



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 RCM3309/RCM3319

C-Programmable Core Module
with Serial Flash Mass Storage and Ethernet

User's Manual

019-0166 • 080528-C

RabbitCore RCM3309/RCM3319 User's Manual

Part Number 019-0166 • 080528-C • Printed in U.S.A.

©2008 Digi International Inc. • All rights reserved.

No part of the contents of this manual may be reproduced or transmitted in any form or by any means without the express written permission of Digi International.

Permission is granted to make one or more copies as long as the copyright page contained therein is included. These copies of the manuals may not be let or sold for any reason without the express written permission of Digi International.

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

Trademarks

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

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	1
1.1 RCM3309/RCM3319 Features	2
1.2 Comparing the RCM3309/RCM3319 and RCM3305/RCM3315	4
1.3 Advantages of the RCM3309 and RCM3319	5
1.4 Development and Evaluation Tools	6
1.4.1 RCM3309/RCM3319 Development Kit	6
1.4.2 Software	7
1.4.3 Connectivity Interface Kits	7
1.4.4 Online Documentation	7
Chapter 2. Getting Started	9
2.1 Install Dynamic C	9
2.2 Hardware Connections	10
2.2.1 Step 1 — Attach Module to Prototyping Board	10
2.2.2 Step 2 — Connect Programming Cable	11
2.2.3 Step 3 — Connect Power	12
2.3 Starting Dynamic C	13
2.4 Run a Sample Program	13
2.4.1 Troubleshooting	13
2.5 Where Do I Go From Here?	14
2.5.1 Technical Support	14
Chapter 3. Running Sample Programs	15
3.1 Introduction	15
3.2 Sample Programs	16
3.2.1 Use of Serial Flash	17
3.2.1.1 Onboard Serial Flash	17
3.2.1.2 SF1000 Serial Flash Card	17
3.2.2 Serial Communication	17
3.2.3 Real-Time Clock	19
3.2.4 RabbitNet	19
3.2.5 Other Sample Programs	19
Chapter 4. Hardware Reference	21
4.1 RCM3309/RCM3319 Digital Inputs and Outputs	22
4.1.1 Memory I/O Interface	27
4.1.2 LEDs	27
4.1.3 Other Inputs and Outputs	27
4.2 Serial Communication	28
4.2.1 Serial Ports	28
4.2.2 Ethernet Port	29
4.2.3 Serial Programming Port	30
4.3 Programming Cable	31
4.3.1 Changing Between Program Mode and Run Mode	31
4.3.2 Standalone Operation of the RCM3309/RCM3319	32

4.4 Other Hardware	33
4.4.1 Clock Doubler	33
4.4.2 Spectrum Spreader.....	33
4.5 Memory	34
4.5.1 SRAM.....	34
4.5.2 Flash EPROM.....	34
4.5.3 Serial Flash	34
4.5.4 Dynamic C BIOS Source Files.....	34
Chapter 5. Software Reference	35
5.1 More About Dynamic C	35
5.1.1 Developing Programs Remotely with Dynamic C	37
5.2 Dynamic C Functions.....	38
5.2.1 Digital I/O.....	38
5.2.2 SRAM Use.....	38
5.2.3 Serial Communication Drivers	39
5.2.4 TCP/IP Drivers	39
5.2.5 Serial Flash Drivers	39
5.2.6 Prototyping Board Function Calls	40
5.2.6.1 Board Initialization	40
5.2.6.2 Digital I/O.....	41
5.2.6.3 Switches, LEDs, and Relay	43
5.2.6.4 Serial Communication	46
5.2.6.5 RabbitNet Port	47
5.3 Upgrading Dynamic C	49
5.3.1 Extras.....	49
Chapter 6. Using the TCP/IP Features	51
6.1 TCP/IP Connections	51
6.2 TCP/IP Primer on IP Addresses	53
6.2.1 IP Addresses Explained.....	55
6.2.2 How IP Addresses are Used	56
6.2.3 Dynamically Assigned Internet Addresses.....	57
6.3 Placing Your Device on the Network	58
6.4 Running TCP/IP Sample Programs.....	59
6.4.1 How to Set IP Addresses in the Sample Programs.....	60
6.4.2 How to Set Up your Computer for Direct Connect.....	61
6.5 Run the PINGME.C Sample Program.....	62
6.6 Running Additional Sample Programs With Direct Connect	62
6.6.1 RabbitWeb Sample Programs.....	63
6.6.2 Remote Application Update	63
6.6.3 Dynamic C FAT File System, RabbitWeb, and SSL Libraries	63
6.7 Where Do I Go From Here?.....	65
Appendix A. RCM3309/RCM3319 Specifications	67
A.1 Electrical and Mechanical Characteristics	68
A.1.1 Headers	72
A.2 Bus Loading	73
A.3 Rabbit 3000 DC Characteristics	76
A.4 I/O Buffer Sourcing and Sinking Limit.....	77
A.5 Jumper Configurations	78
A.6 Conformal Coating	80
Appendix B. Prototyping Board	81
B.1 Introduction	82
B.1.1 Prototyping Board Features	83
B.2 Mechanical Dimensions and Layout	85

