



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 RCM3000

C-Programmable Module with Ethernet

User's Manual

019-0110 • 070831-H

RabbitCore RCM3000 User's Manual

Part Number 019-0110 • 070831-H • Printed in U.S.A.

©2002–2007 Rabbit Semiconductor 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 Rabbit Semiconductor.

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 Rabbit Semiconductor.

Rabbit Semiconductor reserves the right to make changes and improvements to its products without providing notice.

Trademarks

Rabbit and Dynamic C are registered trademarks of Rabbit Semiconductor Inc.

Rabbit 3000 and RabbitCore are trademarks of Rabbit Semiconductor Inc.

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

Rabbit Semiconductor Inc.

www.rabbit.com

Rabbit Semiconductor Inc.

www.rabbit.com

Rabbit Semiconductor Inc.

www.rabbit.com

TABLE OF CONTENTS

Chapter 1. Introduction	1
1.1 RCM3000 Features	1
1.2 Advantages of the RCM3000	3
1.3 Development and Evaluation Tools.....	3
1.4 How to Use This Manual	3
1.4.1 Additional Product Information	3
1.4.2 Online Documentation	4
Chapter 2. Hardware Setup	5
2.1 Development Kit Contents.....	5
2.2 Hardware Connections.....	6
2.2.1 Attach Module to Prototyping Board.....	6
2.2.2 Connect Programming Cable	7
2.2.3 Connect Power.....	8
2.2.3.1 Overseas Development Kits	8
2.3 Run a Sample Program	9
2.3.1 Troubleshooting	9
2.4 Where Do I Go From Here?	10
2.4.1 Technical Support	10
Chapter 3. Running Sample Programs	11
3.1 Introduction.....	11
3.2 Sample Programs	12
3.2.1 Serial Communication.....	13
3.2.2 Other Sample Programs	14
Chapter 4. Hardware Reference	15
4.1 RCM3000 Digital Inputs and Outputs	16
4.1.1 Memory I/O Interface	21
4.1.2 Other Inputs and Outputs	21
4.1.3 5 V Tolerant Inputs	21
4.2 Serial Communication	22
4.2.1 Serial Ports	22
4.2.2 Ethernet Port	22
4.2.3 Serial Programming Port.....	23
4.3 Serial Programming Cable.....	24
4.3.1 Changing Between Program Mode and Run Mode	24
4.3.2 Standalone Operation of the RCM3000.....	25
4.4 Other Hardware.....	26
4.4.1 Clock Doubler	26
4.4.2 Spectrum Spreader	26
4.5 Memory.....	27
4.5.1 SRAM	27
4.5.2 Flash EPROM	27
4.5.3 Dynamic C BIOS Source Files	27

Chapter 5. Software Reference	29
5.1 More About Dynamic C	29
5.2 Dynamic C Functions.....	31
5.2.1 Digital I/O.....	31
5.2.2 Serial Communication Drivers	31
5.2.3 TCP/IP Drivers	31
5.2.4 Prototyping Board Functions.....	32
5.2.5 Board Initialization.....	32
5.3 Upgrading Dynamic C	33
5.3.1 Upgrades.....	33
Chapter 6. Using the TCP/IP Features	35
6.1 TCP/IP Connections	35
6.2 TCP/IP Primer on IP Addresses	37
6.3 IP Addresses Explained.....	39
6.4 How IP Addresses are Used	40
6.5 Dynamically Assigned Internet Addresses	41
6.6 Placing Your Device on the Network	42
6.7 Running TCP/IP Sample Programs.....	43
6.8 How to Set IP Addresses in the Sample Programs.....	44
6.9 How to Set Up Your Computer's IP Address for Direct Connect	45
6.10 Run the PINGME.C Sample Program.....	46
6.11 Running More Sample Programs With Direct Connect.....	46
6.12 Where Do I Go From Here?.....	47
Appendix A. RCM3000 Specifications	49
A.1 Electrical and Mechanical Characteristics	50
A.1.1 Headers	53
A.1.2 Physical Mounting.....	53
A.2 Bus Loading	54
A.3 Rabbit 3000 DC Characteristics	57
A.4 I/O Buffer Sourcing and Sinking Limit.....	58
A.5 Conformal Coating.....	59
A.6 Jumper Configurations	60
Appendix B. Prototyping Board	61
B.1 Introduction	62
B.1.1 Prototyping Board Features	63
B.2 Mechanical Dimensions and Layout	65
B.3 Power Supply.....	66
B.4 Using the Prototyping Board.....	67
B.4.1 Adding Other Components	68
B.4.2 Measuring Current Draw	68
B.4.3 Other Prototyping Board Modules and Options	69
B.5 Use of Rabbit 3000 Parallel Ports.....	70
Appendix C. LCD/Keypad Module	73
C.1 Specifications.....	73
C.2 Contrast Adjustments for All Boards	75
C.3 Keypad Labeling.....	76
C.4 Header Pinouts.....	77
C.4.1 I/O Address Assignments	77
C.5 Mounting LCD/Keypad Module on the Prototyping Board.....	78
C.6 Bezel-Mount Installation	79
C.6.1 Connect the LCD/Keypad Module to Your Prototyping Board	81

