



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 RCM3100

C-Programmable Module

## User's Manual

019-0115 • 070831-H

# RabbitCore RCM3100 User's Manual

Part Number 019-0115 • 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](http://www.rabbit.com), for free, unregistered download.

**Rabbit Semiconductor Inc.**

[www.rabbit.com](http://www.rabbit.com)

# TABLE OF CONTENTS

<b>Chapter 1. Introduction</b>	<b>1</b>
1.1 RCM3100 Features .....	1
1.2 Advantages of the RCM3100 .....	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 .....	3
<b>Chapter 2. Hardware Setup</b>	<b>5</b>
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 Starting Dynamic C .....	9
2.4 Run a Sample Program .....	9
2.4.1 Troubleshooting .....	9
2.5 Where Do I Go From Here? .....	10
2.5.1 Technical Support .....	10
<b>Chapter 3. Running Sample Programs</b>	<b>11</b>
3.1 Introduction.....	11
3.2 Sample Programs .....	12
3.2.1 Serial Communication.....	13
3.2.2 Real-Time Clock .....	15
3.2.3 Other Sample Programs .....	15
<b>Chapter 4. Hardware Reference</b>	<b>17</b>
4.1 RCM3100 Digital Inputs and Outputs .....	18
4.1.1 Memory I/O Interface .....	23
4.1.2 Other Inputs and Outputs .....	23
4.1.3 5 V Tolerant Inputs .....	23
4.2 Serial Communication .....	24
4.2.1 Serial Ports .....	24
4.2.2 Serial Programming Port.....	24
4.3 Serial Programming Cable.....	25
4.3.1 Changing Between Program Mode and Run Mode .....	25
4.3.2 Standalone Operation of the RCM3100.....	26
4.4 Other Hardware.....	27
4.4.1 Clock Doubler .....	27
4.4.2 Spectrum Spreader .....	27
4.5 Memory.....	28
4.5.1 SRAM .....	28
4.5.2 Flash EPROM .....	28
4.5.3 Dynamic C BIOS Source Files .....	28

<b>Chapter 5. Software Reference</b>	<b>29</b>
5.1 More About Dynamic C .....	29
5.2 Dynamic C Function Calls .....	31
5.2.1 I/O.....	31
5.2.2 Serial Communication Drivers .....	31
5.2.3 Prototyping Board Functions.....	31
5.2.3.1 Board Initialization .....	31
5.3 Upgrading Dynamic C .....	32
5.3.1 Upgrades.....	32
<b>Appendix A. RabbitCore RCM3100 Specifications</b>	<b>33</b>
A.1 Electrical and Mechanical Characteristics .....	34
A.1.1 Exclusion Zone .....	36
A.1.2 Headers .....	37
A.1.3 Physical Mounting.....	37
A.2 Bus Loading .....	38
A.3 Rabbit 3000 DC Characteristics .....	41
A.4 I/O Buffer Sourcing and Sinking Limit.....	42
A.5 Conformal Coating .....	43
A.6 Jumper Configurations .....	44
<b>Appendix B. Prototyping Board</b>	<b>45</b>
B.1 Introduction .....	46
B.1.1 Prototyping Board Features .....	47
B.2 Mechanical Dimensions and Layout .....	49
B.3 Power Supply.....	50
B.4 Using the Prototyping Board .....	51
B.4.1 Adding Other Components .....	52
B.4.2 Measuring Current Draw .....	52
B.4.3 Other Prototyping Board Modules and Options .....	53
B.5 Use of Rabbit 3000 Parallel Ports.....	54
<b>Appendix C. LCD/Keypad Module</b>	<b>57</b>
C.1 Specifications.....	57
C.2 Contrast Adjustments for All Boards .....	59
C.3 Keypad Labeling.....	60
C.4 Header Pinouts.....	61
C.4.1 I/O Address Assignments .....	61
C.5 Mounting LCD/Keypad Module on the Prototyping Board .....	62
C.6 Bezel-Mount Installation .....	63
C.6.1 Connect the LCD/Keypad Module to Your Prototyping Board .....	65
C.7 LCD/Keypad Module Function Calls.....	66
C.7.1 LCD/Keypad Module Initialization .....	66
C.7.2 LEDs .....	66
C.7.3 LCD Display .....	67
C.7.4 Keypad .....	82
C.8 Sample Programs .....	85
<b>Appendix D. Power Supply</b>	<b>87</b>
D.1 Power Supplies .....	87
D.1.1 Battery-Backup Circuits .....	87
D.1.2 Reset Generator .....	88