B.3 Power Supply	87
B.4 Using the Prototyping Board.....	88
B.4.1 Adding Other Components.....	89
B.4.2 Digital I/O.....	90
B.4.2.1 Digital Inputs	90
B.4.3 CMOS Digital Outputs.....	91
B.4.4 Sinking Digital Outputs.....	91
B.4.5 Relay Outputs	91
B.4.6 Serial Communication.....	92
B.4.6.1 RS-232	93
B.4.6.2 RS-485	94
B.4.7 RabbitNet Ports	95
B.4.8 Other Prototyping Board Modules	96
B.4.9 Quadrature Decoder	96
B.4.10 Stepper-Motor Control	96
B.5 Prototyping Board Jumper Configurations	98
B.6 Use of Rabbit 3000 Parallel Ports	100
Appendix C. LCD/Keypad Module	103
C.1 Specifications	103
C.2 Contrast Adjustments for All Boards.....	105
C.3 Keypad Labeling	106
C.4 Header Pinouts	107
C.4.1 I/O Address Assignments.....	107
C.5 Mounting LCD/Keypad Module on the Prototyping Board	108
C.6 Bezel-Mount Installation.....	109
C.6.1 Connect the LCD/Keypad Module to Your Prototyping Board.....	111
C.7 Sample Programs	112
C.8 LCD/Keypad Module Function Calls	113
C.8.1 LCD/Keypad Module Initialization.....	113
C.8.2 LEDs.....	114
C.8.3 LCD Display.....	115
C.8.4 Keypad.....	151
Appendix D. Power Supply	159
D.1 Power Supplies.....	159
D.1.1 Battery Backup.....	159
D.1.2 Battery-Backup Circuit	160
D.1.3 Reset Generator	161
Appendix O. RabbitNet	163
E.1 General RabbitNet Description	163
E.1.1 RabbitNet Connections.....	163
E.1.2 RabbitNet Peripheral Cards.....	164
E.2 Physical Implementation	165
E.2.1 Control and Routing	165
E.3 Function Calls.....	166
E.3.1 Status Byte.....	178
Index	179
Schematics	183



1. INTRODUCTION

The RCM3309 and RCM3319 RabbitCore modules feature a compact module that incorporates the latest revision of the powerful Rabbit[®] 3000 microprocessor, flash memory, mass storage (serial flash), static RAM, and digital I/O ports. The RCM3309 and RCM3319 feature an integrated 10/100Base-T Ethernet port, and provide for LAN and Internet-enabled systems to be built as easily as serial-communication systems.

In addition to the features already mentioned above, the RCM3309 and RCM3319 have two clocks (main oscillator and real-time clock), reset circuitry, and the circuitry necessary for management of battery backup of the Rabbit 3000's internal real-time clock and the static RAM. Two 34-pin headers bring out the Rabbit 3000 I/O bus lines, parallel ports, and serial ports.

The RCM3309's and the RCM3319's mass-storage capabilities make them suited to running the Dynamic C FAT file system module and the featured remote application update where data are stored and handled using the same directory file structure commonly used on PCs.

The RCM3309 or RCM3319 receives +3.3 V power from the customer-supplied motherboard on which it is mounted. The RCM3309 and RCM3319 can interface with all kinds of CMOS-compatible digital devices through the motherboard.

The Development Kit has what you need to design your own microprocessor-based system: a complete Dynamic C software development system and a Prototyping Board that allows you to evaluate the RCM3309 or RCM3319, and to prototype circuits that interface to the RCM3309 or RCM3319 module.