C.7 LCD/Keypad Module Function Calls	82
C.7.1 LCD/Keypad Module Initialization.....	82
C.7.2 LEDs.....	82
C.7.3 LCD Display.....	83
C.7.4 Keypad.....	98
C.8 Sample Programs	101
Appendix D. Power Supply	103
D.1 Power Supplies.....	103
D.1.1 Battery-Backup Circuits.....	103
D.1.2 Reset Generator.....	104
D.2 Optional +5 V Output	104
Appendix E. Motor Control Features	105
E.1 Overview	105
E.2 Header J6	106
E.3 Using Parallel Port F	107
E.3.1 Parallel Port F Registers	107
E.4 PWM Outputs.....	110
E.5 PWM Registers.....	111
E.6 Quadrature Decoder	112
Index	115
Schematics	119

1. INTRODUCTION

The RCM3000 RabbitCore module is designed to be the heart of embedded control systems. The RCM3000 features an integrated Ethernet port and provides for LAN and Internet-enabled systems to be built as easily as serial-communication systems.

Throughout this manual, the term RCM3000 refers to the complete series of RCM3000 RabbitCore modules unless other production models are referred to specifically.

The RCM3000 has a Rabbit 3000 microprocessor operating at 29.4 MHz, static RAM, flash memory, two clocks (main oscillator and timekeeping), and the circuitry necessary for reset and 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 RCM3000 receives its +3.3 V power from the customer-supplied motherboard on which it is mounted. The RabbitCore RCM3000 can interface with all kinds of CMOS-compatible digital devices through the motherboard.

1.1 RCM3000 Features

- Small size: 1.85" x 2.65" x 0.86"
(47 mm × 67 mm × 22 mm)
- Microprocessor: Rabbit 3000 running at 29.4 MHz
- 52 parallel 5 V tolerant I/O lines: 44 configurable for I/O, 4 fixed inputs, 4 fixed outputs
- Two additional digital inputs, two additional digital outputs
- External reset input
- Alternate I/O bus can be configured for 8 data lines and 6 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
- 256K–512K flash memory, 128K–512K SRAM
- Real-time clock
- Watchdog supervisor

- Provision for customer-supplied backup battery via connections on header J2
- 10/100-compatible RJ-45 Ethernet port with 10Base-T interface
- 10-bit free-running PWM counter and four width registers
- Two-channel Input Capture can be used to time input signals from various port pins
- Two-channel Quadrature Decoder accepts inputs from external incremental encoder modules
- Six CMOS-compatible serial ports: maximum asynchronous baud rate of 1.84 Mbps, maximum synchronous baud rate of 7.35 Mbps. Four ports are configurable as a clocked serial port (SPI), and two ports are configurable as SDLC/HDLC serial ports.
- Supports 1.15 Mbps IRDA transceiver

There are two production models in the RCM3000 series. If the standard models do not serve your needs, other variations can be specified and ordered in production quantities. Contact your Rabbit Semiconductor sales representative for details.

Table 1 below highlights the differences between the two models in the RCM3000 family.

Table 1. RCM3000 Versions

Feature	RCM3000	RCM3010
Microprocessor	Rabbit 3000 running at 29.4 MHz	
Flash Memory	512K	256K
Static RAM	512K	128K
Serial Ports	6 shared high-speed, CMOS-compatible ports: 6 are configurable as asynchronous serial ports; 4 are configurable as clocked serial ports (SPI); 2 are configurable as SDLC/HDLC serial ports; 1 asynchronous clocked serial port is dedicated for programming	

NOTE: The RCM3010 is the RabbitCore module supplied with the Development Kit.

In addition, there is an RCM3100 series of RabbitCore modules that omits the RCM3000 series' Ethernet connectivity, but offer a much smaller footprint, which is about one-half that of the RCM3000 series.

The RabbitCore modules can be programmed locally, remotely, or via a network using appropriate interface hardware.

Appendix A, "RCM3000 Specifications," provides detailed specifications for the RCM3000.

1.2 Advantages of the RCM3000

- Fast time to market using a fully engineered, “ready to run” microprocessor core.
- Competitive pricing when compared with the alternative of purchasing and assembling individual components.
- Easy C-language program development and debugging
- Utility programs 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, royalty-free TCP/IP software.

1.3 Development and Evaluation Tools

A complete Development Kit, including a Prototyping Board and Dynamic C development software, is available for the RCM3000. The Development Kit puts together the essentials you need to design an embedded microprocessor-based system rapidly and efficiently.

1.4 How to Use This Manual

This user’s manual is intended to give users detailed information on the RCM3000 module. It does not contain detailed information on the Dynamic C development environment or the TCP/IP software support for the integrated Ethernet port. Most users will want more detailed information on some or all of these topics in order to put the RCM3000 module to effective use.

