



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!



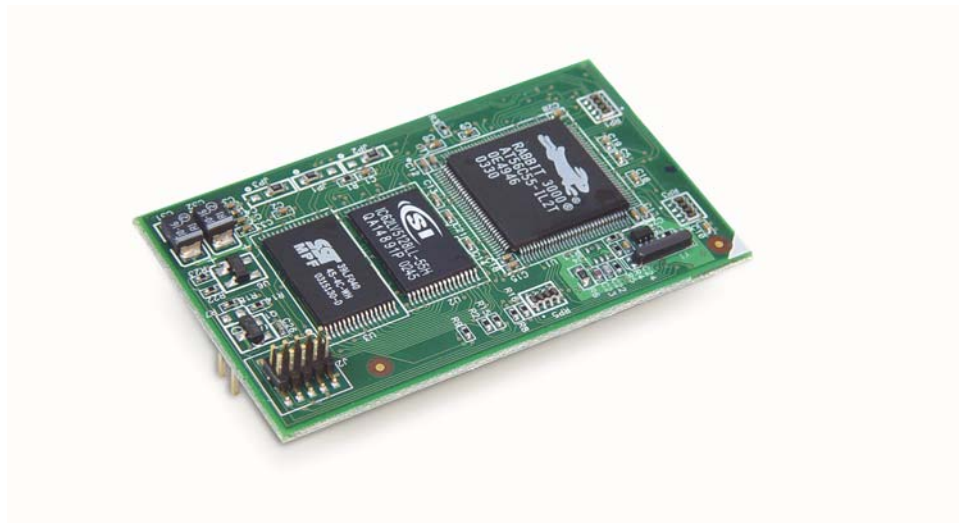
## Contact us

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 RCM3600

C-Programmable Core Module

## User's Manual

019-0135 • 070831-E

# RabbitCore RCM3600 User's Manual

Part Number 019-0135 • 070831-E • Printed in U.S.A.

©2003–2007 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 RCM3600 Features .....	1
1.2 Advantages of the RCM3600 .....	3
1.3 Development and Evaluation Tools.....	4
1.3.1 Development Kit .....	4
1.3.2 Software .....	5
1.3.3 Connectivity Interface Kits .....	5
1.3.4 Online Documentation .....	5
<b>Chapter 2. Getting Started</b>	<b>7</b>
2.1 Install Dynamic C .....	7
2.2 Hardware Connections.....	8
2.2.1 Attach Module to Prototyping Board.....	8
2.2.2 Connect Programming Cable .....	9
2.2.3 Connect Power .....	10
2.2.3.1 Overseas Development Kits .....	10
2.3 Starting Dynamic C .....	11
2.4 Run a Sample Program .....	11
2.4.1 Troubleshooting .....	11
2.5 Where Do I Go From Here? .....	12
2.5.1 Technical Support .....	12
<b>Chapter 3. Running Sample Programs</b>	<b>13</b>
3.1 Introduction.....	13
3.2 Sample Programs .....	14
3.2.1 Serial Communication.....	16
3.2.2 A/D Converter Inputs.....	18
<b>Chapter 4. Hardware Reference</b>	<b>21</b>
4.1 RCM3600 Digital Inputs and Outputs .....	22
4.1.1 Memory I/O Interface .....	26
4.1.2 Other Inputs and Outputs .....	26
4.2 Serial Communication .....	27
4.2.1 Serial Ports .....	27
4.2.2 Serial Programming Port.....	28
4.3 Serial Programming Cable.....	29
4.3.1 Changing Between Program Mode and Run Mode .....	29
4.3.2 Standalone Operation of the RCM3600.....	30
4.4 Other Hardware.....	31
4.4.1 Clock Doubler .....	31
4.4.2 Spectrum Spreader .....	31
4.5 Memory.....	32
4.5.1 SRAM .....	32
4.5.2 Flash EPROM .....	32
4.5.3 Dynamic C BIOS Source Files .....	32