<b>Appendix E. Motor Control Features</b>	<b>89</b>
E.1 Overview .....	89
E.2 Header J6 .....	90
E.3 Using Parallel Port F .....	91
E.3.1 Parallel Port F Registers .....	91
E.4 PWM Outputs .....	94
E.5 PWM Registers .....	95
E.6 Quadrature Decoder .....	96
<b>Index</b>	<b>99</b>
<b>Schematics</b>	<b>103</b>



# 1. INTRODUCTION

The RCM3100 RabbitCore module is designed to be the heart of embedded control systems.

Throughout this manual, the term RCM3100 refers to the complete series of RCM3100 RabbitCore modules unless other production models are referred to specifically.

The RCM3100 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 RCM3100 receives its +3.3 V power from the customer-supplied motherboard on which it is mounted. The RabbitCore RCM3100 can interface with all kinds of CMOS-compatible digital devices through the motherboard.

## 1.1 RCM3100 Features

- Small size: 1.65" × 1.85" × 0.55"  
(42 mm × 47 mm × 14 mm)
- Microprocessor: Rabbit 3000 running at 29.4 MHz
- 54 parallel 5 V tolerant I/O lines: 46 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-bit free-running PWM counter and four pulse-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 devices
- Six CMOS-compatible serial ports: maximum asynchronous baud rate of 3.68 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 RCM3100 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 RCM3100 family.

**Table 1. RCM3100 Versions**

Feature	RCM3100	RCM3110
Microprocessor	Rabbit 3000 running at 29.4 MHz	
Flash Memory	2 × 256K	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	

In addition, there is an RCM3000 series of RabbitCore modules that includes Ethernet connectivity.

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

Appendix A, “RabbitCore RCM3100 Specifications,” provides detailed specifications for the RCM3100.

## 1.2 Advantages of the RCM3100

- 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.

## 1.3 Development and Evaluation Tools

A complete Development Kit, including a Prototyping Board and Dynamic C development software, is available for the RCM3100. 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 RCM3100 module. It does not contain detailed information on the Dynamic C development environment.

### 1.4.1 Additional Product Information

Information about the RCM3100 and its associated Development Kit and Prototyping Board can be found in the *RabbitCore RCM3100 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 RCM3100 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 RCM3100 modules:

- *Dynamic C User’s Manual*
- *Dynamic C Function Reference 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 RCM3100 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 RCM3100 Development Kit. If you purchased an RCM3100 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

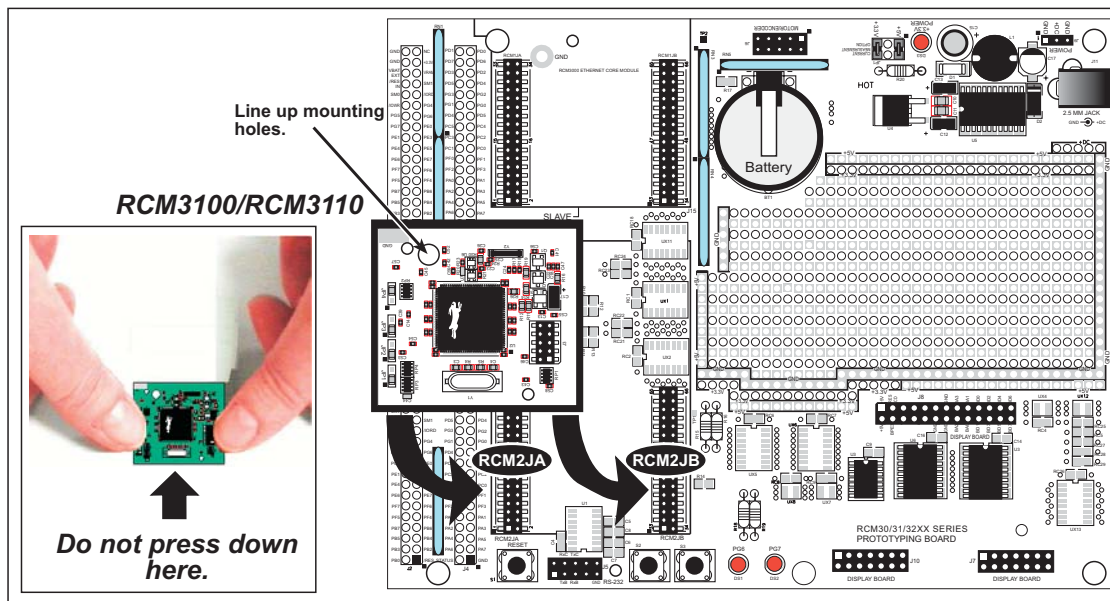
The RCM3100 Development Kit contains the following items:

- RCM3100 module, 512K flash memory, and 512K 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 RCM3100 module so that the mounting holes on the RCM3100 and on the Prototyping Board line up, 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. Installing the RCM3100 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 RCM3100 module exactly with the corresponding pins of header 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 RCM3100 module to avoid flexing the module, which could damage the module or the components on the module.

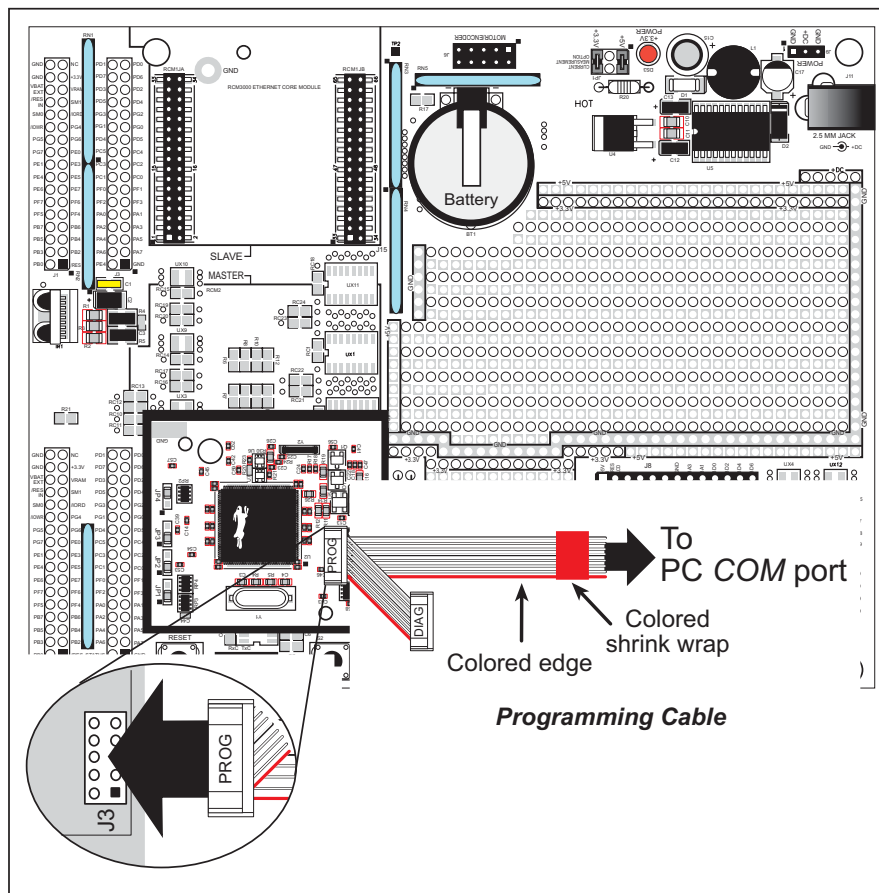
Should you need to remove the RCM3100 module, grasp it with your fingers along the sides by the connectors and gently work the module up to pull the pins away from the sockets where they are installed. Do **not** remove the module by grasping it at the top and bottom.