1.1 RCM3309/RCM3319 Features

- Small size: 1.85" x 2.73" x 0.86"
(47 mm x 69 mm x 22 mm)
- Microprocessor: 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
- 49 parallel 5 V tolerant I/O lines: 43 configurable for I/O, 3 fixed inputs, 3 fixed outputs
- Three additional digital inputs, two additional digital outputs
- External reset
- Alternate I/O bus can be configured for 8 data lines and 6 address lines (shared with parallel I/O lines), plus 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
- 4MB/8MB mass-storage flash-memory options, which are required to run the Dynamic C FAT file system module and the featured remote application update.
- Real-time clock
- Watchdog supervisor
- Provision for customer-supplied backup battery via connections on header J4
- 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

The RCM3900/RCM3910 and RCM3365/RCM3375 RabbitCore modules are similar to the RCM3305/RCM3315 and RCM3309/RCM3319, but they use fixed NAND and/or removable media for their mass-storage memories instead of the fixed serial flash options of the RCM3305/RCM3315 and the RCM3309/RCM3319.

Table 1 below summarizes the main features of the RCM3309 and the RCM3319 modules.

Table 1. RCM3309/RCM3319 Features

Feature	RCM3309	RCM3319
Microprocessor	Rabbit 3000 running at 44.2 MHz	
SRAM	512K program (fast SRAM) + 512K data	
Flash Memory (program)	512K	
Flash Memory (mass data storage)	8MB (serial flash)	4MB (serial flash)
Serial Ports	5 shared high-speed, 3.3 V CMOS-compatible ports: all 5 are configurable as asynchronous serial ports; 3 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 RCM3309 and RCM3319 are programmed over a standard PC serial port through a programming cable supplied with the Development Kit, and can also be programmed directly over an Ethernet link using the featured remote application update or the Dynamic C download manager with or without a RabbitLink.

Appendix A provides detailed specifications for the RCM3309 and the RCM3319.

1.2 Comparing the RCM3309/RCM3319 and RCM3305/RCM3315

- **Temperature Specifications** — We can no longer obtain certain components for the RCM3305/RCM3315 RabbitCore modules that support the -40°C to +70°C temperature range. RCM3305/RCM3315 RabbitCore modules manufactured after May, 2008, are specified to operate at 0°C to +70°C. The RCM3309/RCM3319, rated for -40°C to +85°C, are offered to customers requiring a larger temperature range after May, 2008.
- **Maximum Current** — The RCM3305/RCM3315 draws 250 mA vs. the 325 mA required by the RCM3309/RCM3319.
- **LEDs** — The **SPEED** and user (**USR/BSY**)LED locations have been swapped between the RCM3305/RCM3315 and the RCM3309/RCM3319, the **LNK/ACT** LEDs have been combined to one LED on the RCM3309/RCM3319, and the RCM3309/RCM3319 has an **FDX/COL** LED instead of the **SF** LED on the RCM3305/RCM3315. The **SF** LED on the RCM3305/RCM3315 blinks when data are being written to or read from the serial flash. The **FDX/COL** LED on the RCM3309/RCM3319 indicates whether the Ethernet connection is in full-duplex mode (steady on) or that a half-duplex connection is experiencing collisions (blinks).

NOTE: The change in LED indicators means that there is no indication on the RCM3309/RCM3319 when data are being written to or read from the serial flash.

- **Ethernet chip.** A different Ethernet controller chip is used on the RCM3309/RCM3319. 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 were used in your application developed for the RCM3305/RCM3315, you may run that application on the RCM3309/RCM3319 after you recompile it using Dynamic C v. 9.60.

1.3 Advantages of the RCM3309 and RCM3319

- 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 RCM3305 Series Development Kit

The RCM3305 Series Development Kit contains the hardware you need to use your RCM3309 or RCM3319 module.

- RCM3309 module.
- Prototyping Board.
- 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.

