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.

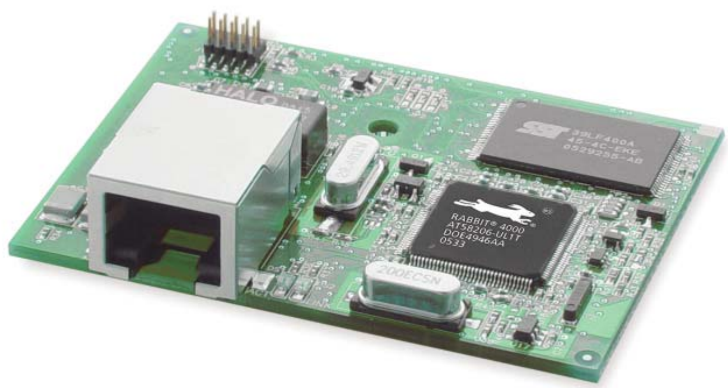
We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

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

RABBIT

PRODUCT MANUAL



RabbitCore RCM4000

C-Programmable Analog Core Module
with Ethernet

User's Manual

019–0157_J

RabbitCore RCM4000 User's Manual

Part Number 019-0157 • Printed in U.S.A.

Trademarks

Rabbit Semiconductor Inc.

www.rabbit.com

TABLE OF CONTENTS

1. INTRODUCTION

The RCM4000 series of RabbitCore® modules is one of the next generation of core modules that take advantage of new Rabbit® 4000 features such as hardware DMA, clock speeds of up to 60 MHz, I/O lines shared with up to five serial ports and four levels of alternate pin functions that include variable-phase PWM, auxiliary I/O, quadrature decoder, and input capture. Coupled with more than 500 new opcode instructions that help to reduce code size and improve processing speed, this equates to a core module that is fast, efficient, and the ideal solution for a wide range of embedded applications. The RCM4000 also features an integrated 10Base-T Ethernet port.

Each production model has a Development Kit with the essentials that you need to design your own microprocessor-based system, and includes a complete Dynamic C software development system. The Development Kits also contains a Prototyping Board that will allow you to evaluate the specific RCM4000 module and to prototype circuits that interface to the module. You will also be able to write and test software for the RCM4000 modules.

Throughout this manual, the term RCM4000 refers to the complete series of RCM4000 RabbitCore modules unless other production models are referred to specifically.

The RCM4000 has a Rabbit 4000 microprocessor operating at up to 58.98 MHz, static RAM, flash memory, NAND flash mass-storage option, an 8-channel A/D converter, two clocks (main oscillator and timekeeping), and the circuitry necessary for reset and management of battery backup of the Rabbit 4000's internal real-time clock and the static RAM. One 50-pin header brings out the Rabbit 4000 I/O bus lines, parallel ports, A/D converter channels, and serial ports.

The RCM4000 receives its +3.3 V power from the customer-supplied motherboard on which it is mounted. The RCM4000 can interface with all kinds of CMOS-compatible digital devices through the motherboard.

1.1 RCM4000 Features

- Small size: 1.84" × 2.42" × 0.77" (47 mm × 61 mm × 20 mm)
- Microprocessor: Rabbit 4000 running at 58.98 MHz
- Up to 29 general-purpose I/O lines configurable with up to four alternate functions
- 3.3 V I/O lines with low-power modes down to 2 kHz
- Up to five CMOS-compatible serial ports — four ports are configurable as a clocked serial ports (SPI), and one port is configurable as an SDLC/HDLC serial port.
- Combinations of up to eight single-ended or four differential 12-bit analog inputs (RCM4000 only)
- 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
- 512KB or 1MB flash memory, 512KB or 1 MB SRAM, with a fixed mass-storage flash-memory option that may be used with the standardized directory structure supported by the Dynamic C FAT File System module
- Real-time clock
- Watchdog supervisor

There are three RCM4000 production models. Table 1 summarizes their main features.

