<b>10</b>	<b>RS-485 Cables &amp; Interfacing</b>	<b>205</b>
	<b>Long and Short Lines 205</b>	
	When Is a Line Long?	
	Calculating Line Length	
	<b>Line Terminations 210</b>	
	Characteristic Impedance	
	Adding a Termination	
	Effects of Terminations	
	Reflections	
	Terminations for Short Lines	
	<b>Choosing a Driver Chip 222</b>	
	<b>Network Topologies 223</b>	
	<b>Open and Short-circuit Biasing 225</b>	
	Open-circuit Protection	
	Short-circuit Protection	
	<b>Cable Types 230</b>	
	How a Wire Picks Up Noise	
	Twisted-pair Cable	
	Selecting Cable	
	<b>Grounds in a Differential Link 233</b>	
	Ensuring a Common Ground	
	Isolated Links	
	<b>Extending a Link with Repeaters 239</b>	
<b>11</b>	<b>Network Programming</b>	<b>241</b>
	<b>Managing Traffic 241</b>	
	Steps in Exchanging a Message	
	Protocols	
	<b>Addressing 245</b>	
	Assigning	
	Detecting	
	Other Information in Messages	
	Using Existing Protocols	
	<b>Transmitter Enable Timing 249</b>	
<b>12</b>	<b>Two Networks</b>	<b>253</b>
	<b>An RS-485 Network 253</b>	
	The Protocol	
	The Link	
	The Master's Programming	
	Selecting Nodes	
	Slave Programming	

Jameco Part number 154771

**A Simple Stamp Network 259**  
**Debugging Tips 275**

**Appendices**

**A Resources 287**  
**B RS-232 Signals 291**  
**C Number Systems 293**

**Index 297**

## Introduction

This book is a guide to designing, programming, installing, and troubleshooting computer links, including networks of multiple computers. Most of the links described use one of two serial interfaces popularly known as RS-232 and RS-485. The computers may be personal computers, embedded controllers, or any devices that share a common interface. Common uses for these links include data acquisition and control systems.

### What's Inside

Putting together a serial link of this type requires expertise both in circuit design, to choose the components that make up the link, and in programming, to write the code that controls the link. This book covers each of these.

#### PCs

For links that use PCs (personal computers), I describe the architecture shared by the ports in all PCs, and show how to use a port's features, with program code in Visual Basic.

A template project gives a quick start for applications you write. The template includes routines for finding ports in a system, enabling users to select and config

ure a port, and other general-purpose functions commonly used in applications that access serial ports.

### **Embedded Controllers**

The examples for embedded controllers use one of two popular microcontroller chips: Parallax's Basic Stamp and the 80(C)52-Basic from Intel, Micromint, and other sources. Both are based on popular microcontroller hardware: the PIC and 8051 families. Both can be programmed in dialects of Basic.

### **Applications**

The example applications include circuit designs and code for serial links between two or more PCs and microcontrollers, using RS-232, RS-485, and simple direct links. I also show how to use an RS-232 serial port as the power source for low-power external circuits.

### **Cables and Interfacing**

In a serial link, the proper cable and terminating components can prevent errors due to electrical noise or malfunctioning hardware. This book shows how to choose an appropriate cable and how to select or design the circuits that interface to the cable.

### **Who should read this book?**

Readers from many backgrounds will find this book useful:

**Programmers** will find code examples that show how to use serial ports on PCs and microcontrollers. Programming a network shared by multiple computers is more involved than programming a link with just two computers. This book explains the options and shows how to avoid the pitfalls.

For **hardware designers**, there are details about serial-port circuits and how to interface to them, including the design of converters to translate between RS-232, RS-485, and 5V TTL logic. Examples show how to implement an RS-485 port on PCs and microcontrollers and how to design links with features like bullet-proof failsafe circuits, high noise immunity, and low power consumption.

For system **troubleshooters**, there are hardware and software debugging techniques, plus code for finding and testing ports and links.

**Hobbyists and experimenters** will find many ideas, along with explanations and tips for modifying the examples for a unique application.

Jameco Part number 154771

**Teachers and students** have found serial ports to be a handy tool for experiments with electronics and computer control and monitoring. Many of the examples in this book are suitable as a starting point for school projects.

And last but not least, users, or anyone who uses a computer with serial ports, will find useful information, including advice on configuring and adding ports.

### **How did you learn all this?**

Readers who have found my books useful sometimes ask how I came to know the information they contain. The short answer is that each of my books, including this one, is the result of a continuous cycle of research and experiments.

Many companies do an excellent job of providing information to help customers use their products. Manufacturers of the chips used in RS-232 and RS-485 interfaces have published dozens of application notes. The data sheets for the chips themselves also hold many answers. All of this information is now easily available on the Internet. There are also many good books that cover related topics. See the appendix of this book and Lakeview Research's web site for pointers to many good sources of information.

But in the end there's no substitute for real-life experiments: putting together the hardware, writing the program code, and watching what happens when the code executes. Then when the result isn't as expected--as it often isn't--it means trying something else or searching the documentation for clues. This book is result of many, many hours of such research and experiments.

### **About the Program Code**

The book's program disk includes all of the code presented in the book (and more). All of the code uses some variant of the Basic programming language.