<b>Chapter 5. Software Reference</b>	<b>33</b>
5.1 More About Dynamic C .....	33
5.2 Dynamic C Functions.....	35
5.2.1 Board Initialization .....	35
5.2.2 Analog Inputs .....	36
5.2.3 Digital I/O.....	52
5.2.4 Serial Communication Drivers .....	53
5.3 Upgrading Dynamic C .....	54
5.3.1 Add-On Modules .....	54
<b>Appendix A. RCM3600 Specifications</b>	<b>55</b>
A.1 Electrical and Mechanical Characteristics .....	56
A.1.1 Headers .....	59
A.2 Bus Loading .....	60
A.3 Rabbit 3000 DC Characteristics .....	63
A.4 I/O Buffer Sourcing and Sinking Limit.....	64
A.5 Conformal Coating .....	65
A.6 Jumper Configurations .....	66
<b>Appendix B. Prototyping Board</b>	<b>67</b>
B.1 Introduction .....	68
B.1.1 Prototyping Board Features .....	69
B.2 Mechanical Dimensions and Layout .....	71
B.3 Power Supply.....	72
B.4 Using the Prototyping Board .....	73
B.4.1 Adding Other Components .....	74
B.4.2 Analog Features.....	75
B.4.2.1 A/D Converter Inputs.....	75
B.4.2.2 Thermistor Input .....	77
B.4.2.3 Other A/D Converter Features .....	78
B.4.2.4 A/D Converter Calibration.....	79
B.4.3 Serial Communication .....	80
B.4.3.1 RS-232 .....	81
B.4.3.2 RS-485 .....	82
B.4.4 Other Prototyping Board Modules.....	83
B.5 RCM3600 Prototyping Board Jumper Configurations .....	84
<b>Appendix C. LCD/Keypad Module</b>	<b>87</b>
C.1 Specifications.....	87
C.2 Contrast Adjustments for All Boards .....	89
C.3 Keypad Labeling.....	90
C.4 Header Pinouts.....	91
C.4.1 I/O Address Assignments .....	91
C.5 Install Connectors on Prototyping Board .....	92
C.6 Mounting LCD/Keypad Module on the Prototyping Board .....	93
C.7 Bezel-Mount Installation .....	94
C.7.1 Connect the LCD/Keypad Module to Your Prototyping Board .....	96
C.8 Sample Programs.....	97
C.9 LCD/Keypad Module Function Calls.....	98
C.9.1 LCD/Keypad Module Initialization .....	98
C.9.2 LEDs .....	98
C.9.3 LCD Display .....	99
C.9.4 Keypad .....	119

<b>Appendix D. Power Supply</b>	<b>123</b>
D.1 Power Supplies.....	123
D.1.1 Battery-Backup Circuits.....	123
D.1.2 Reset Generator.....	124
<b>Index</b>	<b>125</b>
<b>Schematics</b>	<b>129</b>



# 1. INTRODUCTION

The RCM3600 is a compact module that incorporates the powerful Rabbit® 3000 microprocessor, flash memory, static RAM, and digital I/O ports.

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 acts as a motherboard to allow you to evaluate the RCM3600 and to prototype circuits that interface to the RCM3600 module.

The RCM3600 has a Rabbit 3000 microprocessor operating at 22.1 MHz, static RAM, flash memory, two clocks (main oscillator and real-time clock), and the circuitry necessary for reset and management of battery backup of the Rabbit 3000's internal real-time clock and the static RAM. One 40-pin header brings out the Rabbit 3000 I/O bus lines, parallel ports, and serial ports.

The RCM3600 receives its +5 V power from the customer-supplied motherboard on which it is mounted. The RCM3600 can interface with all kinds of CMOS-compatible digital devices through the motherboard.

## 1.1 RCM3600 Features

- Small size: 1.23" x 2.11" x 0.62"  
(31 mm x 54 mm x 16 mm)
- Microprocessor: Rabbit 3000 running at 22.1 MHz
- 33 parallel 5 V tolerant I/O lines: 31 configurable for I/O, 2 fixed outputs
- External reset I/O
- Alternate I/O bus can be configured for 8 data lines and 5 address lines (shared with parallel I/O lines), I/O read/write
- Ten 8-bit timers (six cascadable) and one 10-bit timer with two match registers
- 512K flash memory, 512K SRAM (options for 256K flash memory and 128K SRAM)



- Real-time clock
- Watchdog supervisor
- Connections via header J1 for customer-supplied backup battery
- 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 modules
- Four available 3.3 V CMOS-compatible serial ports with a maximum asynchronous baud rate of 2.76 Mbps. Three ports are configurable as a clocked serial port (SPI), and one port is configurable as an HDLC serial port. Shared connections to the Rabbit micro-processor make a second HDLC serial port available at the expense of two of the SPI configurable ports, giving you two HDLC ports and one asynchronous/SPI serial port.
- Supports 1.15 Mbps IrDA transceiver

There are two RCM3600 production models. If the standard models do not serve your needs, variations can be specified and ordered in production quantities. Contact your Rabbit Semiconductor sales representative for details.

Table 1 below summarizes the main features of the RCM3600.

**Table 1. RCM3600 Features**

Feature	RCM3600	RCM3610
Microprocessor	Rabbit 3000 running at 22.1 MHz	
Flash Memory	512K	256K
SRAM	512K	128K
Serial Ports	4 shared high-speed, 3.3 V CMOS-compatible ports: all 4 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 RCM3600 can be programmed through a USB port with an RS-232/USB converter, or over an Ethernet with the RabbitLink.

Appendix A provides detailed specifications for the RCM3600.

## 1.2 Advantages of the RCM3600

- 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
- Rabbit Field Utility to download compiled Dynamic C .bin files, 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.