Table 1. RCM4000 Features

Feature	RCM4000	RCM4010	RCM4050
Microprocessor	Rabbit® 4000 at 58.98 MHz		
SRAM	512KB		1MB
Flash Memory (program)	512KB		1MB
Flash Memory (mass data storage)	32 MB (NAND flash)	—	32 MB (NAND flash)
A/D Converter	12 bits	—	—
Serial Ports	4 shared high-speed, CMOS-compatible ports: • all 4 configurable as asynchronous (with IrDA) or as clocked serial (SPI) • 1 asynchronous clocked serial port shared with programming port • 1 clocked serial port shared with A/D converter	5 shared high-speed, CMOS-compatible ports: • all 5 configurable as asynchronous (with IrDA), 4 as clocked serial (SPI), and 1 as SDLC/HDLC • 1 asynchronous clocked serial port shared with programming port	4 shared high-speed, CMOS-compatible ports: • all 4 configurable as asynchronous (with IrDA) or as clocked serial (SPI) • 1 asynchronous clocked serial port shared with programming port

The RCM4000 is programmed over a standard PC USB port through a programming cable supplied with the Development Kit.

> **NOTE:** The RabbitLink cannot be used to program RabbitCore modules based on the Rabbit 4000 microprocessor.

Appendix A provides detailed specifications for the RCM4000.

1.2 Advantages of the RCM4000

- Fast time to market using a fully engineered, “ready-to-run/ready-to-program” microprocessor 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 RCM4010 Development Kit

The RCM4010 Development Kit contains the hardware essentials you will need to use the RCM4010 module. The items in the Development Kit and their use are as follows.

- RCM4010 module.
- Prototyping Board.
- Universal AC adapter, 12 V DC, 1 A (includes Canada/Japan/U.S., Australia/N.Z., U.K., and European style plugs). Development Kits sold in North America may contain an AC adapter with only a North American style plug.
- USB programming cable with 10-pin header.
- 10-pin header to DB9 serial cable.
- ***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 4000 Processor Easy Reference*** poster.
- Registration card.

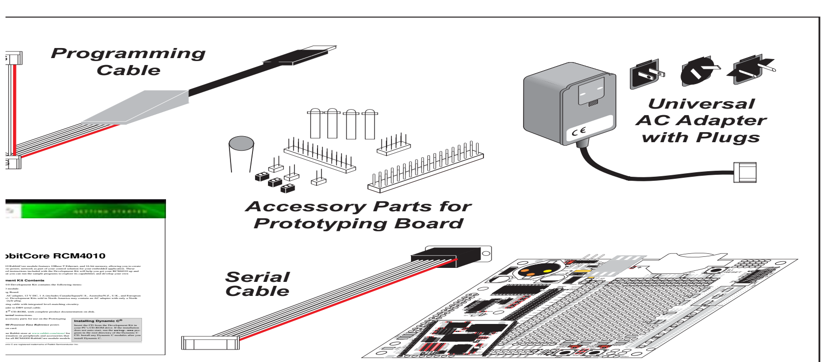


Figure 1. RCM4010 Development Kit

1.3.2 RCM4000 Analog Development Kit

The RCM4000 Analog Development Kit contains the hardware essentials you will need to use the RCM4000 module. The RCM4000 Analog Development Kit contents are similar to those of the RCM4010 Development Kit, except that the RCM4000 module is included instead of the RCM4010 module.

1.3.3 Software

The RCM4000 is programmed using version 10.03 or later of Dynamic C. A compatible version is included on the Development Kit CD-ROM.

Rabbit also offers add-on Dynamic C modules containing the popular µC/OS-II real-time operating system, as well as 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 or contact your Rabbit 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 RCM4000 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 RCM4000 Analog or the RCM4010 Development Kit. If you purchased an RCM4000 or RCM4010 module by itself, you will have to adapt the information in this chapter and elsewhere to your test and development setup.