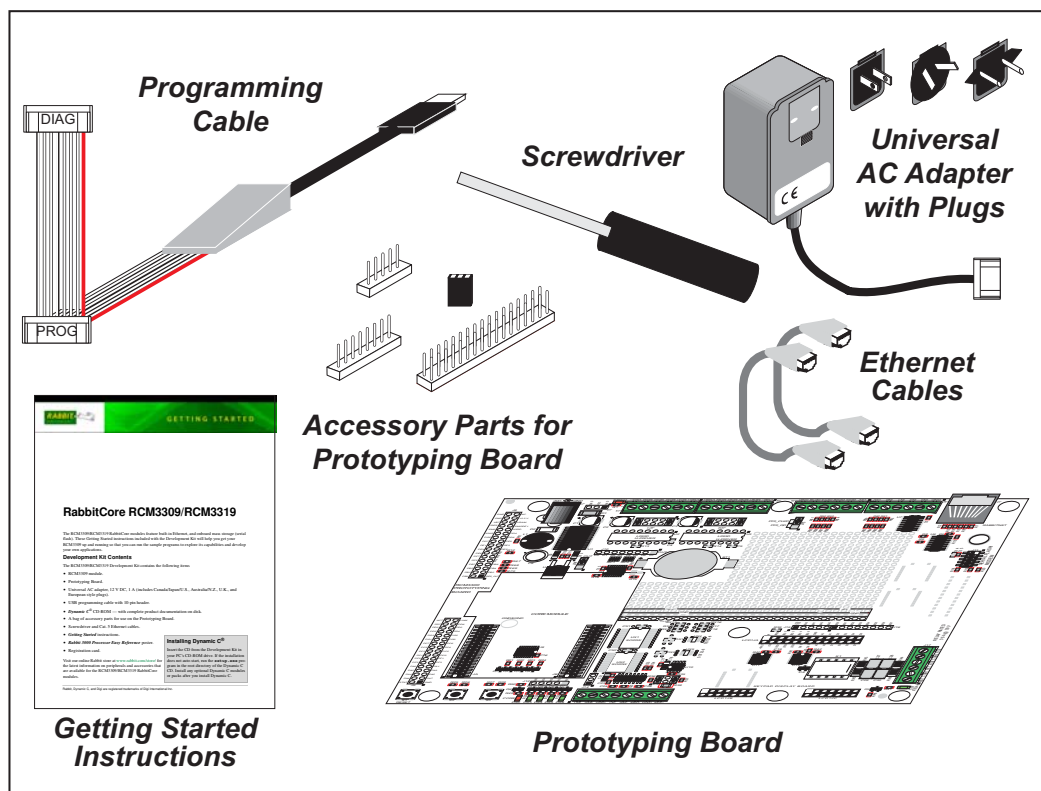


Figure 1. RCM3305 Series Development Kit

1.4.2 Software

The RCM3309 and the RCM3319 are programmed using version 9.60 of Rabbit's 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 for purchase 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 the RCM3309/RCM3319 with header sockets that have a 0.1" pitch.

- Connector Adapter Board (Part No. 151-0114)—allows you to plug the RCM3309/RCM3319 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.

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

2. GETTING STARTED

This chapter explains how to set up and use the RCM3309/RCM3319 modules with the accompanying Prototyping Board.

NOTE: It is assumed that you have a Development Kit. If you purchased an RCM3309 or RCM3319 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 RCM3309/RCM3319 (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 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 Disk 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 have purchased 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 connecting the Prototyping Board for use with Dynamic C and the sample programs:

1. Attach the RCM3309/RCM3319 module to the Prototyping Board.
2. Connect the programming cable between the RCM3309/RCM3319 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 RCM3309/RCM3319 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.

