



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China





RabbitCore RCM3900

C-Programmable Core Module
with *microSD*[™] Card Storage and Ethernet

User's Manual

019-0164_F

RabbitCore RCM3900 User's Manual

Part Number 019-0164_F • Printed in U.S.A.

©2008-2010 Digi International Inc. • All rights reserved.

Digi International reserves the right to make changes and improvements to its products without providing notice.

Trademarks

Rabbit and Dynamic C are registered trademarks of Digi International Inc.

Rabbit 3000 and RabbitCore are trademarks of Digi International Inc.

SD and microSD are trademarks of the SD Card Association.

The latest revision of this manual is available on the Rabbit Web site, www.rabbit.com, for free, unregistered download.

Rabbit Semiconductor Inc.

www.rabbit.com

TABLE OF CONTENTS

Chapter 1. Introduction	6
1.1 RCM3900 Features	6
1.2 Comparing the RCM3365/RCM3375 and the RCM3900/RCM3910	9
1.3 Advantages of the RCM3900	10
1.4 Development and Evaluation Tools.....	11
1.4.1 Development Kit	11
1.4.2 Software	12
1.4.3 Connectivity Interface Kits	12
1.4.4 Online Documentation	12
Chapter 2. Getting Started	13
2.1 Install Dynamic C	13
2.2 Hardware Connections.....	14
2.2.1 Step 1 — Attach Module to Prototyping Board.....	14
2.2.2 Step 2 — Connect Programming Cable.....	15
2.2.3 Connect Power	16
2.3 Starting Dynamic C	17
2.4 Run a Sample Program	17
2.4.1 Troubleshooting	17
2.5 Where Do I Go From Here?	18
2.5.1 Technical Support	18
Chapter 3. Running Sample Programs	19
3.1 Introduction.....	19
3.2 Sample Programs	20
3.2.1 Use of NAND Flash (RCM3900 only)	21
3.2.2 Use of <i>microSD</i> [™] Cards.....	23
3.2.3 Serial Communication.....	23
3.2.4 Real-Time Clock	25
3.2.5 Other Sample Programs	25
Chapter 4. Hardware Reference	26
4.1 RCM3900 Inputs and Outputs	27
4.1.1 Memory I/O Interface	32
4.1.2 LEDs	32
4.1.3 Other Inputs and Outputs	32
4.2 Serial Communication	33
4.2.1 Serial Ports	33
4.2.2 Ethernet Port	34
4.2.3 Serial Programming Port.....	35
4.3 Serial Programming Cable.....	36
4.3.1 Changing Between Program Mode and Run Mode	36
4.3.2 Standalone Operation of the RCM3900.....	37

4.4 Memory.....	38
4.4.1 SRAM	38
4.4.2 Flash EPROM	38
4.4.3 NAND Flash (RCM3900 only).....	38
4.4.4 <i>microSD</i> [™] Cards	39
4.5 Other Hardware.....	42
4.5.1 Clock Doubler	42
4.5.2 Spectrum Spreader	42
Chapter 5. Software Reference	43
5.1 More About Dynamic C	43
5.1.1 Developing Programs Remotely with Dynamic C.....	45
5.2 Dynamic C Functions	46
5.2.1 Digital I/O	46
5.2.2 SRAM Use	46
5.2.3 Serial Communication Drivers.....	47
5.2.4 TCP/IP Drivers.....	47
5.2.5 NAND Flash Drivers	47
5.2.6 <i>microSD</i> [™] Card Drivers.....	48
5.2.7 Prototyping Board Function Calls.....	49
5.2.7.1 Board Initialization.....	49
5.2.7.2 Digital I/O	50
5.2.7.3 Switches, LEDs, and Relay	52
5.2.7.4 Serial Communication.....	55
5.3 Upgrading Dynamic C	56
5.3.1 Extras	56
Chapter 6. Using the TCP/IP Features	57
6.1 TCP/IP Connections	57
6.2 TCP/IP Primer on IP Addresses.....	59
6.2.1 IP Addresses Explained	61
6.2.2 How IP Addresses are Used.....	62
6.2.3 Dynamically Assigned Internet Addresses	63
6.3 Placing Your Device on the Network	64
6.4 Running TCP/IP Sample Programs	65
6.4.1 How to Set IP Addresses in the Sample Programs	66
6.4.2 How to Set Up your Computer for Direct Connect	67
6.5 Run the PINGME.C Sample Program	68
6.6 Running Additional Sample Programs With Direct Connect.....	68
6.6.1 RabbitWeb Sample Programs	69
6.7 Where Do I Go From Here?	70
Appendix A. RCM3900 Specifications	71
A.1 Electrical and Mechanical Characteristics	72
A.1.1 Headers.....	76
A.2 Bus Loading	77
A.3 Rabbit 3000 DC Characteristics.....	80
A.4 I/O Buffer Sourcing and Sinking Limit	81
A.5 Conformal Coating.....	82
A.6 Jumper Configurations.....	83
Appendix B. Prototyping Board	85
B.1 Introduction.....	86
B.1.1 Prototyping Board Features.....	87
B.2 Mechanical Dimensions and Layout.....	89
B.3 Power Supply	91
B.4 Using the Prototyping Board.....	92

B.4.1 Adding Other Components.....	93
B.4.2 Digital I/O.....	94
B.4.2.1 Digital Inputs.....	94
B.4.3 CMOS Digital Outputs.....	95
B.4.4 Sinking Digital Outputs.....	95
B.4.5 Relay Outputs.....	95
B.4.6 Serial Communication.....	96
B.4.6.1 RS-232.....	97
B.4.6.2 RS-485.....	98
B.4.7 RabbitNet Port.....	99
B.4.8 Other Prototyping Board Modules.....	100
B.4.9 Quadrature Decoder.....	100
B.4.10 Stepper-Motor Control.....	100
B.5 Prototyping Board Jumper Configurations.....	102
B.6 Use of Rabbit 3000 Parallel Ports.....	104
Appendix C. LCD/Keypad Module	106
C.1 Specifications.....	106
C.2 Contrast Adjustments for All Boards.....	108
C.3 Keypad Labeling.....	109
C.4 Header Pinouts.....	110
C.4.1 I/O Address Assignments.....	110
C.5 Mounting LCD/Keypad Module on the Prototyping Board.....	111
C.6 Bezel-Mount Installation.....	112
C.6.1 Connect the LCD/Keypad Module to Your Prototyping Board.....	114
C.7 Sample Programs.....	115
C.8 LCD/Keypad Module Function Calls.....	116
C.8.1 LCD/Keypad Module Initialization.....	116
C.8.2 LEDs.....	117
C.8.3 LCD Display.....	118
C.8.4 Keypad.....	154
Appendix D. Power Supply	161
D.1 Power Supplies.....	161
D.1.1 Battery Backup.....	161
D.1.2 Battery-Backup Circuit.....	162
D.1.3 Reset Generator.....	163
Index	164
Schematics	168