2.1 Install Dynamic C

To develop and debug programs for the RCM4000 series of modules (and for all other Rabbit hardware), you must install and use Dynamic C.

If you have not yet installed Dynamic C version 10.03 (or a later version), do so now by inserting the Dynamic C CD from the Development Kit in your PC's CD-ROM drive. If autorun is enabled, the CD installation will begin automatically.

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

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

Dynamic C uses a COM (serial) port to communicate with the target development system. The installation allows you to choose the COM port that will be used. The default selection is COM1. 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 new 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 connecting the Prototyping Board for use with Dynamic C and the sample programs:

1. Prepare the Prototyping Board for Development.
2. Attach the RCM4000 or RCM4010 module to the Prototyping Board.
3. Connect the programming cable between the RCM4000 or RCM4010 and the PC.
4. Connect the power supply to the Prototyping Board.

2.2.1 Step 1 — Prepare the Prototyping Board for Development

Snap in four of the plastic standoffs supplied in the bag of accessory parts from the Development Kit in the holes at the corners as shown.

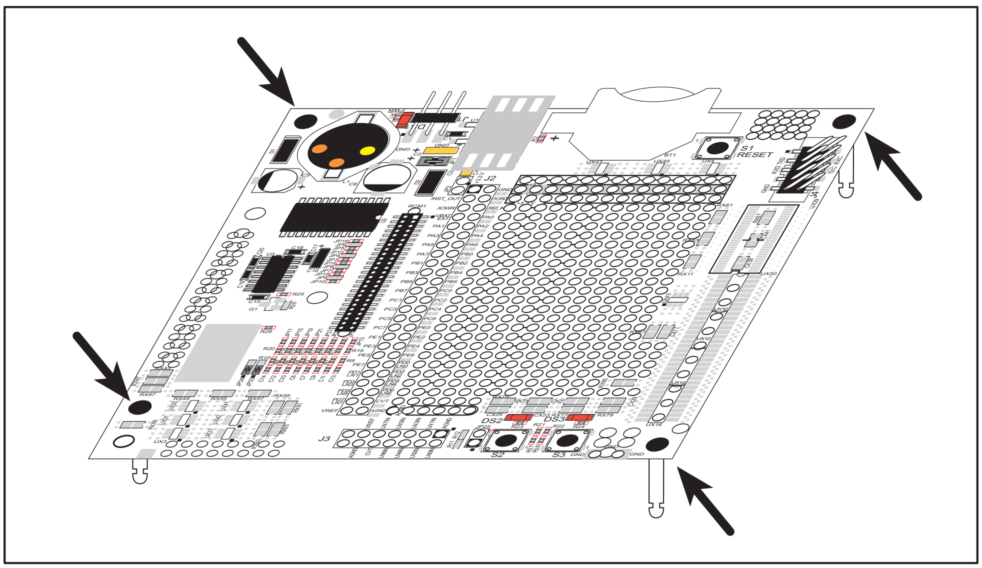


Figure 2. Insert Standoffs

2.2.2 Step 2 — Attach Module to Prototyping Board

Turn the RCM4000/RCM4010/RCM4050 module so that the mounting holes line up with the corresponding holes on the Prototyping Board. Insert the metal standoffs as shown in Figure 3, secure them from the bottom using the 4-40 screws and washers, then insert the module's header J3 on the bottom side into socket RCM1 on the Prototyping Board. (You may use plastic standoffs instead of the metal standoffs and screws.)