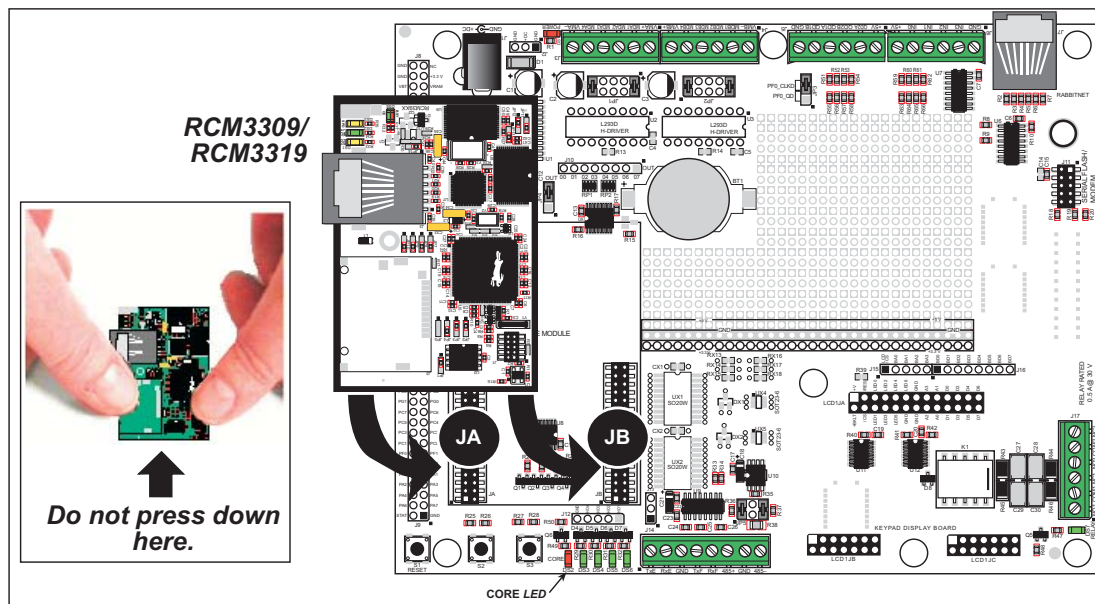


Figure 2. Install the RCM3309/RCM3319 Module on the Prototyping Board

NOTE: It is important that you line up the pins on headers J61 and J62 of the RCM3309/RCM3319 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 connectors as shown in Figure 2. Do **not** press down on the middle of the RCM3309/RCM3319 module to avoid flexing the module, which could damage the module or the components on the module.

Should you need to remove the RCM3309/RCM3319 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 RCM3309/RCM3319 to the PC running Dynamic C to download programs and to monitor the RCM3309/RCM3319 module during debugging.

Connect the 10-pin connector of the programming cable labeled **PROG** to header J1 on the RCM3309/RCM3319 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 non-programming serial connection.)

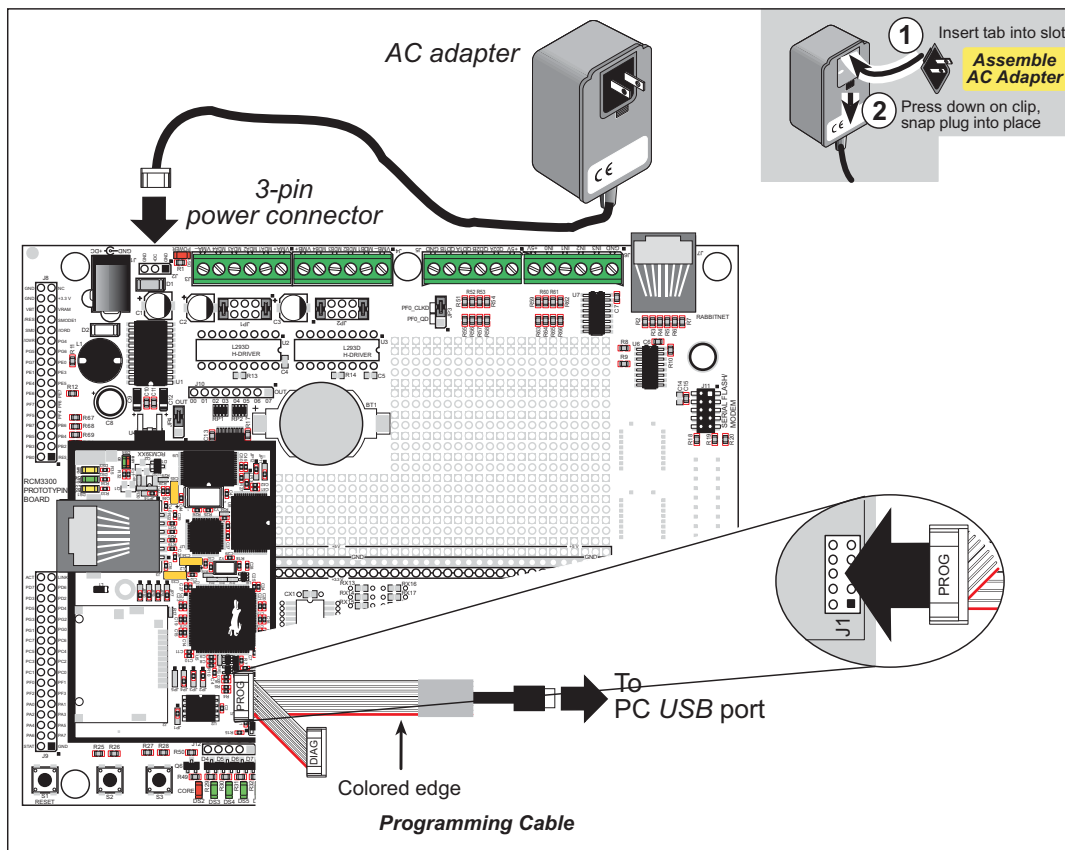


Figure 3. Connect Programming Cable and Power Supply

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 Step 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 RCM3909/RCM3319 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 RCM3309/RCM3319 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 RCM3309/RCM3319 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. 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**.