## 1.3 Development and Evaluation Tools

### 1.3.1 Development Kit

The Development Kit contains the hardware you need to use your RCM3600 module.

- RCM3600 module.
- Prototyping Board.
- AC adapter, 12 V DC, 500 mA (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 their own power supply with a DC output of 7.5–30 V.
- Programming cable with 10-pin header and DB9 connections, and integrated level-matching circuitry.
- Cable kits to access RS-485 and analog input connectors on Prototyping Board.
- *Dynamic C* CD-ROM, with complete product documentation on disk.
- *Getting Started* instructions.
- Accessory parts for use on the Prototyping Board.
- *Rabbit 3000 Processor Easy Reference* poster.
- Registration card.

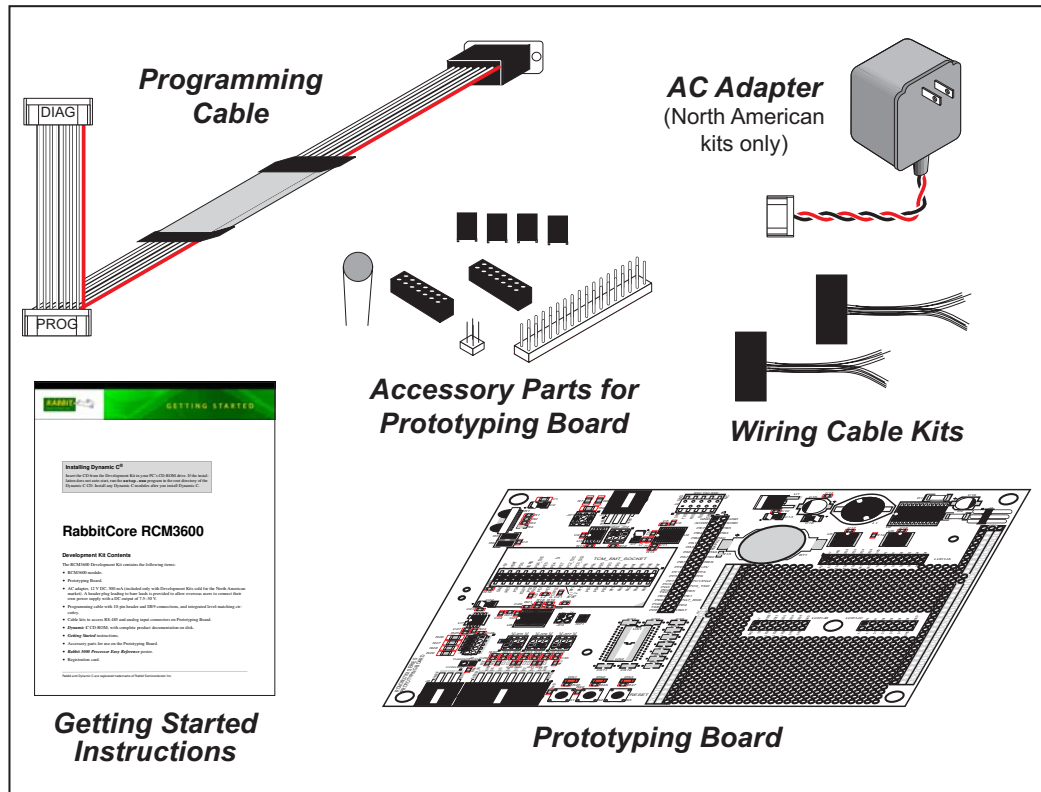


Figure 1. RCM3600 Development Kit

### 1.3.2 Software

The RCM3600 is programmed using version 8.11 or later of Dynamic C. A compatible version is included on the Development Kit CD-ROM.

Rabbit Semiconductor also offers add-on Dynamic C modules including the popular  $\mu$ C/OS-II real-time operating system, as well as point-to-point protocol (PPP), Advanced Encryption Standard (AES), and other select libraries. In addition to the Web-based technical support included at no extra charge, a one-year telephone-based technical support module is also available for purchase. Visit our Web site at [www.rabbit.com](http://www.rabbit.com) or contact your Rabbit Semiconductor sales representative or authorized distributor for further information.

### 1.3.3 Connectivity Interface Kits

Rabbit Semiconductor has available an interface kit to allow you to provide a wireless interface to the RCM3600.

- 802.11b Wi-Fi Add-On Kit (Part No. 101-0999)—The Wi-Fi Add-On Kit for the RCM3600/RCM3700 footprint consists of an RCM3600/RCM3700 Interposer Board, a Wi-Fi CompactFlash card with a CompactFlash Wi-Fi Board, a ribbon interconnecting cable, and the software drivers and sample programs to help you enable your RCM3600 module with Wi-Fi capabilities. The RCM3600/RCM3700 Interposer Board is placed between the RCM3600 module and the Prototyping Board so that the CompactFlash Wi-Fi Board, which holds the Wi-Fi CompactFlash card, can be connected to the RCM3600-based system via the ribbon cable provided.