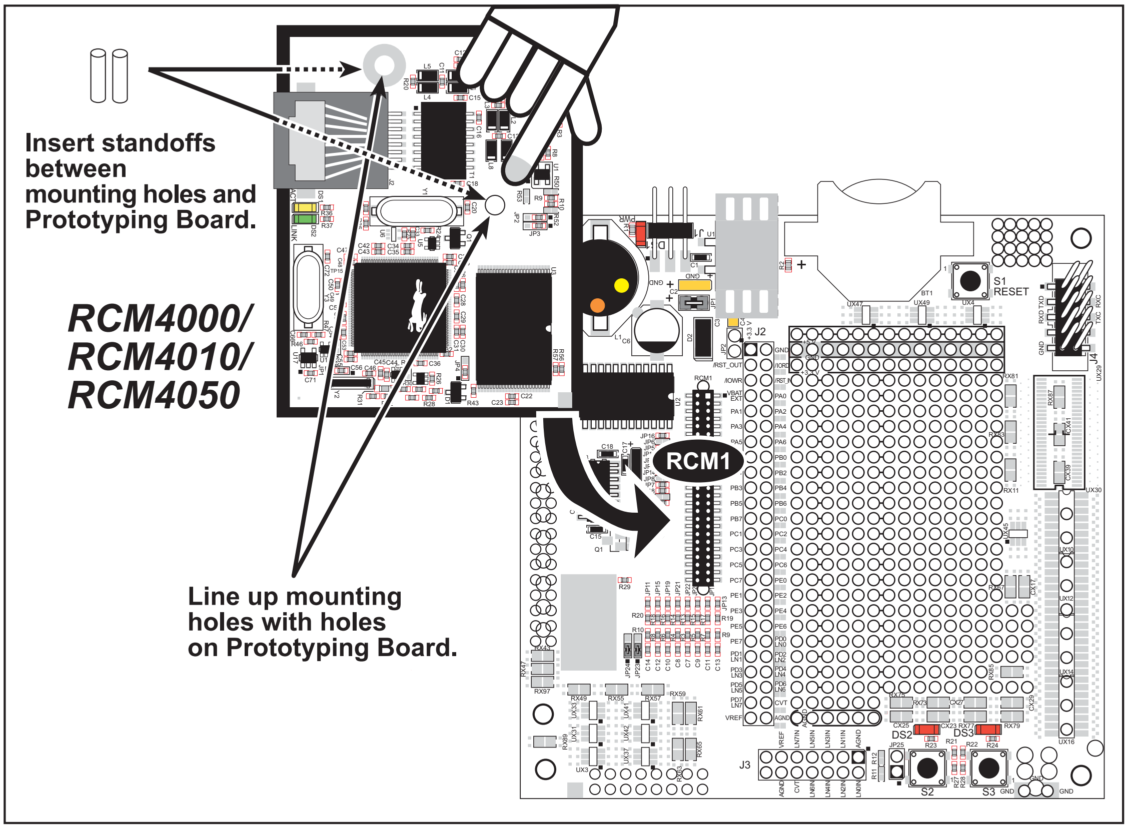


Figure 3. Install the Module on the Prototyping Board

> **NOTE:** It is important that you line up the pins on header J3 of the module exactly with socket RCM1 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 gently into the Prototyping Board socket—press down in the area above the header pins. For additional integrity, you may secure the RCM4000/RCM4010/RCM4050 to the standoffs from the top using the remaining two screws and washers.

2.2.3 Step 3 — Connect Programming Cable

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

Connect the 10-pin connector of the programming cable labeled **PROG** to header J1 on the RCM4000/RCM4010/RCM4050 as shown in Figure 4. 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.)