1.4.1 Additional Product Information

Information about the RCM3000 and its associated Development Kit and Prototyping Board can be found in this *RabbitCore RCM3000 User’s Manual*, which is provided on the accompanying CD-ROM in both HTML and Adobe PDF format.

In addition to the product-specific information contained in the *RabbitCore RCM3000 User’s Manual* (this manual), several higher level reference manuals are provided in HTML and PDF form on the accompanying CD-ROM. Advanced users will find these references valuable in developing systems based on the RCM3000 modules:

- *Dynamic C User’s Manual*
- *Dynamic C Function Reference Manual*
- *An Introduction to TCP/IP*
- *Dynamic C TCP/IP User’s Manual*
- *Rabbit 3000 Microprocessor User’s Manual*

1.4.2 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. HARDWARE SETUP

This chapter describes the RCM3000 hardware in more detail, and explains how to set up and use the accompanying Prototyping Board.

NOTE: This chapter (and this manual) assume that you have the RCM3000 Development Kit. If you purchased an RCM3000 module by itself, you will have to adapt the information in this chapter and elsewhere to your test and development setup.

2.1 Development Kit Contents

The RCM3000 Development Kit contains the following items:

- RCM3010 module with Ethernet port, 256K flash memory, and 128K SRAM.
- RCM30/31/32XX Prototyping Board.
- AC adapter, 12 V DC, 1 A. (Included only with Development Kits sold for the North American market. A header plug leading to bare leads is provided to allow overseas users to connect a power supply compatible with their local mains power.)
- 10-pin header to DB9 programming cable with integrated level-matching circuitry.
- *Dynamic C* CD-ROM, with complete product documentation on disk.
- *Getting Started* instructions.
- A bag of accessory parts for use on the Prototyping Board.
- *Rabbit 3000 Processor Easy Reference* poster.
- Registration card.

2.2 Hardware Connections

2.2.1 Attach Module to Prototyping Board

Turn the RCM3000 module so that the Ethernet connector end of the module extends off the Prototyping Board, as shown in Figure 1 below. Align the pins from headers J1 and J2 on the bottom side of the module into header sockets RCM2JA and RCM2JB on the Prototyping Board (these sockets were labeled J12 and J13 on earlier versions of the Prototyping Board).

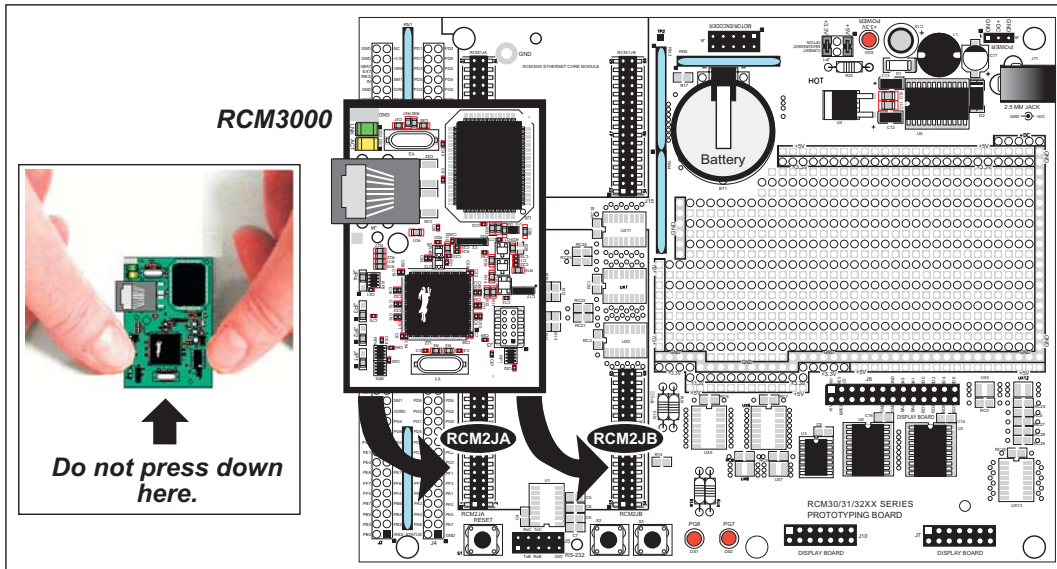


Figure 1. Install the RCM3000 Module on the Prototyping Board

Although you can install a single module into either the **MASTER** or the **SLAVE** position on the Prototyping Board, all the Prototyping Board features (switches, LEDs, serial port drivers, etc.) are connected to the **MASTER** position — install a single module in the **MASTER** position.

NOTE: It is important that you line up the pins on headers J1 and J2 of the RCM3000 module exactly with the corresponding pins of sockets RCM2JA and RCM2JB 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 1. Do **not** press down on the middle of the RCM3000 module to avoid flexing the module, which could damage the module or the components on the module.