Visit our Web site at [www.rabbit.com](http://www.rabbit.com) or contact your Rabbit Semiconductor sales representative or authorized distributor for further information.

### 1.3.4 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 describes the RCM3600 hardware in more detail, and explains how to set up and use the accompanying Prototyping Board.

**NOTE:** It is assumed that you have the RCM3600 Development Kit. If you purchased an RCM3600 module by itself or with another kit, 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 RCM3600 (and for all other Rabbit Semiconductor hardware), you must install and use Dynamic C.

If you have not yet installed Dynamic C version 8.11 (or a later version), do so now by inserting the Dynamic C CD from the RCM3600 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 on your PC 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. You may select any available port for Dynamic C's use. If you are not certain which port is available, select COM1. 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 any of the optional Dynamic C modules, install them after installing Dynamic C. The modules may be installed in any order. You must install the modules in the same directory where Dynamic C was installed.

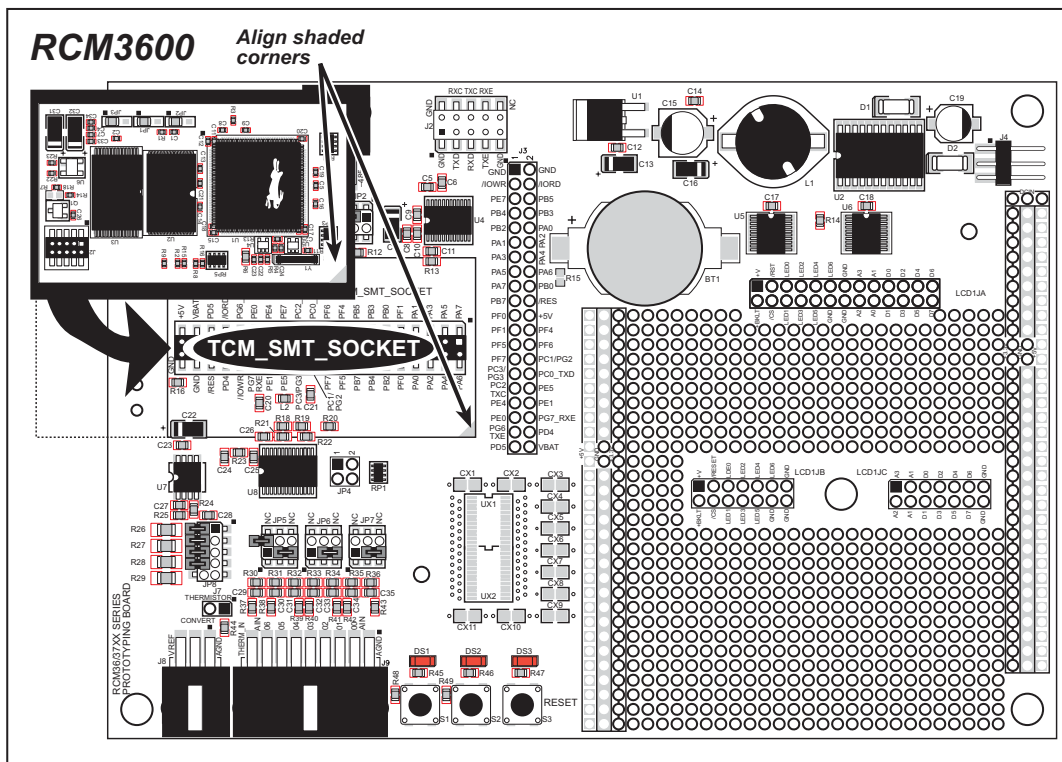
## 2.2 Hardware Connections

There are three steps to prepare the RCM3600 for use with Dynamic C and the sample programs:

1. Attach the RCM3600 module to the Prototyping Board.
2. Connect the programming cable between the RCM3600 and the COM port on the workstation PC.
3. Connect the power supply to the Prototyping Board.

### 2.2.1 Attach Module to Prototyping Board

Turn the RCM3600 module so that the Rabbit 3000 chip is facing up as shown in Figure 2 below. Insert the pins from the module's J1 header on the bottom side of the RCM3600 into the TCM\_SMT\_SOCKET socket on the Prototyping Board. The shaded corner notch at the bottom right corner of the RCM3600 module should face the same direction as the corresponding notch below it on the Prototyping Board.