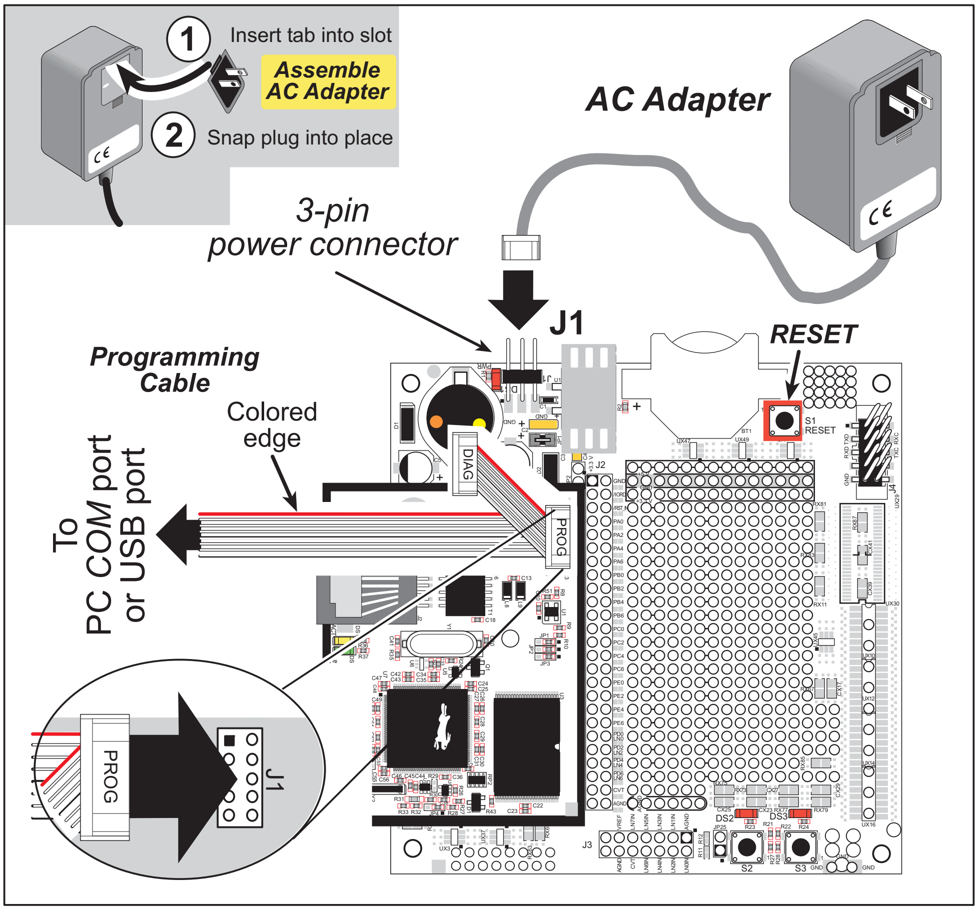


Figure 4. Connect Programming Cable and Power Supply

NOTE: Never disconnect the programming cable by pulling on the ribbon cable. Carefully pull on the connector to remove it from the header.

NOTE: Either a serial or a USB programming cable was supplied with this Development Kit. If you have a serial programming cable, an RS-232/USB converter (Rabbit Part No. 20-151-0178) is available to allow you to use the serial programming cable with a USB port.

Depending on the programming cable, connect the other end to a COM port or a USB port on your PC.

If you are using a USB programming cable, your PC should recognize the new USB hardware, and the LEDs in the shrink-wrapped area of the programming cable will flash — if you get an error message, you will have to install USB drivers. Drivers for Windows XP are available in the Dynamic C **`Drivers\Rabbit USB Programming Cable\WinXP_2K`** folder — double-click **`DPInst.exe`** to install the USB drivers. Drivers for other operating systems are available online at www.ftdichip.com/Drivers/VCP.htm.

2.2.4 Step 4 — Connect Power

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

If you have the universal AC adapter, prepare the AC adapter for the country where it will be used by selecting the appropriate plug. Snap in the top of the plug assembly into the slot at the top of the AC adapter as shown in Figure 4, then press down on the plug until it clicks into place.

Connect the AC adapter to 3-pin header J1 on the Prototyping Board as shown in Figure 4 above. The connector may be attached either way as long as it is not offset to one side—the center pin of J1 is always connected to the positive terminal, and either edge pin is ground.

Plug in the AC adapter. The **PWR** LED on the Prototyping Board next to the power connector at J1 should light up. The RCM4000/RCM4010/RCM4050 and the Prototyping Board are now ready to be used.