1. INTRODUCTION

The RCM3900 RabbitCore modules feature a compact module that incorporates the latest revision of the powerful Rabbit[®] 3000 microprocessor, flash memory, onboard mass storage (NAND flash), static RAM, digital I/O ports, and removable (“hot-swappable”) memory cards. The RCM3900 RabbitCore modules both have an integrated 10/100Base-T Ethernet port, and provide for LAN and Internet-enabled systems to be built as easily as serial-communication systems.

A Development Kit provides the essentials that you need to design your own microprocessor-based system, and includes a complete Dynamic C software development system. The Development Kit also contains a Prototyping Board that will allow you to evaluate the RCM3900 module and to prototype circuits that interface to the module. You will also be able to write and test software for the RCM3900 modules.

Throughout this manual, the term RCM3900 refers to the complete series of RCM3900 RabbitCore modules unless other production models are referred to specifically.

The RCM3900 has a Rabbit 3000 microprocessor operating at 44.2 MHz, a fast program-execution SRAM, data SRAM, flash memory, two clocks (main oscillator and real-time clock), and the circuitry necessary for reset and management of battery backup of the Rabbit 3000’s internal real-time clock and the data SRAM. Two 34-pin headers bring out the Rabbit 3000 I/O bus lines, parallel ports, and serial ports.

The RCM3900 receives its +3.3 V power from the customer-supplied motherboard on which it is mounted. The RCM3900 can interface with all kinds of CMOS-compatible digital devices through the motherboard.

The RCM3900’s mass-storage capabilities make it suited to running the optional Dynamic C FAT file system module where data are stored and handled using the same directory file structure commonly used on PCs. A removable *microSD*[™] *Card* can be hot-swapped to transfer data quickly and easily using a standardized file system that can be read away from the RCM3900 installation.

1.1 RCM3900 Features

- Small size: 1.85" x 2.73" x 0.86"
(47 mm x 69 mm x 22 mm)
- Microprocessor: latest revision of Rabbit 3000 running at 44.2 MHz

- 10/100Base-T auto MDI/MDIX Ethernet port chooses Ethernet interface automatically based on whether a crossover cable or a straight-through cable is used in a particular setup
- 52 parallel 5 V tolerant I/O lines: 44 configurable for I/O, 4 fixed inputs, 4 fixed outputs
- Three additional digital inputs, two additional digital outputs
- External reset
- External I/O bus can be configured for 8 data lines and 5 address lines (shared with parallel I/O lines), I/O read/write
- Ten 8-bit timers (six cascadable) and one 10-bit timer with two match registers
- 512K flash memory, 512K program execution SRAM, 512K data SRAM
- Fixed and hot-swappable mass-storage memory options, which may be used with the standardized directory structure supported by the Dynamic C FAT File System module.
- Real-time clock
- Watchdog supervisor
- Provision for customer-supplied backup battery via connections on header J62
- 10-bit free-running PWM counter and four pulse-width registers
- Two-channel Input Capture (shared with parallel I/O ports) can be used to time input signals from various port pins
- Two-channel Quadrature Decoder accepts inputs from external incremental encoder modules
- Five or six 3.3 V CMOS-compatible serial ports with a maximum asynchronous baud rate of 5.525 Mbps. Three ports are configurable as a clocked serial port (SPI), and two ports are configurable as SDLC/HDLC serial ports (shared with parallel I/O ports).
- Supports 1.15 Mbps IrDA transceiver

There are two RCM3900 production models. Table 1 below summarizes their main features.

Table 1. RCM3900 Features

Feature	RCM3900	RCM3910
Microprocessor	Rabbit 3000 running at 44.2 MHz	
SRAM	512K program (fast SRAM) + 512K data	
Flash Memory (program)	512K	
Mass Data Storage	32MB (fixed NAND flash) + 128MB–1GB <i>microSD™ Card</i>	128MB–1GB <i>microSD™ Card</i>
Serial Ports	6 shared high-speed, 3.3 V CMOS-compatible ports: <ul style="list-style-type: none"> • all 6 are configurable as asynchronous serial ports; • 4 are configurable as a clocked serial port (SPI) and 1 is configurable as an HDLC serial port; • option for second HDLC serial port at the expense of 2 clocked serial ports (SPI) 	

The RCM3900 is programmed over a standard PC serial port through a USB programming cable supplied with the Development Kit, and can also be programmed directly over an Ethernet link using the Dynamic C download manager with or without a RabbitLink.

Appendix A provides detailed specifications for the RCM3900.

1.2 Comparing the RCM3365/RCM3375 and the RCM3900/RCM3910

- **Temperature Specifications** — We can no longer obtain certain components for the RCM3365/RCM3375 RabbitCore modules that support the -40°C to +70°C temperature range. RCM3365/RCM3375 RabbitCore modules manufactured after May, 2008, are specified to operate at 0°C to +70°C. The RCM3900/RCM3910, rated for -20°C to +85°C, are available after May, 2008.
- **Removable Mass Storage** — The hot-swappable *xD-Picture Card*[™] mass storage device with up to 128MB of memory has been replaced with the *SD Card* with up to 1GB of memory. The *SD Card* is more readily available today, and is expected to remain readily available for a long time. In addition, *SD Cards* provide a significantly larger memory capacity, which has been requested by customers. The trade-off for the larger memory capacity is that the data transfer rate to/from the *SD Card* is about an order of magnitude slower than to/from the *xD-Picture Card*.