**Figure 2. Install the RCM3600 Series on the Prototyping Board**

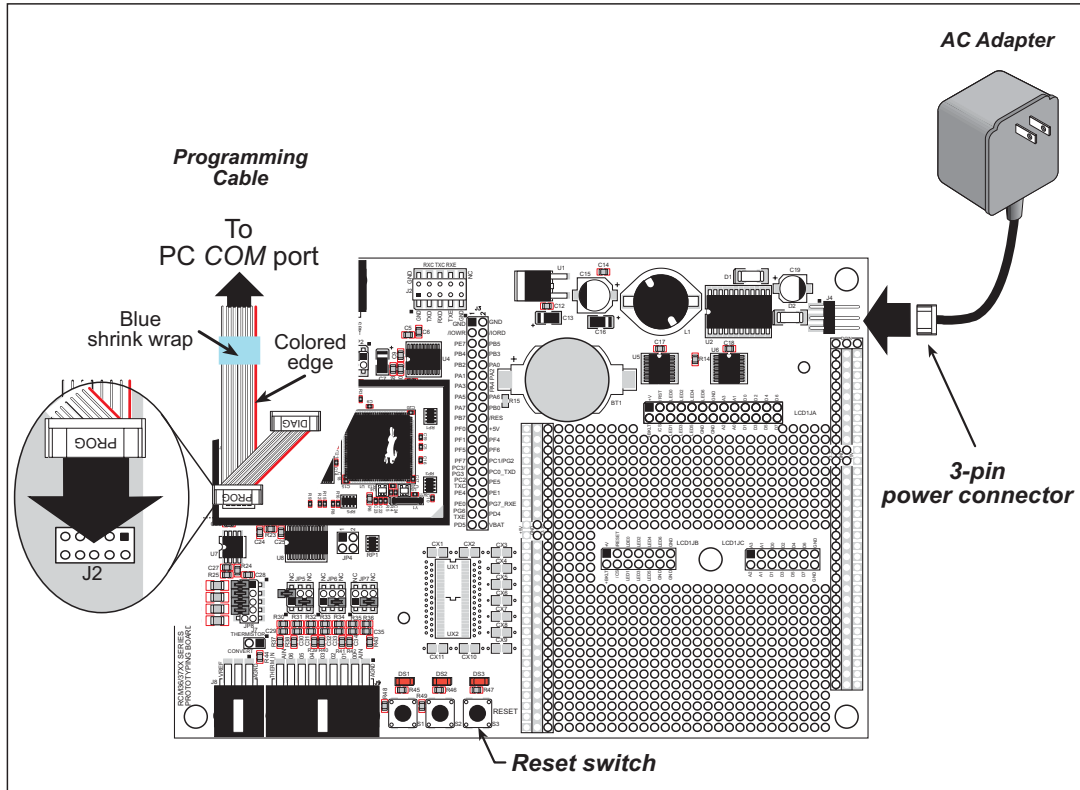
**NOTE:** It is important that you line up the pins on header J1 of the RCM3600 module exactly with the corresponding pins of the TCM\_SMT\_SOCKET socket on the Prototyping Board. The header pins may become bent or damaged if the pin alignment is off-set, and the module will not work. Permanent electrical damage to the module may also result if a misaligned module is powered up.

Press the module's pins firmly into the Prototyping Board headers.

## 2.2.2 Connect Programming Cable

The programming cable connects the RCM3600 to the PC running Dynamic C to download programs and to monitor the RCM3600 module during debugging.

Connect the 10-pin connector of the programming cable labeled **PROG** to header J2 on the RCM3600 as shown in Figure 3. 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.)



**Figure 3. Connect Programming Cable and Power Supply**

**NOTE:** Be sure to use the programming cable (Part No. 101-0542) supplied with this Development Kit—the programming cable has blue shrink wrap around the RS-232 converter section located in the middle of the cable. Programming cables from other Rabbit Semiconductor kits are not designed to work with RCM3600 modules.

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 cable supplied with the RCM3600 Development Kit. Note that not all RS-232/USB converters work with Dynamic C.



## 2.2.3 Connect Power

When all other connections have been made, connect the wall transformer to 3-pin header J4 on the Prototyping Board as shown in Figure 3. The connector may be attached either way as long as it is not offset to one side.

Plug in the wall transformer. The LED above the **RESET** button on the Prototyping Board should light up. The RCM3600 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.

### 2.2.3.1 Overseas Development Kits

Development kits sold outside North America include a header connector that may be connected to 3-pin header J4 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 7.5 V–30 V DC at 500 mA.

## 2.3 Starting Dynamic C

Once the RCM3600 is connected as described in the preceding pages, start Dynamic C by double-clicking on the Dynamic C icon or by double-clicking on **dcrabXXXX.exe** in the Dynamic C root directory, where **XXXX** are version-specific characters. Dynamic C uses the serial COM port on your PC that you specified during installation.

If you are using a USB port to connect your computer to the RCM3600 module, choose **Options > Project Options** and select “Use USB to Serial Converter.”

## 2.4 Run a Sample Program

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

### 2.4.1 Troubleshooting

If 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 that the RCM3600 is powered correctly — the power LED above the **RESET** button on the Prototyping Board should be lit.
- 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 RCM3600.
- Ensure that the RCM3600 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?