## 2.2.2 Connect Programming Cable

The programming cable connects the RCM3100 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 RCM3100 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 are not designed to work with RCM3100 modules.



**Figure 2. Connect Programming Cable to RCM3100**

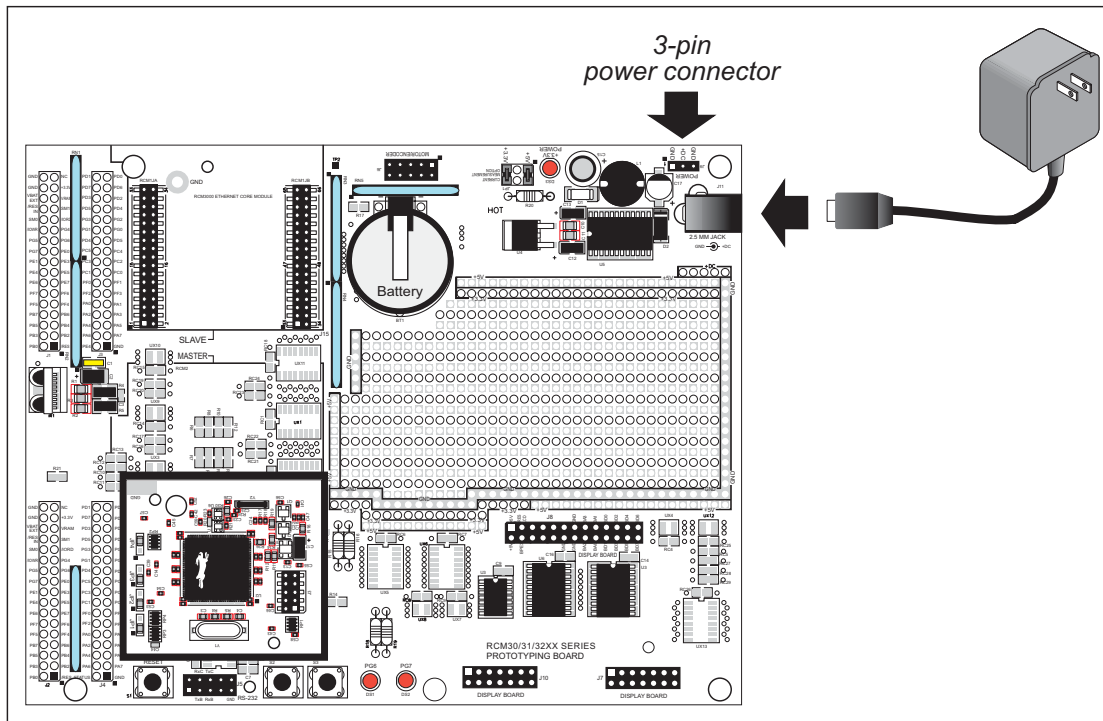
Connect the other end of the programming cable to a COM port on your PC.

**NOTE:** Some PCs now come equipped only with a USB port. It may be possible to use an RS-232/USB converter (Part No. 540-0070) with the programming cables mentioned above. 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 RCM3100 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 RCM3100 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 Starting Dynamic C

Once the RCM3100 is connected as described in the preceding pages, start Dynamic C by double-clicking on the Dynamic C icon or by double-clicking on `dcrab_XXXX.exe` in the Dynamic C root directory, where `XXXX` are version-specific characters. Dynamic C uses the serial port specified during installation.

If you are using a USB port to connect your computer to the RCM3100, choose **Options > Project Options** and select “Use USB to Serial Converter.” You may have to determine which COM port was assigned to the RS-232/USB converter.

## 2.4 Run a Sample Program

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), then compile and run it by pressing **F9**. The **STDIO** window will open and will display a small square bouncing around in a box.

This program shows that the CPU is working.

### 2.4.1 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 RCM3100.
- Ensure that the RCM3100 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.



## 2.5 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 RCM3100 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 RCM3100 module's capabilities.
2. For further development, refer to the *RabbitCore RCM3100 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*, also in the online documentation set.