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 **STUDIO** window will open on your PC and will display a small square bouncing around in a box.

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.

2.4.1 Troubleshooting

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

- Check that the RCM3309/RCM3319 is powered correctly — the red core LED on the Prototyping Board should be lit when the RCM3309/RCM3319 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 RCM3309/RCM3319 with the marked (colored) edge of the programming cable towards pin 1 of the programming header.
- Ensure that the RCM3309/RCM3319 module is firmly and correctly installed in its connectors on the Prototyping Board.
- Dynamic C uses the USB port specified during installation. Select a different COM port within Dynamic C. From the **Options** menu, select **Project Options**, then select **Communications**. Select another USB COM port from the list, then click **OK**. Press **<Ctrl-Y>** to force Dynamic C to recompile the BIOS. If Dynamic C still reports it is unable to locate the target system, repeat the above steps until you locate the USB COM port used by the RCM3309/RCM3319 programming cable.
- 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 *RCM3309/RCM3319 User's Manual* also provides complete hardware reference information and describes the software function calls for the RCM3309 and the RCM3319, 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 RCM3309/RCM3319 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 RCM3309/RCM3319 (and for all other Rabbit hardware), you must install and use Dynamic C.

3.1 Introduction

To help familiarize you with the RCM3309 and RCM3319 modules, Dynamic C includes several sample programs. Loading, executing and studying these programs will give you a solid hands-on overview of the RCM3309/RCM3319's capabilities, as well as a quick start using 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 RCM3309/RCM3319 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 on the RCM3309/RCM3319 to your PC.
4. Power must be applied to the RCM3309/RCM3319 through the Prototyping Board.

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

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

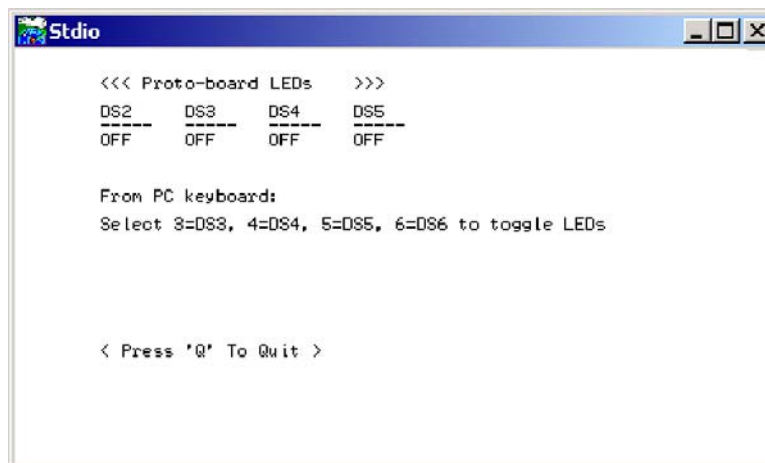
Complete information on Dynamic C is provided in the *Dynamic C User's Manual*.

3.2 Sample Programs

Of the many sample programs included with Dynamic C, several are specific to the RCM3309 and the RCM3319. Sample programs illustrating the general operation of the RCM3309/RCM3319, serial communication, and the serial flash are provided in the `SAMPLES\RCM3300` 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. Note that the RCM3309/RCM3319 must be installed on the Prototyping Board when using the sample programs described in this chapter.

- **CONTROLLED.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 **CONTROLLED.c**, the following display will appear in the Dynamic C **STDIO** window.



```
<<< Proto-board LEDs >>>
DS2   DS3   DS4   DS5
---   ---   ---   ---
OFF   OFF   OFF   OFF

From PC keyboard:
Select 3=DS3, 4=DS4, 5=DS5, 6=DS6 to toggle LEDs

< Press 'Q' To Quit >
```

Press “2” or “3” or “4” or “5” on your keyboard to select LED DS3 or DS4 or DS5 or DS6 on the Prototyping Board. Then follow the prompt in the Dynamic C **STDIO** window to turn the LED on or off.