Should you need to remove the RCM3000 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 Connect Programming Cable

The programming cable connects the RCM3000 module to the PC running Dynamic C to download programs and to monitor the module for debugging.

Connect the 10-pin connector of the programming cable labeled **PROG** to header J3 on the RCM3000 module as shown in Figure 2. 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 normal serial connection.)

NOTE: Be sure to use the programming cable (part number 101-0513) supplied with this Development Kit—the programming cable has red shrink wrap around the RS-232 converter section located in the middle of the cable. Programming cables with blue or clear shrink wrap from other Rabbit Semiconductor kits were not designed to work with RCM3000 modules.

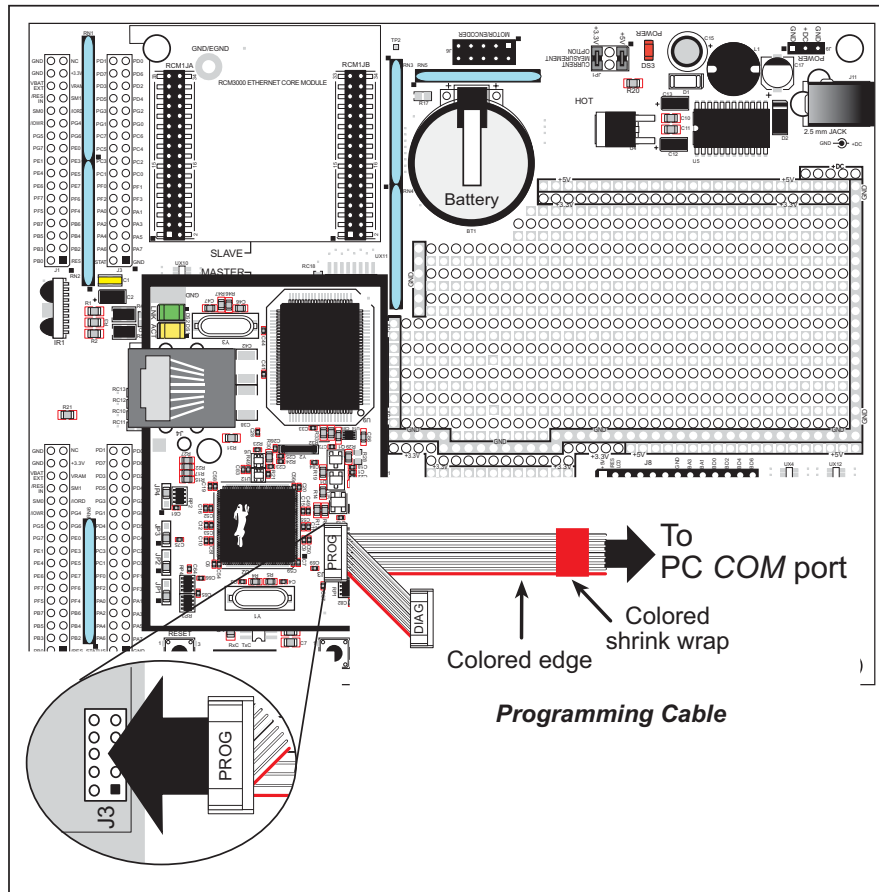


Figure 2. Connect Programming Cable to RCM3000

Connect the other end of the programming cable to a COM port on your PC.

NOTE: Some PCs now come equipped only with a USB port. It may be possible to use an RS-232/USB converter with the programming cable supplied with your RabbitCore module. An RS-232/USB converter is available through the Web store. Note that not all RS-232/USB converters work with Dynamic C.

2.2.3 Connect Power

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

Connect the wall transformer to jack J11 on the Prototyping Board as shown in Figure 3 below.