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 *RCM3600 User's Manual* also provides complete hardware reference information and describes the software function calls for the RCM3600, the Prototyping Board, and the optional LCD/keypad module.

For advanced development topics, refer to the *Dynamic C User's Manual*, which is available in the online documentation set.

### 2.5.1 Technical Support

**NOTE:** If you purchased your RCM3600 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 RCM3600 (and for all other Rabbit Semiconductor hardware), you must install and use Dynamic C.

### 3.1 Introduction

To help familiarize you with the RCM3600 modules, Dynamic C includes several sample programs. Loading, executing and studying these programs will give you a solid hands-on overview of the RCM3600's capabilities, as well as a quick start with Dynamic C as an application development tool.

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

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

1. Your RCM3600 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 (J2) on the RCM3600 to your PC.
4. Power must be applied to the RCM3600 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** or by selecting **Run** in the **Run** menu. The RCM3600 must be in Program Mode (see Figure 8) and must be connected to a PC using the programming cable.

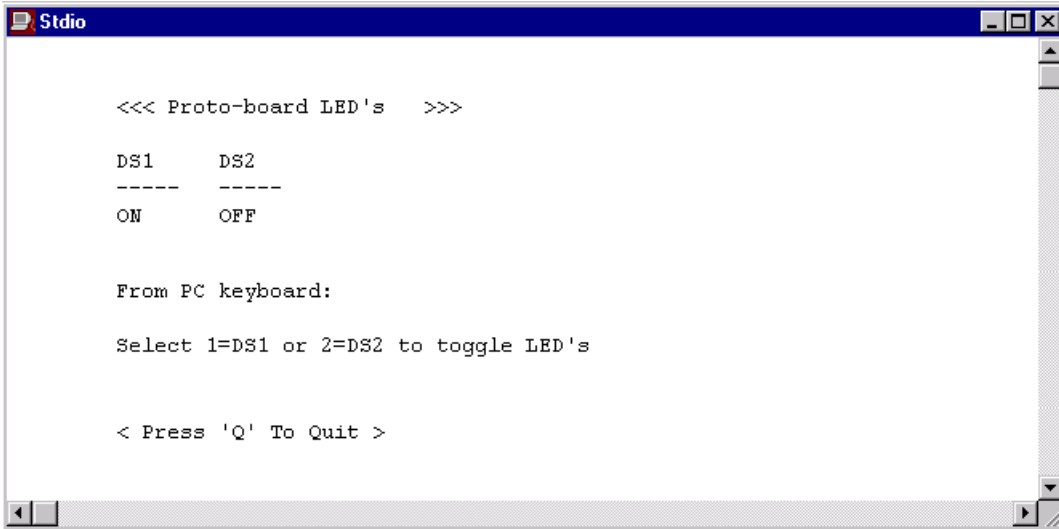
Complete information on Dynamic C is provided in the *Dynamic C User's Manual*.

## 3.2 Sample Programs

Of the many sample programs included with Dynamic C, several are specific to the RCM3600. Sample programs illustrating the general operation of the RCM3600, serial communication, and the A/D converter on the Prototyping Board are provided in the **SAMPLES\RCM3600** 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 RCM3600 must be installed on the Prototyping Board when using these sample programs. Sample programs for the optional LCD/keypad module are described in Appendix C.

- **CONTROLLED.c**—Demonstrates use of the digital inputs 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.



```
Stdio

<<< Proto-board LED's >>>

DS1    DS2
-----
ON     OFF

From PC keyboard:

Select 1=DS1 or 2=DS2 to toggle LED's

< Press 'Q' To Quit >
```

Press “1” or “2” on your keyboard to select LED DS1 or DS2 on the Prototyping Board. Then follow the prompt in the Dynamic C **STDIO** window to turn the LED on or off.

- **FLASHLED.c**—Demonstrates the use of assembly language 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.

- **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 requires a second Prototyping Board or Rabbit Semiconductor single-board computer that has an IrDA chip and is running the **IR\_DEMO.c** sample program associated with it.

First, compile and run the **IR\_DEMO.c** sample program from the **SAMPLES** folder specific to the other system on the second system, then remove the programming cable and press the **RESET** button so that the first assembly is operating in the **Run** mode. Then connect the programming cable to the RCM3600 module, and compile and run the **IR\_DEMO.c** sample program from the **SAMPLES\RCM3600** folder on the RCM3600 system. With the two IrDA transceivers facing each other, press switch S1 on the RCM3600 Prototyping Board to transmit a packet. The other system will return a response packet that will then appear in the Dynamic C **STDIO** window. The test packets and response packets have different codes.

- **DIO.c**—Demonstrates the digital I/O capabilities of the A/D converter on the Prototyping Board by configuring two lines to outputs and two lines as inputs on Prototyping Board header JP4.

Install a 2 × 2 header at JP4 on the Prototyping Board and connect pins 1–3 and pins 2–4 on header JP4 before running this sample program.