### 2.5.1 Technical Support

**NOTE:** If you purchased your RCM3100 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/](http://www.rabbit.com/support/bb/).
- Use the Technical Support e-mail form at [www.rabbit.com/support/](http://www.rabbit.com/support/).

## 3. RUNNING SAMPLE PROGRAMS

To develop and debug programs for the RCM3100 (and for all other Rabbit Semiconductor hardware), you must install and use Dynamic C. Chapter 3 walks you through the sample programs associated with the RCM3100.

### 3.1 Introduction

To help familiarize you with the RCM3100 modules, Dynamic C includes several sample programs. Loading, executing and studying these programs will give you a solid hands-on overview of the RCM3100'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 RCM3100 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 RCM3100 module must be connected to your PC through the serial programming cable.
4. Power must be applied to the RCM3100 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 RCM3100. Sample programs illustrating the general operation of the RCM3100, and serial communication are provided in the **SAMPLES\RCM3100** 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 compiled and run this program, 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 compiled and run this program, 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 compiled and run this program, 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 RCM3100 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.

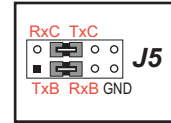
Once you have loaded and executed these sample programs and have an understanding of how Dynamic C and the RCM3100 modules interact, you can move on and try the other sample programs, or begin building your own.

### 3.2.1 Serial Communication

The following sample programs can be found in the `SAMPLES\RCM3100\SERIAL` folder.

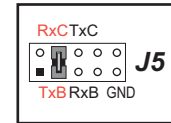
- **FLOWCONTROL.C**—This program demonstrates hardware flow control by configuring Serial Port C (PC3/PC2) for CTS/RTS with serial data coming from TxB at 115,200 bps. One character at a time is received and is displayed in the **STDIO** window.

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.

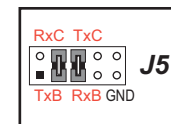
- **PARITY.C**—This program demonstrates the use of parity modes by repeatedly sending byte values 0–127 from Serial Port B to Serial Port C. The program will switch between generating parity or not on Serial Port B. Serial Port C will always be checking parity, so parity errors should occur during every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

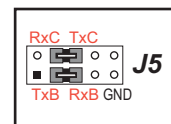
- **SIMPLE3WIRE.C**—This program demonstrates basic RS-232 serial communication. Lower case characters are sent by TxC, and are received by RxB. The characters are converted to upper case and are sent out by TxB, are received by RxC, and are displayed in the Dynamic C **STDIO** window.



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.

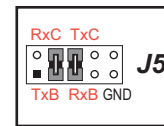
To set up the Prototyping Board, you will need to tie TxB and RxB together on the RS-232 header at J5, and you will also tie TxC and RxC together using the jumpers supplied in the Development Kit as shown in the diagram.



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.

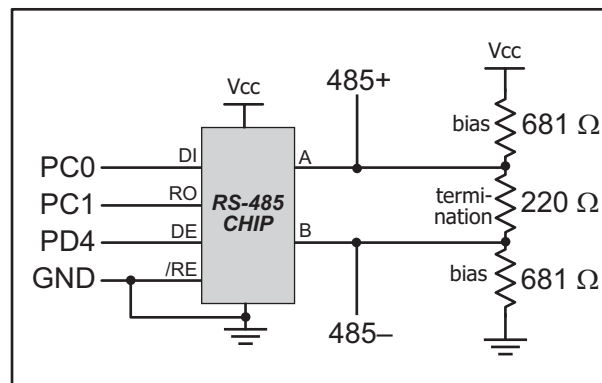
- **SWITCHCHAR.C**—This program demonstrates transmitting and then receiving an ASCII string on Serial Ports B and C. It also displays the serial data received from both ports in the **STDIO** window.

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 RCM3100. The slave will send back converted upper case letters to the master RCM3100 and display them in the **STDIO** window. Use **SIMPLE485SLAVE.C** to program the slave RCM3100.
- **SIMPLE485SLAVE.C**—This program demonstrates a simple RS-485 transmission of lower case letters to a master RCM3100. The slave will send back converted upper case letters to the master RCM3100 and display them in the **STDIO** window. Use **SIMPLE485MASTER.C** to program the master RCM3100.