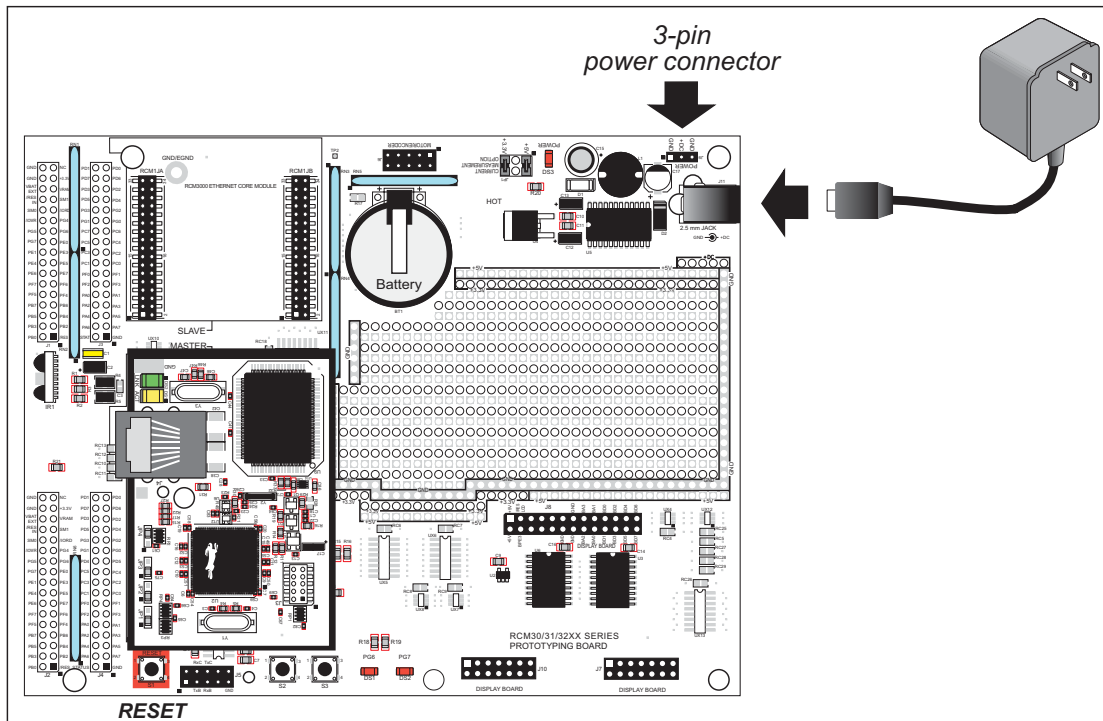


Figure 3. Power Supply Connections

Plug in the wall transformer. The power LED on the Prototyping Board should light up. The RCM3000 and the Prototyping Board are now ready to be used.

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

To power down the Prototyping Board, unplug the power connector from J11. You should disconnect power before making any circuit adjustments in the prototyping area, changing any connections to the board, or removing the RCM3000 from the Prototyping Board.

2.2.3.1 Overseas Development Kits

Development kits sold outside North America include a header connector that may be connected to 3-pin header J9 on the Prototyping Board. The connector may be attached either way as long as it is not offset to one side. The red and black wires from the connector can then be connected to the positive and negative connections on your power supply. The power supply should deliver 8 V–24 V DC at 8 W.

2.3 Run a Sample Program

If you already have Dynamic C installed, you are now ready to test your programming connections by running a sample program.

If you are using a USB port to connect your computer to the RCM3000 module, choose **Options > Project Options** and select “Use USB to Serial Converter” under the **Communications** tab.

Find the file **PONG.C**, which is in the Dynamic C **SAMPLES** folder. To run the program, open it with the **File** menu (if it is not still open), compile it using the **Compile** menu, and then run it by selecting **Run** in the **Run** menu. The **STDIO** window will open 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 6.10, “Run the PINGME.C Sample Program,” tests the TCP/IP portion of the board.

2.3.1 Troubleshooting

If Dynamic C appears to compile the BIOS successfully, but you then receive a communication error message when you compile and load the 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 in the Dynamic C **Options > Project Options > Communications** menu. Select a slower Max download baud rate.

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.

If there are any other problems:

- Check to make sure you are using the **PROG** connector, not the **DIAG** connector, on the programming cable.
- Check both ends of the programming cable to ensure that they are firmly plugged into the PC and the programming port on the RCM3000.
- Ensure that the RCM3000 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 **Communications**. Select another 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 active COM port.

2.4 Where Do I Go From Here?

We recommend that you proceed to the next chapter and install Dynamic C (if you do not already have it installed), then run the `PONG.C` sample program to verify that the RCM3000 module and the Prototyping Board are set up and functioning correctly.

If everything appears to be working, we recommend the following sequence of action:

1. Run all of the sample programs described in Chapter 3 to get a basic familiarity with Dynamic C and the RCM3000 module's capabilities.
2. For further development, refer to the *RabbitCore RCM3000 User's Manual* for details of the module's hardware and software components.
3. 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.

2.4.1 Technical Support

NOTE: If you purchased your RCM3000 through a distributor or through a Rabbit Semiconductor 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 Semiconductor Technical Bulletin Board at www.rabbit.com/support/bb/.
- Use the Technical Support e-mail form at www.rabbit.com/support/.

3. RUNNING SAMPLE PROGRAMS

To develop and debug programs for the RCM3000 (and for all other Rabbit Semiconductor hardware), you must install and use Dynamic C.

3.1 Introduction

To help familiarize you with the RCM3000 modules, Dynamic C includes several sample programs. Loading, executing and studying these programs will give you a solid hands-on overview of the RCM3000'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.

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

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

1. Your RCM3000 module must be plugged in to the Prototyping Board as described in Chapter 2, "Hardware Setup."
2. Dynamic C must be installed and running on your PC.
3. The RCM3000 module must be connected to your PC through the serial programming cable.
4. Power must be applied to the RCM3000 through the Prototyping Board.

Refer to Chapter 2, "Hardware Setup," if you need further information on these steps.

To run a sample program, open it with the **File** menu, then press function key **F9** to compile and run the program.