Once the sample program is compiled and running, it will prompt you in the **STDIO** window to select either pin 1 of header JP4 or pin 2 of header JP4 for the output. Once you have made that selection, you will be prompted to enter a logic 0 or 1. The specified logic level will then be output on pins 1–3 or pins 2–4 on header JP4.

- **TOGGLESWITCH.c**—Uses costatements to detect switches using debouncing. The corresponding LEDs (DS1 and DS2) will turn on or off. LEDs DS1 and DS2 on the Prototyping Board are turned on and off when you press switches S1 and S2. S1 and S2 are controlled by PF4 and PB7 respectively.

Once you have loaded and executed these five programs and have an understanding of how Dynamic C and the RCM3600 modules interact, you can move on and try the other sample programs, or begin developing your own.

### 3.2.1 Serial Communication

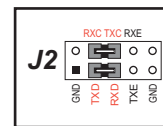
The following sample programs can be found in the Dynamic C `SAMPLES\RCM3600\SERIAL` folder.

**NOTE:** PE5 is set up to enable/disable the RS-232 chip on the Prototyping Board. This pin will also be toggled when you run RS-232 sample programs on the Prototyping Board. If you plan to use this pin for something else while you are running any of the RS-232 sample programs, comment out the following line.

```
BitWrPortI(PEDR, &PEDRShadow, 0, 5); //set low to enable rs232 device
```

- **FLOWCONTROL.C**—This program demonstrates how to configure Serial Port C for CTS/RTS with serial data coming from Serial Port D (TxD) at 115,200 bps. The serial data received are displayed in the **STDIO** window.

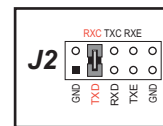
To set up the Prototyping Board, you will need to tie TxD and RxD together on the RS-232 header at J2, and you will also tie TxC and RxC together using the jumpers supplied in the Development Kit as shown in the diagram.



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.

Refer to the function description for `serDflowcontrolOn()` in the *Dynamic C Function Reference Manual* for a general description on how to set up flow-control lines.

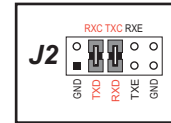
- **PARITY.C**—This program demonstrates the use of parity modes by repeatedly sending byte values 0–127 from Serial Port D to Serial Port C. The program will switch between generating parity or not on Serial Port D. Serial Port C will always be checking parity, so parity errors should occur during every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



To set up the Prototyping Board, you will need to tie TxD and RxC together on the RS-232 header at J2 using the 0.1" jumpers supplied in the Development Kit as shown in the diagram.

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

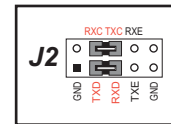
- **SIMPLE3WIRE.C**—This program demonstrates basic RS-232 serial communication. Lower case characters are sent by TxC, and are received by RxD. The characters are converted to upper case and are sent out by TxD, are received by RxC, and are displayed in the Dynamic C **STDIO** window.



To set up the Prototyping Board, you will need to tie TxD and RxC together on the RS-232 header at J2, and you will also tie RxD and TxC together using the 0.1" 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 D.

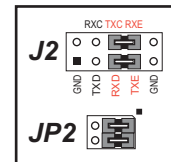
To set up the Prototyping Board, you will need to tie TxD and RxD together on the RS-232 header at J2, and you will also tie TxC and RxC together using the 0.1" jumpers supplied in the Development Kit as shown in the diagram.



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

Before running this sample program, check to make sure that Serial Port E is set up as an RS-232 serial port—pins 1–3 and pins 2–4 on header JP2 must be jumpered together using the 2 mm jumpers supplied in the Development Kit. Then connect TxC to RxE and connect RxC to TxE on the RS-232 header at J2 using the 0.1" jumpers supplied in the Development Kit as shown in the diagram.



**NOTE:** The following two sample programs illustrating RS-485 serial communication will only work with the RCM3600/RCM3700 Prototyping Board.

- **SIMPLE485MASTER.C**—This program demonstrates a simple RS-485 transmission of lower case letters to a slave RCM3600. The slave will send back converted upper case letters back to the master RCM3600 and display them in the **STDIO** window. Use **SIMPLE485SLAVE.C** to program the slave RCM3600, and check to make sure that Serial Port E is set up as an RS-485 serial port—pins 3–5 and pins 4–6 on header JP2 must be jumpered together using the 2 mm jumpers supplied in the Development Kit.
- **SIMPLE485SLAVE.C**—This program demonstrates a simple RS-485 transmission of lower case letters to a master RCM3600. The slave will send back converted upper case letters back to the master RCM3600 and display them in the **STDIO** window. Use **SIMPLE485MASTER.C** to program the master RCM3600, and check to make sure that Serial Port E is set up as an RS-485 serial port—pins 3–5 and pins 4–6 on header JP2 must be jumpered together using the 2 mm jumpers supplied in the Development Kit.