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

To power down the Prototyping Board, unplug the power connector from J1. You should disconnect power before making any circuit adjustments in the prototyping area, changing any connections to the board, or removing the RCM4000 from the Prototyping Board.

2.3 Run a Sample Program

Once the RCM4000/RCM4010/RCM4050 is connected as described in the preceding pages, start Dynamic C by double-clicking on the Dynamic C icon on your desktop or in your **Start** menu.

If you are using a USB port to connect your computer to the RCM4000/RCM4010/RCM4050, click on the "Communications" tab and verify that **Use USB to Serial Converter** is selected to support the USB programming cable. Click **OK**. You may have to determine which COM port was assigned to the RS-232/USB converter. Open **Control Panel > System > Hardware > Device Manager > Ports** and identify which COM port is used for the USB connection. In Dynamic C, select **Options > Project Options**, then select this COM port on the **Communications** tab, then click **OK**. You may type the COM port number followed by **Enter** on your computer keyboard if the COM port number is outside the range on the dropdown menu.

Now find the file **`PONG.C`**, which is in the Dynamic C **`SAMPLES`** folder. To run the program, open it with the **File** menu, compile it using the **Compile** menu, and then run it by selecting **Run** in the **Run** menu. The **STDIO** window will open on your PC and will display a small square bouncing around in a box.

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 a sample program, it is possible that your PC cannot handle the higher program-loading baud rate. Try changing the maximum download rate to a slower baud rate as follows.

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

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

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

If you receive the message **`No Rabbit Processor Detected`**, the programming cable may be connected to the wrong COM port, a connection may be faulty, or the target system may not be powered up. First, check to see that the power LED on the Prototyping Board is lit and that the jumper across pins 5–6 of header JP10 on the Prototyping Board is installed. If the LED is lit, check both ends of the programming cable to ensure that it is firmly plugged into the PC and the programming header on the RCM4000 with the marked (colored) edge of the programming cable towards pin 1 of the programming header. Ensure that the module is firmly and correctly installed in its connector on the Prototyping Board.

If there are no faults with the hardware, select a different COM port within Dynamic C as explained for the USB port above. 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 for another available COM port. You should receive a **`Bios compiled successfully`** message once this step is completed successfully.

2.4 Where Do I Go From Here?

If the sample program ran fine, you are now ready to go on to the sample programs in Chapter 3 and to develop your own applications. The sample programs can be easily modified for your own use. The user's manual also provides complete hardware reference information and software function calls for the RCM4000 series of modules and the Prototyping Board.

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

2.4.1 Technical Support

> **NOTE:** If you purchased your RCM4000/RCM4010/RCM4050 through a distributor or through a Rabbit partner, contact the distributor or partner first for technical support.

If there are any problems at this point:

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

3. Running Sample Programs

To develop and debug programs for the RCM4000 (and for all other Rabbit hardware), you must install and use Dynamic C. This chapter provides a tour of its major features with respect to the RCM4000.

3.1 Introduction

To help familiarize you with the RCM4000 modules, Dynamic C includes several sample programs. Loading, executing and studying these programs will give you a solid hands-on overview of the RCM4000'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 ANSI C. 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 module must be plugged in to the Prototyping Board as described in Chapter 2, "Getting Started."
2. Dynamic C must be installed and running on your PC.
3. The programming cable must connect the programming header on the module to your PC.
4. Power must be applied to the module 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 (if it is not still open), then compile and run it by pressing **F9**.

Each sample program has comments that describe the purpose and function of the program. Follow the instructions at the beginning of the sample program.