3.2 Sample Programs

Of the many sample programs included with Dynamic C, several are specific to the RCM3000. Sample programs illustrating the general operation of the RCM3000, and serial communication are provided in the `SAMPLES\RCM3000` 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.

- **CONTROLLED.C**—uses the **STDIO** window to demonstrate digital outputs by toggling LEDs DS1 and DS2 on the Prototyping Board on and off.

Parallel Port G bit 6 = LED DS1

Parallel Port G bit 7 = LED DS2

Once you have compile this program and it is running, you will be prompted via the Dynamic C **STDIO** window to select LED DS1 or DS2. Use your PC keyboard to make your selection.

Once you have selected the LED, you will be prompted to select to turn the LED either ON or OFF. A logic low will light up the LED you selected.

- **FLASHLED1.C**—demonstrates the use of costatements to flash LEDs DS1 and DS2 on the Prototyping Board at different rates. Once you have compile this program and it is running, LEDs DS1 and DS2 will flash on/off at different rates.
- **FLASHLED2.C**—demonstrates the use of cofunctions and costatements to flash LEDs DS1 and DS2 on the Prototyping Board at different rates. Once you have compile this program and it is running, LEDs DS1 and DS2 will flash on/off at different rates.
- **TOGGLESWITCH.C**—demonstrates the use of costatements to detect switches using the press-and-release method of debouncing. LEDs DS1 and DS2 on the Prototyping Board are turned on and off when you press switches S2 and S3.
- **IR_DEMO.C**—Demonstrates sending Modbus ASCII packets between two Prototyping Board assemblies via the IrDA transceivers with the IrDA transceivers facing each other. Note that this sample program will only work with the RCM30/31/32XX Prototyping Board.

First, compile and run this program on one Prototyping Board assembly, then remove the programming cable and press the **RESET** button on the Prototyping Board so that the first RabbitCore module is operating in the **Run** mode. Then connect the programming cable to the second Prototyping Board assembly with the RCM3000 and compile and run the same sample program. With the programming cable still connected to the second Prototyping Board assembly, press switch S2 on the second Prototyping Board to transmit a packet. Once the first Prototyping Board assembly receives a test packet, it will send back a response packet that will be displayed in the Dynamic C **STDIO** window. The test packets and response packets have different codes.

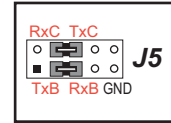
Once you have loaded and executed these sample programs and have an understanding of how Dynamic C and the RCM3000 modules interact, you can move on and try the other sample programs, or begin building your own.

3.2.1 Serial Communication

The following sample programs can be found in the `SAMPLES\RCM3000\SERIAL` folder.

- **FLOWCONTROL.C**—This program demonstrates hardware flow control by configuring Serial Port C (PC3/PC2) for CTS/RTS with serial data coming from TxB at 115,200 bps. One character at a time is received and is displayed in the **STDIO** window.

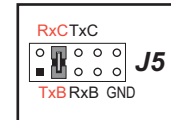
To set up the Prototyping Board, you will need to tie TxB and RxB together on the RS-232 header at J5, and you will also tie TxC and RxC together using the jumpers supplied in the Development Kit as shown in the diagram.



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.

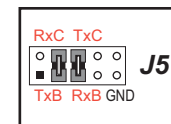
- **PARITY.C**—This program demonstrates the use of parity modes by repeatedly sending byte values 0–127 from Serial Port B to Serial Port C. The program will switch between generating parity or not on Serial Port B. Serial Port C will always be checking parity, so parity errors should occur during every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



To set up the Prototyping Board, you will need to tie TxB and RxC together on the RS-232 header at J5 using the jumpers supplied in the Development Kit as shown in the diagram.

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

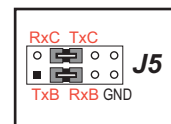
- **SIMPLE3WIRE.C**—This program demonstrates basic RS-232 serial communication. Lower case characters are sent by TxC, and are received by RxB. The characters are converted to upper case and are sent out by TxB, are received by RxC, and are displayed in the Dynamic C **STDIO** window.



To set up the Prototyping Board, you will need to tie TxB and RxC together on the RS-232 header at J5, and you will also tie RxB and TxC together using the jumpers supplied in the Development Kit as shown in the diagram.

- **SIMPLE5WIRE.C**—This program demonstrates 5-wire RS-232 serial communication with flow control on Serial Port C and data flow on Serial Port B.

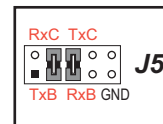
To set up the Prototyping Board, you will need to tie TxB and RxB together on the RS-232 header at J5, and you will also tie TxC and RxC together using the jumpers supplied in the Development Kit as shown in the diagram.



Once you have compiled and run this program, you can test flow control by disconnecting TxC from RxC while the program is running. Characters will no longer appear in the **STDIO** window, and will display again once TxC is connected back to RxC.