NOTE: RCM3365/RCM3375 RabbitCore modules may eventually be discontinued because of changes to the *xD-Picture Card*[™]. The *miniSD*[™] *Card* initially used with the RCM3900 series has since been replaced by the *microSD*[™] *Card*. Aside from using a different sized memory card, the boards function the same.

- **Serial Ports** — Serial Port B, available as either a clocked serial port or an asynchronous serial port on the RCM3365/RCM3375, is used by the RCM3900/RCM3910 as a clocked serial peripheral interface (SPI) for the *microSD*[™] *Card*, and is not brought out for customer use.
- **General-Purpose I/O** — PD2, a configurable I/O pin on the RCM3365/RCM3375, is used to detect whether the *microSD*[™] *Card* is installed on the RCM3900/RCM3910, and so PD2 is not brought out for customer use on the RCM3900/RCM3910.
- **Maximum Current** — The RCM3365/RCM3375 draws 250 mA vs. the 325 mA required by the RCM3900/RCM3910.
- **LEDs** — The **SPEED** and user (**USR/BSY**)LED locations have been swapped between the RCM3365/RCM3375 and the RCM3900/RCM3910, the **LNK/ACT** LEDs have been combined to one LED on the RCM3900/RCM3910, and the RCM3900/RCM3910 has an **FDX/COL** LED instead of the **FM** LED on the RCM3365/RCM3375. The LED placements on the boards remain unchanged.
- **Ethernet chip** — A different Ethernet controller chip is used on the RCM3900. The Ethernet chip is able to detect automatically whether a crossover cable or a straight-through cable is being used in a particular setup, and will configure the signals on the Ethernet jack interface.
- **Dynamic C** — As long as no low-level FAT file system calls or direct *xD-Picture Card* access calls to the **NFLASH.LIB** library were used in your application developed for the RCM3365/RCM3375, you may run that application on the RCM3900/RCM3910 after you recompile it using Dynamic C v. 9.62.

NOTE: The Dynamic C RabbitSys option for programming an RCM3365 over an Ethernet link is not supported for the RCM3900.

1.3 Advantages of the RCM3900

- Fast time to market using a fully engineered, “ready-to-run/ready-to-program” micro-processor core.
- Competitive pricing when compared with the alternative of purchasing and assembling individual components.
- Easy C-language program development and debugging
- Program download utility (Rabbit Field Utility) and cloning board options for rapid production loading of programs.
- Generous memory size allows large programs with tens of thousands of lines of code, and substantial data storage.
- Integrated Ethernet port for network connectivity, with royalty-free TCP/IP software.
- Ideal for network-enabling security and access systems, home automation, HVAC systems, and industrial controls.

1.4 Development and Evaluation Tools

1.4.1 Development Kit

The Development Kit contains the hardware and software needed to use the RCM3900.

- RCM3900 module.
- Prototyping Board.
- 1 GB *microSD*[™] Card with SD Card adapter.
- Universal AC adapter, 12 V DC, 1 A (includes Canada/Japan/U.S., Australia/N.Z., U.K., and European style plugs).
- USB programming cable with 10-pin header.
- *Dynamic C* CD-ROM, with complete product documentation on disk.
- *Getting Started* instructions.
- Accessory parts for use on the Prototyping Board.
- Screwdriver and Cat. 5 Ethernet cables.
- *Rabbit 3000 Processor Easy Reference* poster.
- Registration card.

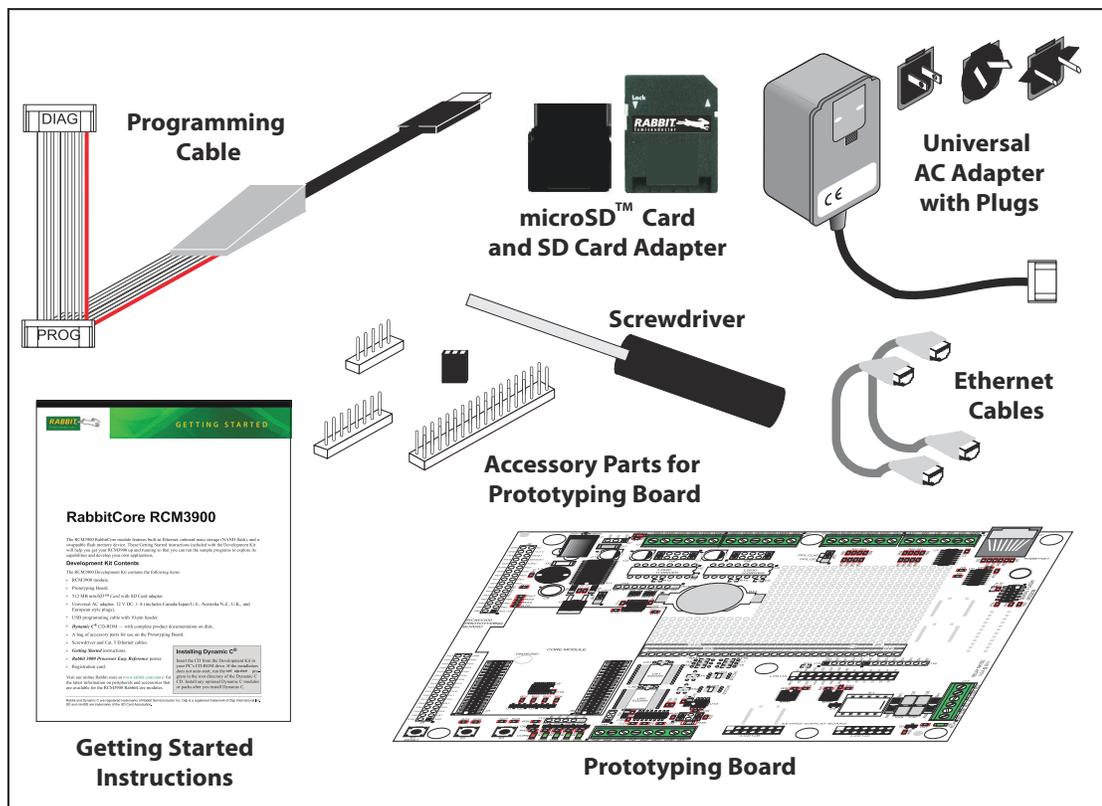


Figure 1. RCM3900 Development Kit

1.4.2 Software

The RCM3900 is programmed using version 9.62 of Dynamic C. A compatible version is included on the Development Kit CD-ROM. This version of Dynamic C includes the popular μ C/OS-II real-time operating system, point-to-point protocol (PPP), FAT file system, RabbitWeb, and other select libraries.