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

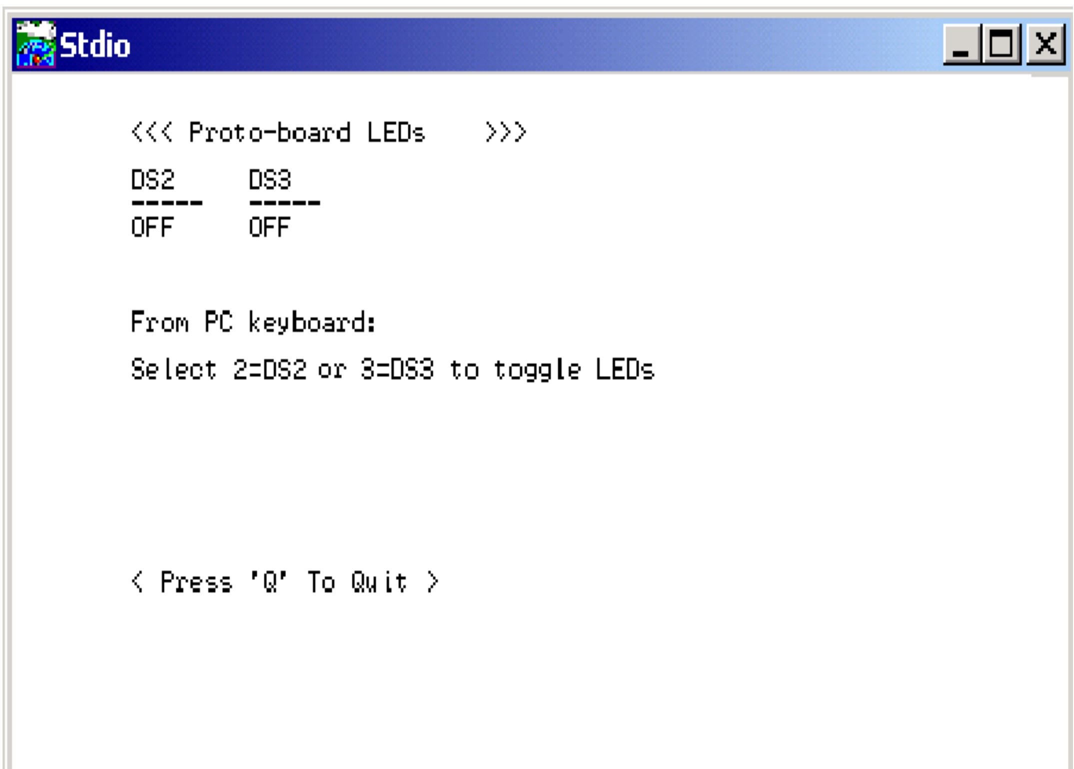
3.2 Sample Programs

Of the many sample programs included with Dynamic C, several are specific to the RCM4000 modules. These programs will be found in the **`SAMPLES\RCM4000`** folder.

- **`CONTROLLED.C`**—Demonstrates use of the digital outputs by having you turn LEDs DS2 and DS3 on the Prototyping Board on or off from the **STDIO** window on your PC.

  Parallel Port B bit 2 = LED DS2
  Parallel Port B bit 3 = LED DS3

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



  Press “2” or “3” on your keyboard to select LED DS2 or DS3 on the Prototyping Board. Then follow the prompt in the Dynamic C **STDIO** window to turn the LED ON or OFF. A logic low will light up the LED you selected.

- **`FLASHLED1.C`**—demonstrates the use of assembly language to flash LEDs DS2 and DS3 on the Prototyping Board at different rates. Once you have compiled and run this program, LEDs DS2 and DS3 will flash on/off at different rates.

- **`FLASHLED2.C`**—demonstrates the use of cofunctions and costatements to flash LEDs DS2 and DS3 on the Prototyping Board at different rates. Once you have compiled and run this program, LEDs DS2 and DS3 will flash on/off at different rates.

- **`LOW_POWER.C`**—demonstrates how to implement a function in RAM to reduce power consumption by the Rabbit microprocessor. There are four features that lead to the lowest possible power draw by the microprocessor.