- **FLASHLED.c**—Demonstrates assembly-language program by flashing the USR LED on the RCM3309/RCM3319 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 through screw-terminal header J17.
- **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 five programs and have an understanding of how Dynamic C and the RCM3309/RCM3319 modules interact, you can move on and try the other sample programs, or begin building your own.

3.2.1 Use of Serial Flash

3.2.1.1 Onboard Serial Flash

The following sample programs can be found in the **SAMPLES\RCM3300\SerialFlash** folder.

- **SFLASH_INSPECT.c**—This program is a handy utility for inspecting the contents of a serial flash chip. When the sample program starts running, it attempts to initialize a serial flash chip on Serial Port B. Once a serial flash chip is found, the user can perform two different commands to either print out the contents of a specified page or clear (set to zero) all the bytes in a specified page.
- **SFLASH_LOG.c**—This program runs a simple Web server and stores a log of hits in the serial flash. This log can be viewed and cleared from a browser.

3.2.1.2 SF1000 Serial Flash Card

The following sample program can be found in the **SAMPLES\RCM3300\SF1000** folder.

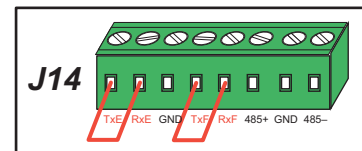
- **SERFLASHTEST.c**—An optional SF1000 Serial Flash card is required to run this demonstration. Install the Serial Flash card into socket J11 on the Prototyping Board. This sample program demonstrates how to read and write from/to the Serial Flash card.

3.2.2 Serial Communication

The following sample programs can be found in the **SAMPLES\RCM3300\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.

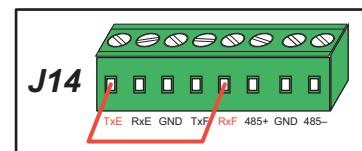
To set up the Prototyping Board, you will need to tie TxE and RxE together on the RS-232 header at J14, and you will also tie TxF and RxF together as shown in the diagram.



A repeating triangular pattern should print out in the **STDIO** window. The program will periodically switch flow control on or off to demonstrate the effect of no flow control.

- **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.

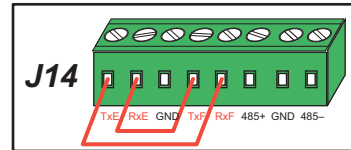
To set up the Prototyping Board, you will need to tie TxE and RxF together on the RS-232 header at J14 as shown in the diagram.



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

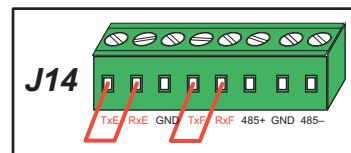
- **SIMPLE3WIRE.C**—This program demonstrates basic RS-232 serial communication. Lower case characters are sent by TxE, and are received by RxF. The characters are converted to upper case and are sent out by TxF, are received by RxE, and are displayed in the Dynamic C **STDIO** window.

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 with flow control on Serial Port F and data flow on Serial Port E.

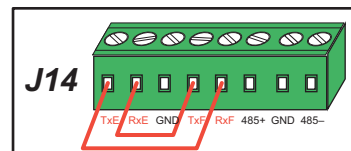
To set up the Prototyping Board, you will need to tie TxE and RxE together on the RS-232 header at J14, and you will also tie TxF and RxF together as shown in the diagram.



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.

- **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 and S3 on the Prototyping Board. 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—another Rabbit single-board computer or RabbitCore module may be used as long as you use the master or slave sample program associated with that board.

Before running either of these sample programs on the RCM3309/RCM3319 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.
- **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. Use **SIMPLE485MASTER.C** to program the master.

3.2.3 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 **SETRTCKB.C** sample program from the Dynamic C **SAMPLES\RTCKLOCK** folder, using the onscreen prompts. The **RTC_TEST.C** sample program in the Dynamic C **SAMPLES\RTCKLOCK** folder provides additional examples of how to read and set the real-time clock.

3.2.4 RabbitNet

Sample programs are available for each RabbitNet peripheral card, and can be found in the Dynamic C **SAMPLES\RabbitNet** folder. When you run any of these sample programs in conjunction with the RCM3309/RCM3319 and the Prototyping Board, you need to add the line

```
#use rcm33xx.lib
```

at the beginning of the sample program.

TIP: You need to add **#use rcm33xx.lib** at the beginning of any sample program that is not in the Dynamic C **SAMPLES\RCM3300** folder.

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.