Rabbit also offers the Rabbit Embedded Security Pack featuring the Secure Sockets Layer (SSL) and a specific Advanced Encryption Standard (AES) library. In addition to the Web-based technical support included at no extra charge, a one-year telephone-based technical support subscription is also available for purchase. Visit our Web site at www.rabbit.com for further information and complete documentation, or contact your Rabbit sales representative or authorized distributor.

1.4.3 Connectivity Interface Kits

Rabbit has available a Connector Adapter Board to allow you to use the RCM3900 with header sockets that have a 0.1" pitch.

- Connector Adapter Board (Part No. 151-0114)—allows you to plug the RCM3900 whose headers have a 2 mm pitch into header sockets with a 0.1" pitch.

Visit our Web site at www.rabbit.com or contact your Rabbit sales representative or authorized distributor for further information.

1.4.4 Online Documentation

The online documentation is installed along with Dynamic C, and an icon for the documentation menu is placed on the workstation's desktop. Double-click this icon to reach the menu. If the icon is missing, use your browser to find and load **default.htm** in the **docs** folder, found in the Dynamic C installation folder.

Each Dynamic C module has complete documentation available with the online documentation described above.

The latest versions of all documents are always available for free, unregistered download from our Web sites as well.

2. GETTING STARTED

This chapter describes the RCM3900 hardware in more detail, and explains how to set up and use the accompanying Prototyping Board.

NOTE: It is assumed that you have the RCM3900 Development Kit. If you purchased an RCM3900 module by itself, you will have to adapt the information in this chapter and elsewhere to your test and development setup.

2.1 Install Dynamic C

To develop and debug programs for the RCM3900 (and for all other Rabbit hardware), you must install and use Dynamic C.

If you have not yet installed Dynamic C, do so now by inserting the Dynamic C CD from the RCM3900 Development Kit in your PC's CD-ROM drive. If autorun is enabled, the CD installation will begin automatically.

If autorun is disabled or the installation otherwise does not start, use the Windows **Start | Run** menu or Windows Explorer to launch **setup.exe** from the root folder of the CD-ROM.

The installation program will guide you through the installation process. Most steps of the process are self-explanatory.

Dynamic C uses a COM (serial) port to communicate with the target development system. The installation allows you to choose the COM port that will be used. The default selection is COM1. Select any available USB port for Dynamic C's use. This selection can be changed later within Dynamic C.

NOTE: The installation utility does not check the selected COM port in any way. Specifying a port in use by another device (mouse, modem, etc.) may lead to a message such as "**could not open serial port**" when Dynamic C is started.

Once your installation is complete, you will have up to three icons on your PC desktop. One icon is for Dynamic C, one opens the documentation menu, and the third is for the Rabbit Field Utility, a tool used to download precompiled software to a target system.

If you plan to use the optional Dynamic C Rabbit Embedded Security Pack, install it after installing Dynamic C. You must install the Rabbit Embedded Security Pack in the same directory where Dynamic C was installed.

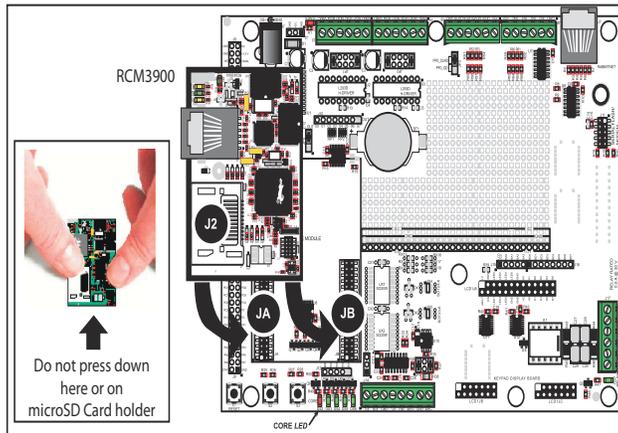
2.2 Hardware Connections

There are three steps to setting up the Prototyping Board:

1. Attach the RCM3900 module to the Prototyping Board.
2. Connect the serial programming cable between the RCM3900 and the workstation PC.
3. Connect the power supply to the Prototyping Board.

2.2.1 Step 1 — Attach Module to Prototyping Board

Turn the RCM3900 module so that the Ethernet jack is facing the direction shown in Figure 2 below. Align the pins from headers J61 and J62 on the bottom side of the module into header sockets JA and JB on the Prototyping Board. The *microSD™ Card* does not have to be inserted into connector J2 on the RCM3900 at this time—there is a protective spacer insert that you simply pull out before inserting a *microSD™ Card* for the first time.



CAUTION: You will sense a soft click once you insert the *microSD™ Card* completely. To remove it, gently press the card towards the middle of the RCM3900 — you will sense a soft click and the card will be ready to be removed. Do *not* attempt to pull the card from the socket before pressing it in — otherwise the ejection mechanism will get damaged. The ejection mechanism is spring-loaded, and will partially eject the card when used correctly.

Figure 2. Install the RCM3900 Series on the Prototyping Board

NOTE: It is important that you line up the pins on headers J61 and J62 of the RCM3900 module exactly with the corresponding pins of header sockets JA and JB on the Prototyping Board. The header pins may become bent or damaged if the pin alignment is offset, and the module will not work. Permanent electrical damage to the module may also result if a misaligned module is powered up.

Press the module's pins firmly into the Prototyping Board header sockets—press down in the area above the header pins using your thumbs or fingers over the header pins as shown in Figure 2. Do not press down on the *microSD™ Card* connector (J2) even if the *microSD™ Card* is installed, but rather press down on the circuit board along the edge by the connector. Also, do not press down on the middle of the module to avoid flexing the module, which could damage the module or components on the module.

Should you need to remove the module, grasp it with your fingers along the sides by the connectors and gently work the module up to pull the pins away from the sockets where they are installed. Do not remove the module by grasping it at the top and bottom.