I chose Basic for three reasons. First, much of this book is about trying things out and learning how they work, and Basic's interactive nature makes it a good choice for this. Second, Basic is popular, so many readers will already be familiar with it. And third, Basic is what I know best. Throughout, I've tried to document the code completely enough so that you can translate into another programming language if you wish.

For each example, I assume you have a basic understanding of the language used, whether it's Visual Basic or a microcontroller Basic, and how to create and debug programs. The book focuses on the details that relate specifically to serial communications.

Jameco Part number 154771

I developed the PC examples with Visual Basic 5. Because they're intended as design tools, and not as finished applications, the disk includes the complete source code but not compiled, executable programs. To compile the programs, you must have a copy of Visual Basic. I tested the code on a system running Windows 95. Visual Basic code is generally compatible with Windows NT as well, but I didn't test under NT.

I also used Visual Basic to illustrate other short calculations. These use Visual-Basic syntax, even when not presented explicitly as a code module. For example, an asterisk (\*) signifies multiplication. For those programming for Windows 3.x, the example code won't load directly into the older, 16-bit versions of Visual Basic, but you can copy and use many of the routines with few or no changes. In some cases, the program disk includes DOS Quick Basic code for use in systems using older or embedded PCs running DOS.

In a similar way, the microcontroller code examples are written for two popular chips, but the ideas behind the code are adaptable to other controllers and languages.

The Visual Basic code uses the line-continuation character (\\_) to enable a single line of code to extend over multiple lines. The routines within each listing are arranged alphabetically. The microcontroller code has no line-continuation character, and some program lines do carry over onto a second line in the listings in the text. The listings on disk are formatted correctly and should load and run without problems.

### **About the Example Circuits**

I've included many schematic diagrams of circuits that you can use or adapt in serial-port projects. In presenting the circuits, I assume you know the basics about digital logic and electronic circuits. The circuit diagrams are complete, with these exceptions:

Power-supply and ground pins are omitted when they are in standard locations on the package (bottom left for ground, top right for power, assuming pin 1 is top left).

Power-supply bypass capacitors are omitted.

Some chips may have additional, unused gates or other elements that aren't shown.

In the schematics and text, active-low signals use a leading hyphen (-RESET) or an overbar (RESET).

For more information on the components, see the manufacturers' data sheets.

## **Corrections and Updates**

In putting together this book, I've done my best to ensure that the information is complete and correct. Every schematic diagram with detail at the pinout level has been built and tested by me, most of it multiple times. In a similar way, I've run and tested every line of code. But I know from experience that on the way from test to publication, errors and omissions do occur.

Any corrections or updates to this book will be available at Lakeview Research's World Wide Web site on the Internet at <http://www.lvr.com>. This is also the place to come for links to other serial-port information on the Web, including software tools, component data sheets, and web sites of vendors of related products.

## **Thanks!**

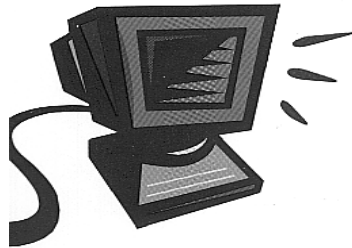
Finally, I want to say thanks to everyone who helped make this book possible. As with my previous *Parallel Port Complete*, I credit the readers of my articles in *The Microcomputer Journal*, who first prompted me to write about these topics. Some of the material in this book was first published in a different form in the magazine. Others deserving thanks are the folks at companies large and small who have generously provided information and answered questions about these topics over the years.

Jameco Part number 154771

*PC hardware & programming, electronics*

# Serial Port Complete

*Create robust, reliable links and networks with popular serial interfaces.*



RS-232 and RS-485 are versatile, inexpensive interfaces for linking computers, yet their hardware and programming requirements are often misunderstood. RS-232 can link two computers or a computer and peripheral, while its sister interface, RS-485, allows longer, faster links and networks of three or more computers.

In *Serial Port Complete*, Jan Axelson provides everything you need to design, program, install, and troubleshoot RS-232 and RS-485 links and networks:

Link PCs and embedded controllers, in any combination.

Select the interface, cable, and connectors that are right for your project.

Learn the secrets for error-free transmitting on RS-485 networks.

Use Visual Basic to access serial ports on PCs.

Design links using the serial ports on the Basic Stamp and 8052-Basic microcontrollers.

Develop projects quickly with circuit diagrams and program code for these applications: PC-to-PC link PC-to-embedded-controller link RS-485 network simple Stamp network

and much more!

## Praise for Jan's companion volume, *Parallel Port Complete*:

"A focused book that delivers what it promises: detailed technical information on the parallel port."  
Ron Burk, *Windows Developer's Journal*

"The author has done most of the work for you."  
Jon Titus, *Test & Measurement World*

"It's been a while since I've seen a book as practical as this one."  
Joseph Carr, *Nuts & Volts*

"An excellent resource."  
Paul Schreier,  
*Personal Engineering & Instrumentation News*

"Good stuff." Jeff Holtzman, *Electronics Now*

**Jan Axelson** is the author of *Parallel Port Complete*, *The Microcontroller Idea Book*, and *Making Printed Circuit Boards*. *Bill Machrone* of *PC Magazine* has described Jan's writing as "a model of clarity and conciseness."



Ⓢ39.95