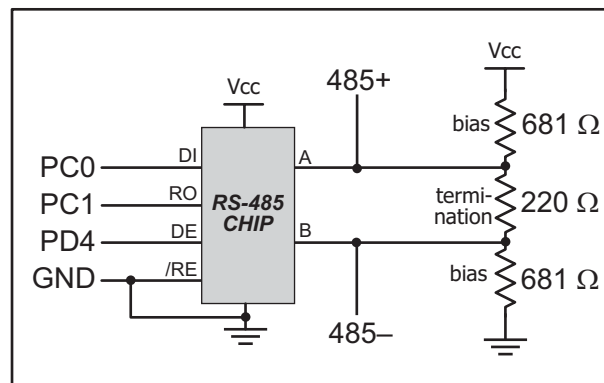
- **SWITCHCHAR.C**—This program demonstrates transmitting and then receiving an ASCII string on Serial Ports B and C. It also displays the serial data received from both ports in the **STDIO** window.

To set up the Prototyping Board, you will need to tie TxB and RxC together on the RS-232 header at J5, and you will also tie RxB and TxC together using the jumpers supplied in the Development Kit 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, and you will also have to add an RS-485 transceiver such as the SP483E and bias resistors to the RCM30/31/32XX Prototyping Board.



The diagram shows the connections. You will have to connect PC0 and PC1 (Serial Port D) on the RCM30/31/32XX Prototyping Board to the RS-485 transceiver, and you will connect PD4 to the RS-485 transceiver to enable or 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 RCM3000. The slave will send back converted upper case letters back to the master RCM3000 and display them in the **STDIO** window. Use **SIMPLE485SLAVE.C** to program the slave RCM3000.
- **SIMPLE485SLAVE.C**—This program demonstrates a simple RS-485 transmission of lower case letters to a master RCM3000. The slave will send back converted upper case letters back to the master RCM3000 and display them in the **STDIO** window. Use **SIMPLE485MASTER.C** to program the master RCM3000.

3.2.2 Other Sample Programs

Section 6.11 describes the TCP/IP sample programs, and Appendix C.8 provides sample programs for the optional LCD/keypad module that can be installed on the Prototyping Board.

4. HARDWARE REFERENCE

Chapter 3 describes the hardware components and principal hardware subsystems of the RCM3000. Appendix A, “RCM3000 Specifications,” provides complete physical and electrical specifications.

Figure 4 shows these Rabbit-based subsystems designed into the RCM3000.

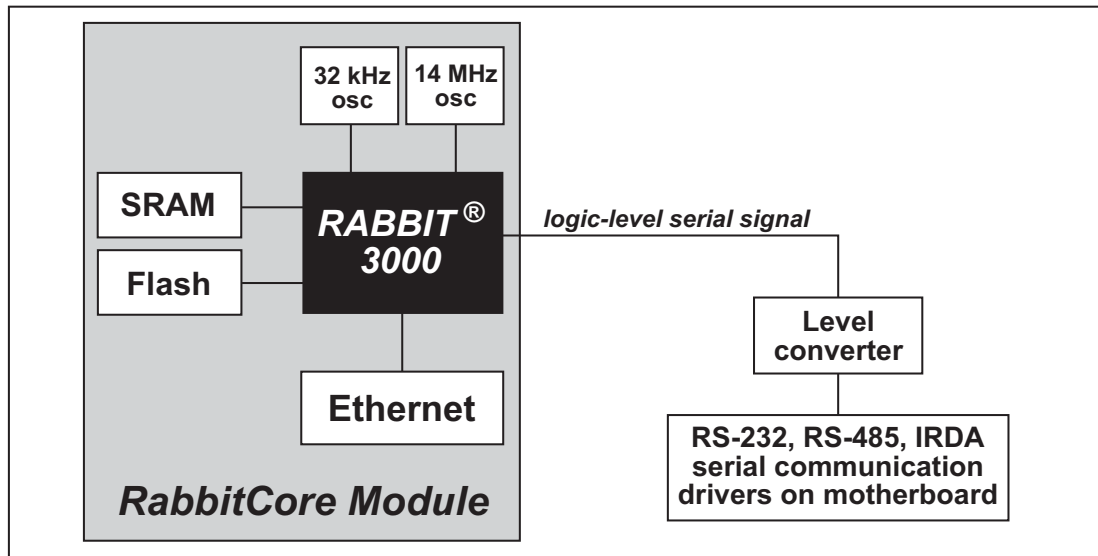


Figure 4. RCM3000 Subsystems

4.1 RCM3000 Digital Inputs and Outputs

The RCM3000 has 52 parallel I/O lines grouped in seven 8-bit ports available on headers J1 and J2. The 44 bidirectional I/O lines are located on pins PA0–PA7, PB0, PB2–PB7, PD2–PD7, PE0–PE1, PE3–PE7, PF0–PF7, and PG0–PG7.

Figure 5 shows the RCM3000 pinouts for headers J1 and J2.