2.2.2 Step 2 — Connect Programming Cable

The programming cable connects the RCM3900 to the PC running Dynamic C to download programs and to monitor the RCM3900 module during debugging.

Connect the 10-pin connector of the programming cable labeled **PROG** to header J1 on the RCM3900 as shown in Figure 3. There is a small dot on the circuit board next to pin 1 of header J1. Be sure to orient the marked (usually red) edge of the cable towards pin 1 of the connector. (Do not use the **DIAG** connector, which is used for a nonprogramming serial connection.)

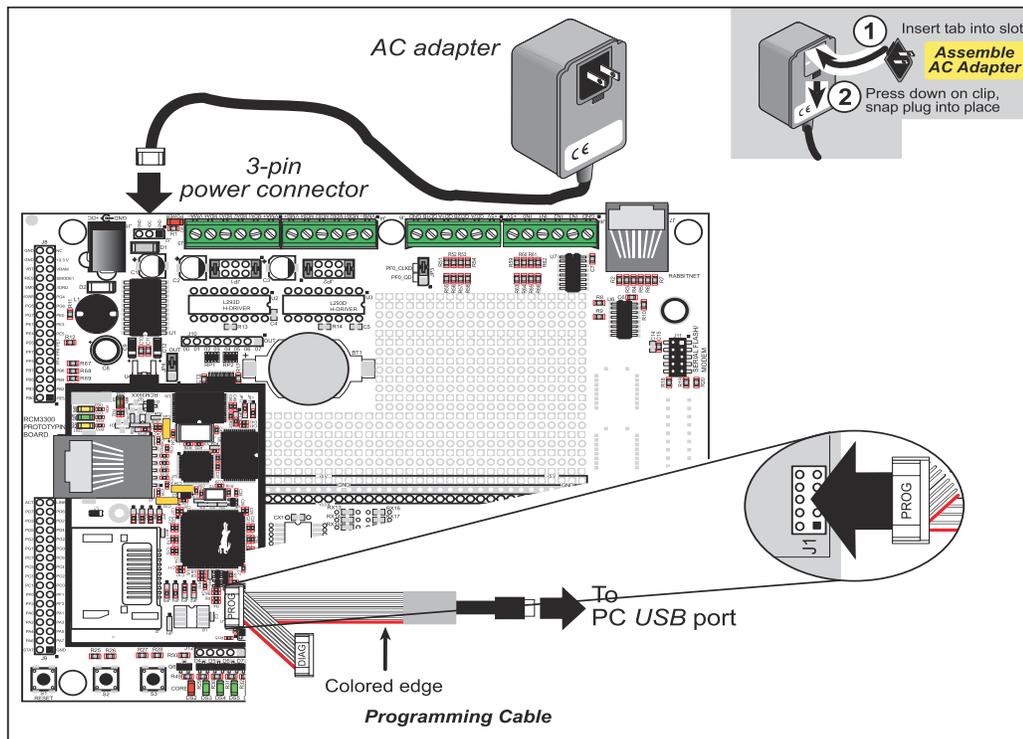


Figure 3. Connect Programming Cable and Power Supply

Connect the other end of the programming cable to an available USB port on your PC or workstation. Your PC should recognize the new USB hardware, and the LEDs in the shrink-wrapped area of the USB programming cable will flash.

2.2.3 Connect Power

When all other connections have been made, you can connect power to the Prototyping Board.

First, prepare the AC adapter for the country where it will be used by selecting the plug. The RCM3900 Development Kit presently includes Canada/Japan/U.S., Australia/N.Z., U.K., and European style plugs. Snap in the top of the plug assembly into the slot at the top of the AC adapter as shown in Figure 3, then press down on the spring-loaded clip below the plug assembly to allow the plug assembly to click into place.

Connect the AC adapter to 3-pin header J2 on the Prototyping Board as shown in Figure 3.

Plug in the AC adapter. The red **CORE** LED on the Prototyping Board should light up. The RCM3900 and the Prototyping Board are now ready to be used.

NOTE: A **RESET** button is provided on the Prototyping Board to allow a hardware reset without disconnecting power.

2.3 Starting Dynamic C

Once the RCM3900 is connected as described in the preceding pages, start Dynamic C by double-clicking on the Dynamic C icon on your desktop or in your **Start** menu. Dynamic C uses the serial port specified during installation. Select **Code and BIOS in Flash, Run in RAM** on the “Compiler” tab in the Dynamic C **Options > Project Options** menu. Then click on the “Communications” tab and verify that **Use USB to Serial Converter** is selected to support the USB programming cable. Click **OK**.

This program shows that the CPU is working. The sample program described in Section 6.5, “Run the PINGME.C Sample Program,” tests the TCP/IP portion of the board.

2.4 Run a Sample Program

Use the **File** menu to open the sample program **PONG.C**, which is in the Dynamic C **SAMPLES** folder. Press function key **F9** to compile and run the program. The **STDIO** window will open on your PC and will display a small square bouncing around in a box.

2.4.1 Troubleshooting

If Dynamic C cannot find the target system (error message "**No Rabbit Processor Detected.** "):

- Check that the RCM3900 is powered correctly — the red **CORE** LED on the Prototyping Board should be lit when the RCM3900 is mounted on the Prototyping Board and the AC adapter is plugged in.
- Check both ends of the programming cable to ensure that they are firmly plugged into the PC and the **PROG** connector, not the **DIAG** connector, is plugged in to the programming port on the RCM3900 with the marked (colored) edge of the programming cable towards pin 1 of the programming header.
- Ensure that the RCM3900 module is firmly and correctly installed in its connectors on the Prototyping Board.
- Select a different COM port within Dynamic C. From the **Options** menu, select **Project Options**, then select another COM port from the list on the **Communications** tab, then click **OK**. Press **<Ctrl-Y>** to force Dynamic C to recompile the BIOS.
- If you get an error message when you plugged the programming cable into a USB port, you will have to install USB drivers. Drivers for Windows XP are available in the Dynamic C **Drivers\Rabbit USB Programming Cable\WinXP_2K** folder — double-click **DPInst.exe** to install the USB drivers. Drivers for other operating systems are available online at www.ftdichip.com/Drivers/VCP.htm.