### **3.2.2 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\RTCLOCK** folder, using the onscreen prompts. The **RTC\_TEST.C** sample program in the Dynamic C **SAMPLES\RTCLOCK** folder provides additional examples of how to read and set the real-time clock.

### **3.2.3 Other Sample Programs**

Appendix C.8 provides sample programs for the optional LCD/keypad module that can be installed on the Prototyping Board.



## 4. HARDWARE REFERENCE

Chapter 2 describes the hardware components and principal hardware subsystems of the RCM3100. Appendix A, “RabbitCore RCM3100 Specifications,” provides complete physical and electrical specifications.

Figure 4 shows these Rabbit-based subsystems designed into the RCM3100.

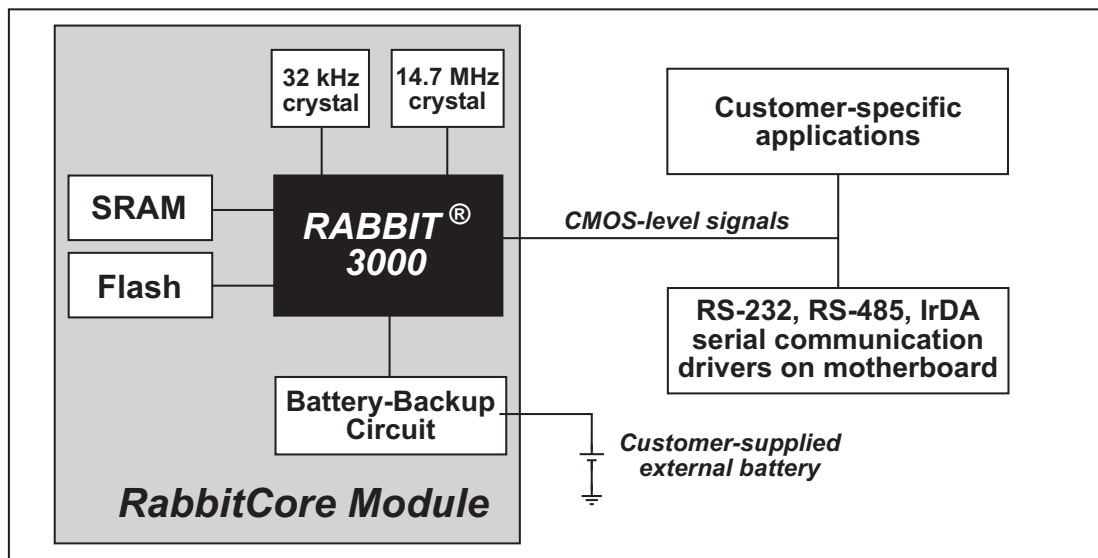


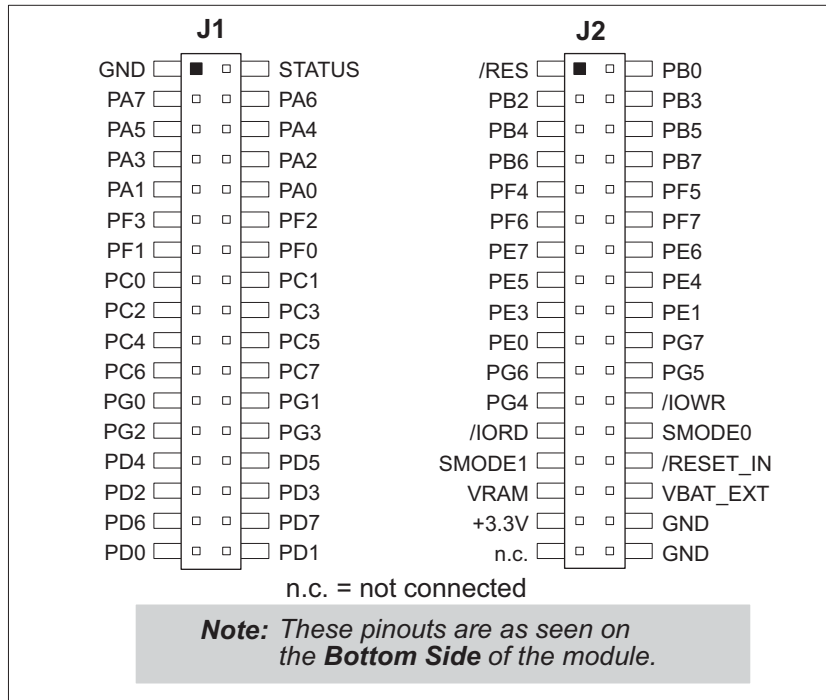
Figure 4. RCM3100 Subsystems