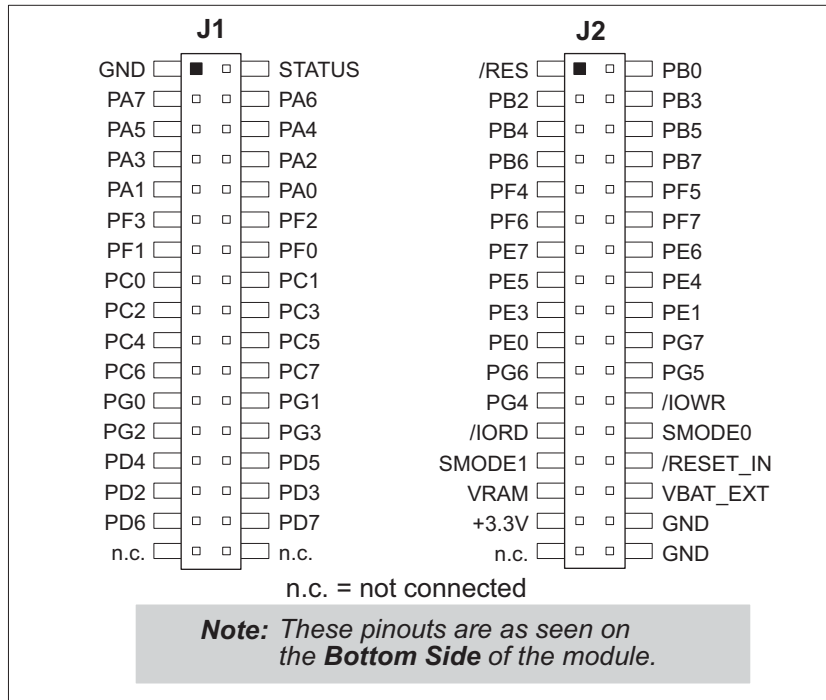


Figure 5. RCM3000 Pinouts

Headers J1 and J2 are standard 2 × 34 headers with a nominal 2 mm pitch. An RJ-45 Ethernet jack is also included with the RCM3000.

The signals labeled PD2, PD3, PD6, and PD7 on header J1 (pins 29–32) and the pins that are not connected (pins 33–34 on header J1 and pin 33 on header J2) are reserved for future use on other models in the RCM3000.

Figure 6 shows the use of the Rabbit 3000 ports in the RCM3000 modules.

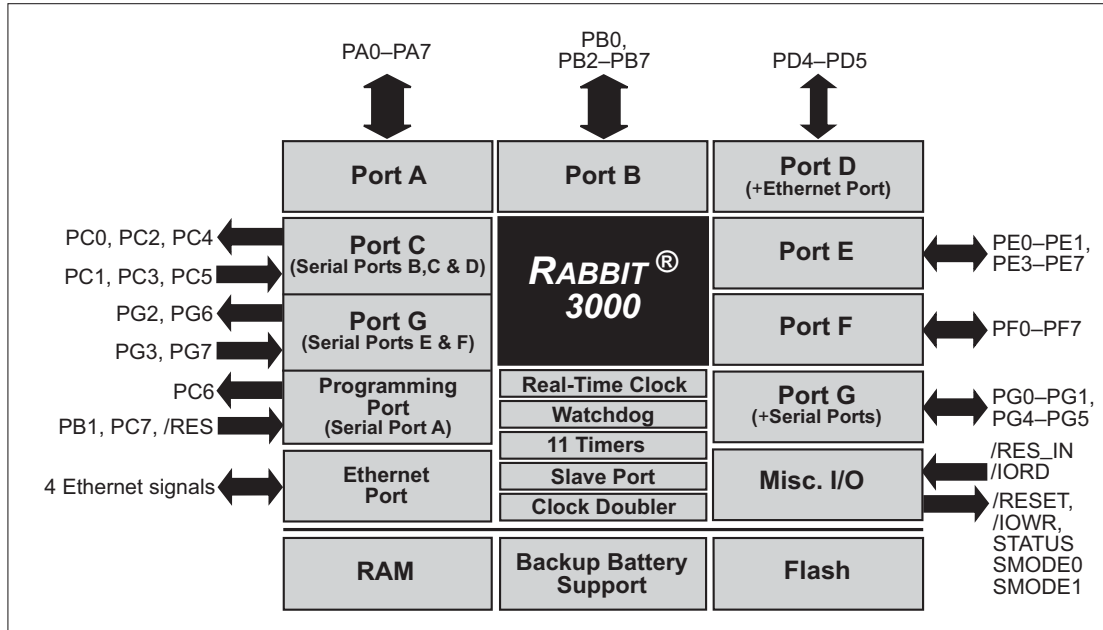


Figure 6. Use of Rabbit 3000 Ports

The ports on the Rabbit 3000 microprocessor used in the RCM3000 are configurable, and so the factory defaults can be reconfigured. Table 2 lists the Rabbit 3000 factory defaults and the alternate configurations.