  1. Run the CPU from the 32 kHz crystal.
  2. Turn off the high-frequency crystal oscillator.
  3. Run from RAM.
  4. Ensure that internal I/O instructions do not use CS0.

  Once you are ready to compile and run this sample program, use **<Alt-F9>** instead of just **F9**. This will disable polling, which will allow Dynamic C to continue debugging once the target starts running off the 32 kHz oscillator.

  This sample program will toggle LEDs DS2 and DS3 on the Prototyping Board. You may use an oscilloscope. DS2 will blink the fastest. After switching to low power, both LEDs will blink together.

- **`TAMPERDETECTION.C`**—demonstrates how to detect an attempt to enter the bootstrap mode. When an attempt is detected, the battery-backed onchip-encryption RAM on the Rabbit 4000 is erased. This battery-backed onchip-encryption RAM can be useful to store data such as an AES encryption key from a remote location.

  This sample program shows how to load and read the battery-backed onchip-encryption RAM and how to enable a visual indicator.

  Once this sample is compiled running (you have pressed the **F9** key while the sample program is open), remove the programming cable and press the reset button on the Prototyping Board to reset the module. LEDs DS2 and DS3 will be flashing on and off.

  Now press switch S2 to load the battery-backed RAM with the encryption key. The LEDs are now on continuously. Notice that the LEDs will stay on even when you press the reset button on the Prototyping Board.

  Reconnect the programming cable briefly and unplug it again. The LEDs will be flashing because the battery-backed onchip-encryption RAM has been erased. Notice that the LEDs will continue flashing even when you press the reset button on the Prototyping Board.

  You may press switch S2 again and repeat the last steps to watch the LEDs.

- **`TOGGLESWITCH.C`**—demonstrates the use of costatements to detect switch presses using the press-and-release method of debouncing. LEDs DS2 and DS3 on the Prototyping Board are turned on and off when you press switches S2 and S3. S2 and S3 are controlled by PB4 and PB5 respectively.

Once you have loaded and executed these five programs and have an understanding of how Dynamic C and the RCM4000 modules interact, you can move on and try the other sample programs, or begin building your own.

3.2.1 Use of NAND Flash (RCM4000 only)

The following sample programs can be found in the **SAMPLES\RCM4000\NANDFlash** folder.

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

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

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

- **NFLASH_INSPECT.c**—This program is a utility for inspecting the contents of a NAND flash chip. When the sample program starts running, it attempts to communicate with the NAND flash chip selected by the user. Once a NAND flash chip is found, the user can execute various commands to print out the contents of a specified page, clear (set to zero) all the bytes in a specified page, erase (set to FF), or write to specified pages.

  CAUTION: When you run this sample program, enabling the **#define NFLASH_CANERASEBADBLOCKS** macro makes it possible to write to bad blocks.

- **NFLASH_LOG.c**—This program runs a simple Web server and stores a log of hits in the NAND flash.

  This log can be viewed and cleared from a browser by connecting the RJ-45 jack on the RCM4000 to your PC as described in Section 6.1. The sidebar on the next page explains how to set up your PC or notebook to view this log.

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

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

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

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

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

   IP Address : 10.10.6.101

   Netmask : 255.255.255.0

   Default gateway : 10.10.6.1

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

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

http://10.10.6.100

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

This sample program does not exhibit ideal behavior in its method of writing to the NAND flash. However, the inefficiency attributable to the small amount of data written in each append operation is offset somewhat by the expected relative infrequency of these writes, and by the sample program's method of "walking" through the flash blocks when appending data as well as when a log is cleared.

- `NFLASH_ERASE.c`—This program is a utility to erase all the good blocks on a NAND flash chip. When the program starts running, it attempts to establish communication with the NAND flash chip selected by the user. If the communication is successful, the progress in erasing the blocks is displayed in the Dynamic C **STDIO** window as the blocks are erased.