## 4.1 RCM3100 Digital Inputs and Outputs

The RCM3100 has 54 parallel I/O lines grouped in seven 8-bit ports available on headers J1 and J2. The 46 bidirectional I/O lines are located on pins PA0–PA7, PB0, PB2–PB7, PD0–PD7, PE0–PE1, PE3–PE7, PF0–PF7, and PG0–PG7.

Figure 5 shows the RCM3100 RabbitCore pinouts for headers J1 and J2.

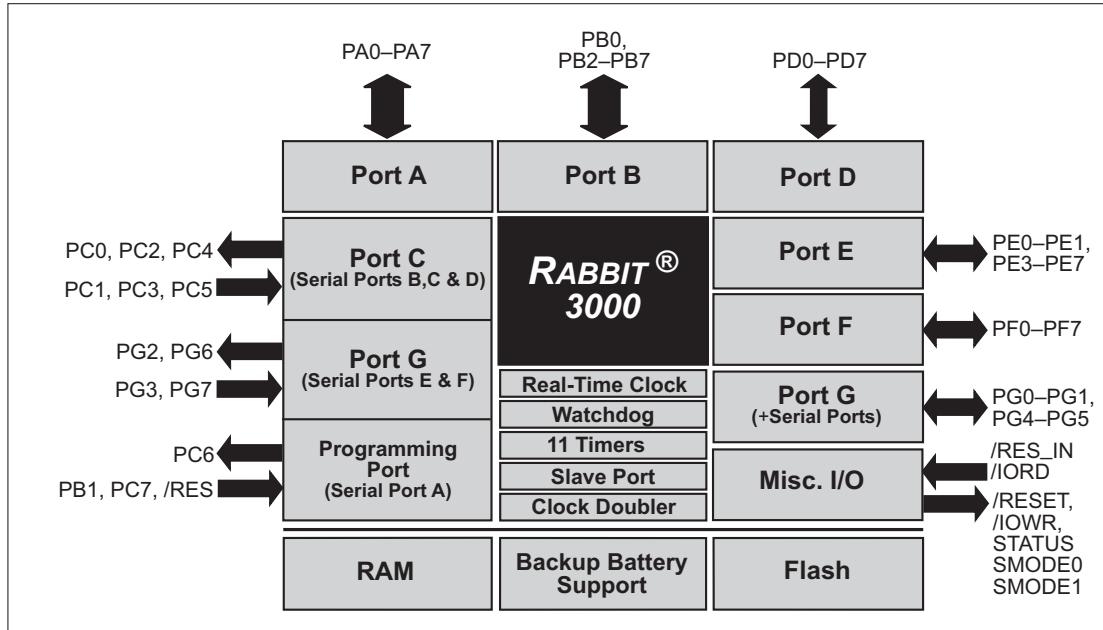


**Figure 5. RCM3100 Pinouts**

Headers J1 and J2 are standard 2 × 34 headers with a nominal 2 mm pitch.

The signals labeled PD0–PD3, PD6, and PD7 on header J1 (pins 29–34) and the pin that is not connected (pin 33 on header J2) are reserved for future use on other RabbitCore modules.

Figure 6 shows the use of the Rabbit 3000 ports in the RCM3100 RabbitCore modules.



**Figure 6. Use of Rabbit 3000 Ports**

The ports on the Rabbit 3000 microprocessor used in the RCM3100 are configurable, and so the factory defaults can be reconfigured. Table 2 lists the Rabbit 3000 factory defaults and the alternate configurations.