If Dynamic C appears to compile the BIOS successfully, but you then receive a communication error message when you compile and load a sample program, it is possible that your PC cannot handle the higher program-loading baud rate. Try changing the maximum download rate to a slower baud rate as follows.

- Locate the **Serial Options** dialog on the “Communications” tab in the Dynamic C **Options > Project Options** menu. Select a slower Max download baud rate. Click **OK** to save.

If a program compiles and loads, but then loses target communication before you can begin debugging, it is possible that your PC cannot handle the default debugging baud rate. Try lowering the debugging baud rate as follows.

- Locate the **Serial Options** dialog in the Dynamic C **Options > Project Options > Communications** menu. Choose a lower debug baud rate. Click **OK** to save.

Press **<Ctrl-Y>** to force Dynamic C to recompile the BIOS. The LEDs on the USB programming cable will blink and you should receive a **Bios compiled successfully** message.

2.5 Where Do I Go From Here?

If the sample program ran fine, you are now ready to go on to other sample programs and to develop your own applications. The source code for the sample programs is provided to allow you to modify them for your own use. The *RCM3900 User's Manual* also provides complete hardware reference information and describes the software function calls for the RCM3900, the Prototyping Board, and the optional LCD/keypad module.

For advanced development topics, refer to the *Dynamic C User's Manual* and the *Dynamic C TCP/IP User's Manual*, also in the online documentation set.

2.5.1 Technical Support

NOTE: If you purchased your RCM3900 through a distributor or through a Rabbit partner, contact the distributor or partner first for technical support.

If there are any problems at this point:

- Use the Dynamic C **Help** menu to get further assistance with Dynamic C.
- Check the Rabbit Technical Bulletin Board and forums at www.rabbit.com/support/bb/ and at www.rabbit.com/forums/.
- Use the Technical Support e-mail form at www.rabbit.com/support/.

3. RUNNING SAMPLE PROGRAMS

To develop and debug programs for the RCM3900 (and for all other Rabbit hardware), you must install and use Dynamic C.

3.1 Introduction

To help familiarize you with the RCM3900 modules, Dynamic C includes several sample programs. Loading, executing and studying these programs will give you a solid hands-on overview of the RCM3900's capabilities, as well as a quick start with Dynamic C as an application development tool.

NOTE: The sample programs assume that you have at least an elementary grasp of the C programming language. If you do not, see the introductory pages of the *Dynamic C User's Manual* for a suggested reading list.

In order to run the sample programs discussed in this chapter and elsewhere in this manual,

1. Your RCM3900 must be plugged in to the Prototyping Board as described in Chapter 2, "Getting Started."
2. Dynamic C must be installed and running on your PC.
3. The programming cable must connect the programming header (J1) on the RCM3900 to your PC.
4. Power must be applied to the RCM3900 through the Prototyping Board.

Refer to Chapter 2, "Getting Started," if you need further information on these steps.

Since the RCM3900 runs at 44.2 MHz and is equipped with a fast program execution SRAM, remember to allow the compiler to run the application in the fast program execution SRAM by selecting **Code and BIOS in Flash, Run in RAM** from the Dynamic C **Options > Project Options > Compiler** menu.

To run a sample program, open it with the **File** menu, then compile and run it by pressing **F9**.

Complete information on Dynamic C is provided in the *Dynamic C User's Manual*.

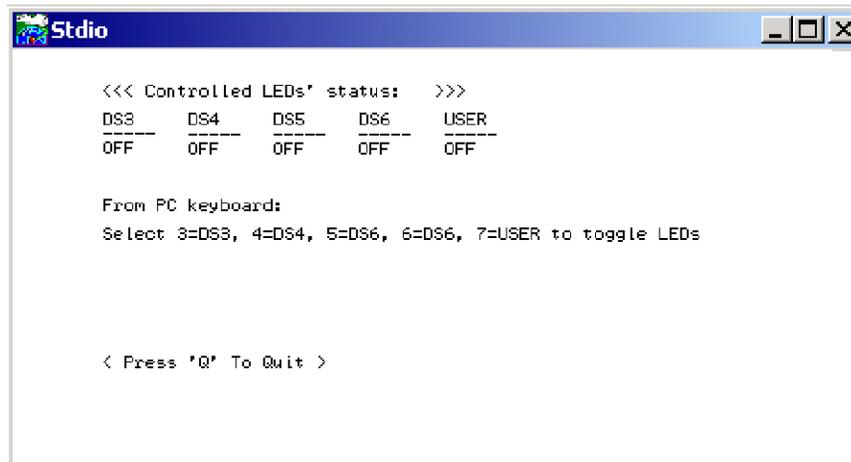
3.2 Sample Programs

Of the many sample programs included with Dynamic C, several are specific to the RCM3900. Sample programs illustrating the general operation of the RCM3900, serial communication, and the NAND flash are provided in the **SAMPLES\RCM3900** folder. Each sample program has comments that describe the purpose and function of the program. Follow the instructions at the beginning of the sample program.

TCP/IP sample programs are described in Chapter 6, “Using the TCP/IP Features.” Sample programs for the optional LCD/keypad module that is used on the RCM3900 Prototyping Board are described in Appendix C.

- **CONTROLLEDS .c**—Demonstrates use of the digital outputs by having you turn the LEDs on the Prototyping Board on or off from the **STDIO** window on your PC.

Once you compile and run **CONTROLLEDS .c**, the following display will appear in the Dynamic C **STDIO** window.



```
Stdio
<<< Controlled LEDs' status: >>>
DS3   DS4   DS5   DS6   USER
----  ----  ----  ----  ----
OFF   OFF   OFF   OFF   OFF

From PC keyboards:
Select 3=DS3, 4=DS4, 5=DS6, 6=DS6, 7=USER to toggle LEDs

< Press 'Q' To Quit >
```

Press “3” or “4” or “5” or “6” or “7” on your keyboard to select LED DS3 or DS4 or DS5 or DS6 on the Prototyping Board or the **USR** LED on the RCM3900. Then follow the prompt in the Dynamic C **STDIO** window to turn the LED on or off.