## 3.2.2 A/D Converter Inputs

The following sample programs are found in the Dynamic C `SAMPLES\RCM3600\ADC` folder.

- **AD\_CALDIFF\_CH.C**—Demonstrates how to recalibrate one differential analog input channel using two known voltages to generate the calibration constants for that channel. Constants will be rewritten into user block data area.

Before running this program, make sure that pins 1–3 are connected on headers JP5, JP6, and JP7 on the Prototyping Board. No pins are connected on header JP8.

- **AD\_CALMA\_CH.C**—Demonstrates how to recalibrate an A/D input channel being used to convert analog current measurements to generate the calibration constants for that channel.

Before running this program, make sure that pins 3–5 are connected on headers JP5, JP6, and JP7 on the Prototyping Board. Connect pins 1–2, 3–4, 5–6, 7–8 on header JP8.

- **AD\_CALSE\_ALL.C**—Demonstrates how to recalibrate all single-ended analog input channels for one gain, using two known voltages to generate the calibration constants for each channel. Constants will be rewritten into the user block data area.

Before running this program, make sure that pins 3–5 are connected on headers JP5, JP6, and JP7 on the Prototyping Board. No pins are connected on header JP8.

- **AD\_CALSE\_CHAN.C**—Demonstrates how to recalibrate one single-ended analog input channel with one gain using two known voltages to generate the calibration constants for that channel. Constants will be rewritten into user block data area.

Before running this program, make sure that pins 3–5 are connected on headers JP5, JP6, and JP7 on the Prototyping Board. No pins are connected on header JP8.

**NOTE:** The above sample programs will overwrite any existing calibration constants.

- **AD\_RDDIFF\_CH.C**—Demonstrates how to read an A/D input channel being used for a differential input using previously defined calibration constants.

Before running this program, make sure that pins 1–3 are connected on headers JP5, JP6, and JP7 on the Prototyping Board. No pins are connected on header JP8.

- **AD\_RDMA\_CH.C**—Demonstrates how to read an A/D input channel being used to convert analog current measurements using previously defined calibration constants for that channel.

Before running this program, make sure that pins 3–5 are connected on headers JP5, JP6, and JP7 on the Prototyping Board. Connect pins 1–2, 3–4, 5–6, 7–8 on header JP8.

- **AD\_RDSE\_ALL.C**—Demonstrates how to read all single-ended A/D input channels using previously defined calibration constants.

Before running this program, make sure that pins 3–5 are connected on headers JP5, JP6, and JP7 on the Prototyping Board. No pins are connected on header JP8.

- **AD\_SAMPLE.C**—Demonstrates how to use a low-level driver on single-ended inputs. The program will continuously display the voltage (average of 10 samples) that is present on the A/D channels.

Before running this program, make sure that pins 3–5 are connected on headers JP5, JP6, and JP7 on the Prototyping Board. No pins are connected on header JP8.

- **ANAINCONFIG.C**—Demonstrates how to use the Register Mode method to read single-ended analog input values for display as voltages. The sample program uses the function call `anaInConfig()` and the ADS7870 CONVERT line to accomplish this task.

Before running this program, make sure that pins 3–5 are connected on headers JP5, JP6, and JP7 on the Prototyping Board. No pins are connected on header JP8. Also connect PE4 on header J3 on the Prototyping Board to the CNVRT terminal on header J8; if you are using this sample program as a template for your own program, be aware that PE4 is also used as the IrDA FIR\_SEL pin.

- **THERMISTOR.C**—Demonstrates how to use analog input THERM\_IN7 to calculate temperature for display to the **STUDIO** window. This sample program assumes that the thermistor is the one included in the Development Kit whose values for beta, series resistance, and resistance at standard temperature are given in the part specification.

Before running this program, install the thermistor into the AIN7 and AGND holes at location J7 on the Prototyping Board.

Before running the next two sample programs, **DNLOADCALIB.C** or **UPLLOADCALIB.C**, connect your PC serial COM port to header J2 on the Prototyping Board as follows.

- Tx to RxE
- Rx to TxE
- GND to GND

Then connect pins 1–3 and 2–4 on header JP2 on the Prototyping Board.

Now start Tera Term on your PC. Once Tera Term is running, configure the serial parameters as follows:

- Baud rate 19200, 8 bits, no parity, and 1 stop bit.
- Enable the "Local Echo" option.
- Set the line feed options to Receive = CR and Transmit = CR + LF.

Now press **F9** to compile and run this program. Verify that the message "Waiting, Please Send Data file" is being display in Tera Term display window before proceeding. From within Tera Term, select **File > Send File > Path and filename**, then select the OPEN option within the dialog box. Once the data file has been downloaded, it will indicate whether the calibration data were written successfully.

- **DNLOADCALIB.C**—Demonstrates how to retrieve analog calibration data to rewrite it back to simulated EEPROM in flash with using a serial utility such as Tera Term.