- **FLASHLEDS .c**—Demonstrates assembly-language program by flashing the **USR** LED on the RCM3900 and LEDs DS3, DS4, DS5, and DS6 on the Prototyping Board.
- **SWRELAY .c**—Demonstrates the relay-switching function call using the relay installed on the Prototyping Board by toggling the relay output state via the Dynamic C **STDIO** window.
- **TOGGLESWITCH .c**—Uses costatements (cooperative multitasking) to detect switches S2 and S3 using debouncing. The corresponding LEDs (DS3 and DS4) will turn on or off.

Once you have loaded and executed these four programs and have an understanding of how Dynamic C and the RCM3900 modules interact, you can move on and try the other sample programs, or begin building your own.

3.2.1 Use of NAND Flash (RCM3900 only)

The following sample programs can be found in the `SAMPLES\RCM3900\NANDFlash` folder.

NOTE: These sample programs cannot be run on the RCM3910, which does not have NAND flash installed.

- **NFLASH_DUMP.C**—This program is a utility for dumping the nonerased contents of a NAND flash chip to the Dynamic C **STDIO** window, and the contents may be redirected to a serial port.

When the sample program starts running, it attempts to communicate with the NAND flash chip. If this communication is successful and the main page size is acceptable, the nonerased page contents (non 0xFF) from the NAND flash page are dumped to the Dynamic C **STDIO** window.

Note that an error message might appear when the first 32 pages (0x20 pages) are “dumped.” You may ignore the error message.

- **NFLASH_ERASE.C**—This program is a utility for erasing all the good blocks of a NAND flash device.

When the sample program starts running, it attempts to initialize the onboard NAND flash chip. If this initialization is successful, the progress of the blocks being erased is reported in the Dynamic C **STDIO** window.

- **NFLASH_INSPECT.C**—This program is a utility for inspecting the contents of a NAND flash chip. When the sample program starts running, it attempts to initialize the onboard NAND flash chip. If this initialization is successful, the user can execute various commands to print out the contents of a specified page, clear (set to zero) all the bytes in a specified page, erase (set to FF), or write a specified value or count pattern to specified pages.

When you run this sample program, setting the **NFLASH_USEERASEBLOCKSIZE** macro to (0) zero makes the NAND flash driver use smaller (512-byte) chunks of data, which are less tedious to manage in this program than the alternative larger (16K) chunks of data. However, using smaller chunks of data means more NAND flash block erases are required to update all the program pages in an erased block, one per program page written. In contrast, updating all the program pages in an erased block can require only a single block erase when all of the program pages within the erase block are treated as a single large page. See the `nf_initDevice()` function help for more information.

- **NFLASH_LOG.C**—This program runs a simple Web server that stores a log of hits in the NAND flash. This log can be viewed and cleared from a Web browser by connecting the RJ-45 jack on the RCM3900 to your PC as described in Section 6.1. The sidebar on the next page explains how to set up your PC or notebook to view this log.

Before you compile and run this sample program, make any configuration changes to match your requirements. Once you are viewing the results in a Web browser, click the [“Source code”](#) link on the Web page to see this sample program’s `#ximported` file content.

Note that this sample program does not use the optimum method of writing to the NAND flash. The inefficiency resulting from the small amount of data written in each append operation is offset somewhat by the expected relative infrequency of these writes, and by the sample program's method of "walking" through the flash blocks when appending data as well as when a log is cleared.

There is little difference in the number of NAND flash block erase operations regardless of whether the `NFLASH_USEERASEBLOCKSIZE` macro is 0 (zero) or 1 (one). It is slightly more efficient to have the NAND flash driver use larger (16K) chunks of data. See the `nf_initDevice()` function help for more information.

Follow these instructions to set up your PC or notebook. Check with your administrator if you are unable to change the settings as described here since you may need administrator privileges. The instructions are specifically for Windows 2000, but the interface is similar for other versions of Windows.

TIP: If you are using a PC that is already on a network, you will disconnect the PC from that network to run these sample programs. Write down the existing settings before changing them to facilitate restoring them when you are finished with the sample programs and reconnect your PC to the network.

1. Go to the control panel (**Start > Settings > Control Panel**), and then double-click the Network icon.
2. Select the network interface card used for the Ethernet interface you intend to use (e.g., **TCP/IP Xircom Credit Card Network Adapter**) and click on the "Properties" button. Depending on which version of Windows your PC is running, you may have to select the "Local Area Connection" first, and then click on the "Properties" button to bring up the Ethernet interface dialog. Then "Configure" your interface card for a "10Base-T Half-Duplex" or an "Auto-Negotiation" connection on the "Advanced" tab.

NOTE: Your network interface card will likely have a different name.

3. Now select the **IP Address** tab, and check **Specify an IP Address**, or select TCP/IP and click on "Properties" to assign an IP address to your computer (this will disable "obtain an IP address automatically"):

IP Address : 10.10.6.101

Netmask : 255.255.255.0

Default gateway : 10.10.6.1

4. Click **<OK>** or **<Close>** to exit the various dialog boxes.

As long as you have not modified the `TCPCONFIG 1` macro in the sample program, enter the following server address in your Web browser to bring up the Web page served by the sample program.

`http://10.10.6.100`

Otherwise use the TCP/IP settings you entered in the `LIB\TCPIP\TCP_CONFIG.LIB` library.

3.2.2 Use of *microSD*[™] Cards

The following sample program can be found in the **SAMPLES\RCM3900\SD_Flash** folder.

- **SDFLASH_INSPECT.C**—This program is a utility for inspecting the contents of a *microSD*[™] Card. When the sample program starts running, it attempts to initialize the *microSD*[™] Card on Serial Port B. The following five commands are displayed in the Dynamic C **STDIO** window if a *microSD*[™] Card is found:

- p — print out the contents of a specified page on the *microSD*[™] Card
- r — print out the contents of a range of pages on the *microSD*[™] Card
- c — clear (set to zero) all of the bytes in a specified page
- f — sets all bytes on the specified page to the given value
- t — write user-specified text to a selected page

The sample program prints out a single line for a page if all bytes in the page are set to the same value. Otherwise it prints a hex/ASCII dump of the page.

This utility works with the *microSD*[™] Card at its lowest level, and writing to pages will likely make the *microSD*[™] Card unreadable by a PC. For PC compatibility, you must use the Dynamic C FAT file system module, which allows you to work with files on the *microSD*[™] Card in a way that they will be PC-compatible.

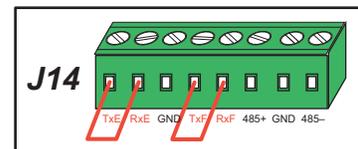
- **SDFLASH_LOG.C**—This program demonstrates a simple Web server that stores a log of hits in the *microSD*[™] Card's data flash. This log can be viewed and cleared from a Web browser—see the **NFLASH_LOG.C** sample program for information on how to access the Web page and configure your PC or workstation.

3.2.3 Serial Communication

The following sample programs can be found in the **SAMPLES\RCM3900\SERIAL** folder.

- **FLOWCONTROL.C**—This program demonstrates hardware flow control by configuring Serial Port F for CTS/RTS with serial data coming from TxE (Serial Port E) at 115,200 bps. One character at a time is received and is displayed in the **STDIO** window.

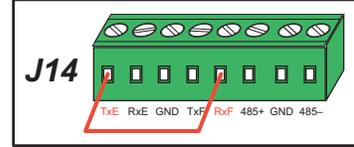
To set up the Prototyping Board, you will need to tie TxE and RxE together on the RS-232 header at J14, and you will also tie TxF and RxF together as shown in the diagram.



A repeating triangular pattern should print out in the **STDIO** window. The program periodically switches RTS (TxF) flow control on or off to demonstrate the effect of hardware flow control. You may use an oscilloscope to observe the CTS/RTS signals to see flow control operating

- **PARITY.C**—This program demonstrates the use of parity modes by repeatedly sending byte values 0–127 from Serial Port E to Serial Port F. The program will switch between generating parity or not on Serial Port E. Serial Port F will always be checking parity, so parity errors should occur during every other sequence.

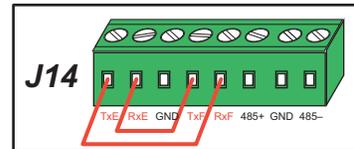
To set up the Prototyping Board, you will need to tie TxE and RxF together on the RS-232 header at J14 as shown in the diagram.



The Dynamic C **STDIO** window will display the error sequence.

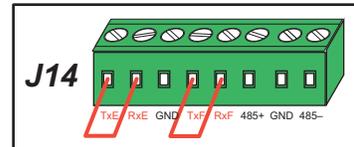
- **SIMPLE3WIRE.C**—This program demonstrates basic RS-232 serial communication. Lower case characters are sent by TxE, and are received by RxF. The characters are converted to upper case and are sent out by TxF, are received by RxE, and are displayed in the Dynamic C **STDIO** window.

To set up the Prototyping Board, you will need to tie TxE and RxF together on the RS-232 header at J14, and you will also tie RxE and TxF together as shown in the diagram.



- **SIMPLE5WIRE.C**—This program demonstrates 5-wire RS-232 serial communication by providing flow control (RTS/CTS) on Serial Port F and data flow on Serial Port E.

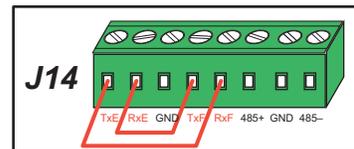
To set up the Prototyping Board, you will need to tie TxE and RxE together on the RS-232 header at J14, and you will also tie TxF and RxF together as shown in the diagram.



Once you have compiled and run this program, you can test flow control by disconnecting TxF from RxF while the program is running. Characters will no longer appear in the **STDIO** window, and will display again once TxF is connected back to RxF. (Do not disconnect the data path between TxE and RxE.)

- **SWITCHCHAR.C**—This program transmits and then receives an ASCII string on Serial Ports E and F. It also displays the serial data received from both ports in the **STDIO** window.

To set up the Prototyping Board, you will need to tie TxE and RxF together on the RS-232 header at J14, and you will also tie RxE and TxF together as shown in the diagram.



Once you have compiled and run this program, press and release S2 on the Prototyping Board to send a message from Serial Port E to Serial Port F; press and release S3 on the Prototyping Board to send a message from Serial Port F to Serial Port E. The data sent between the serial ports will be displayed in the **STDIO** window.

Two sample programs, **SIMPLE485MASTER.C** and **SIMPLE485SLAVE.C**, are available to illustrate RS-485 master/slave communication. To run these sample programs, you will need a second Rabbit-based system with RS-485—the second system may be another RCM3900, or it may be any Rabbit single-board computer or RabbitCore module that supports RS-485 serial communication as long as you use the master or slave sample program associated with that board.

Before running either of these sample programs on the RCM3900 assembly, make sure pins 1–2 and pins 5–6 are jumpered together on header JP5 to use the RS-485 bias and termination resistors. The sample programs use Serial Port C as the RS-485 serial port, and they use PD7 to enable/disable the RS-485 transmitter.

The RS-485 connections between the slave and master devices are as follows.

- RS485+ to RS485+
- RS485– to RS485–
- GND to GND
- **SIMPLE485MASTER.C**—This program demonstrates a simple RS-485 transmission of lower case letters to a slave. The slave will send back converted upper case letters back to the master and display them in the **STDIO** window. Use **SIMPLE485SLAVE.C** to program the slave—reset the slave before you run **SIMPLE485MASTER.C** on the master.
- **SIMPLE485SLAVE.C**—This program demonstrates a simple RS-485 transmission of lower case letters to a master. The slave will send back converted upper case letters back to the master and display them in the **STDIO** window. Compile and run this program on the slave before you use **SIMPLE485MASTER.C** to program the master.

3.2.4 Real-Time Clock

If you plan to use the real-time clock functionality in your application, you will need to set the real-time clock. Set the real-time clock using the Dynamic C **SAMPLES\RTCLOCK\SETRTCKB.C** sample program by following the onscreen prompts. The Dynamic C **SAMPLES\RTCLOCK\RTC_TEST.C** sample program provides additional examples of how to read and set the real-time clock.

3.2.5 Other Sample Programs

Section 6.6 describes the TCP/IP sample programs, and Appendix C.7 provides sample programs for the optional LCD/keypad module that can be installed on the Prototyping Board.