



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!



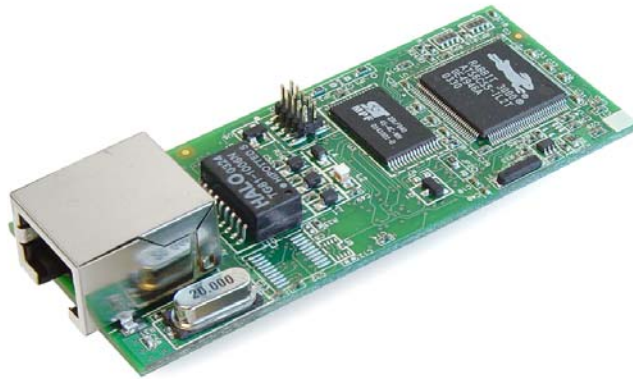
Contact us

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 RCM3700

C-Programmable Core Module
with Ethernet, Serial Flash, and Enhanced Software

User's Manual

019-0136_L

RabbitCore RCM3700 User's Manual

Part Number 019-0136_L • Printed in U.S.A.

©2003–2010 Digi International Inc. • All rights reserved.

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

Trademarks

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

Rabbit 3000 is a trademark of Digi International Inc.

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

Rabbit Semiconductor Inc.

www.rabbit.com

TABLE OF CONTENTS

Chapter 1. Introduction	4
1.1 RCM3700 Features	4
1.2 Advantages of the RCM3700	6
1.3 Development and Evaluation Tools.....	7
1.3.1 Development Kit	7
1.3.2 Software	8
1.3.3 Application Kits	8
1.3.4 Online Documentation	8
Chapter 2. Getting Started	9
2.1 Step 1 — Install Dynamic C	9
2.2 Hardware Connections.....	10
2.2.1 Step 1 — Attach Module to Prototyping Board.....	10
2.2.2 Step 2 — Connect Programming Cable.....	11
2.2.3 Step 3 — Connect Power	12
2.2.3.1 Overseas Development Kits	12
2.3 Starting Dynamic C	13
2.4 Run a Sample Program	13
2.4.1 Troubleshooting	13
2.5 Where Do I Go From Here?	14
2.5.1 Technical Support	14
Chapter 3. Running Sample Programs	15
3.1 Introduction.....	15
3.2 Sample Programs	17
3.2.1 Use of Serial Flash	19
3.2.2 Serial Communication.....	19
3.2.3 A/D Converter Inputs.....	22
Chapter 4. Hardware Reference	25
4.1 RCM3700 Digital Inputs and Outputs	26
4.1.1 Memory I/O Interface	30
4.1.2 Other Inputs and Outputs	30
4.2 Serial Communication	31
4.2.1 Serial Ports	31
4.2.2 Ethernet Port	32
4.2.3 Serial Programming Port.....	33
4.3 Serial Programming Cable.....	34
4.3.1 Changing Between Program Mode and Run Mode	34
4.3.2 Standalone Operation of the RCM3700.....	35
4.4 Other Hardware.....	36
4.4.1 Clock Doubler	36
4.4.2 Spectrum Spreader	36

4.5 Memory.....	37
4.5.1 SRAM	37
4.5.2 Flash EPROM	37
4.5.3 Serial Flash.....	37
4.5.4 Dynamic C BIOS Source Files	37
Chapter 5. Software Reference	38
5.1 More About Dynamic C	38
5.2 Dynamic C Functions	40
5.2.1 Board Initialization	41
5.2.2 Analog Inputs.....	42
5.2.3 Digital I/O	58
5.2.4 Serial Communication Drivers.....	59
5.2.5 Serial Flash.....	59
5.2.6 TCP/IP Drivers.....	59
5.3 Upgrading Dynamic C	60
5.3.1 Extras	60
5.3.1.1 Featured Application Kit.....	60
Chapter 6. Using the TCP/IP Features	61
6.1 TCP/IP Connections	61
6.2 TCP/IP Primer on IP Addresses.....	63
6.2.1 IP Addresses Explained	65
6.2.2 How IP Addresses are Used.....	66
6.2.3 Dynamically Assigned Internet Addresses	67
6.3 Placing Your Device on the Network	68
6.4 Running TCP/IP Sample Programs	69
6.4.1 How to Set IP Addresses in the Sample Programs	70
6.4.2 How to Set Up your Computer for Direct Connect	71
6.5 Run the PINGME.C Sample Program	72
6.6 Running Additional Sample Programs With Direct Connect	72
6.6.1 RabbitWeb Sample Programs	73
6.6.2 Secure Sockets Layer (SSL) Sample Programs	74
6.6.3 Dynamic C FAT File System, RabbitWeb, and SSL Modules	74
6.7 Where Do I Go From Here?	76
Appendix A. RCM3700 Specifications	77
A.1 Electrical and Mechanical Characteristics	78
A.1.1 Headers.....	81
A.2 Bus Loading.....	82
A.3 Rabbit 3000 DC Characteristics.....	85
A.4 I/O Buffer Sourcing and Sinking Limit	86
A.5 Conformal Coating.....	87
A.6 Jumper Configurations.....	88
Appendix B. Prototyping Board	90
B.1 RCM3700 Prototyping Board	91
B.1.1 Features	92
B.1.2 Mechanical Dimensions and Layout	94
B.1.3 Power Supply	95
B.1.4 Using the RCM3700 Prototyping Board	96
B.1.4.1 Adding Other Components	97
B.1.5 Analog Features.....	98
B.1.5.1 A/D Converter Inputs	98
B.1.5.2 Thermistor Input	100
B.1.5.3 Other A/D Converter Features.....	101
B.1.5.4 A/D Converter Calibration	102

B.1.6 Serial Communication.....	103
B.1.6.1 RS-232.....	104
B.1.6.2 RS-485.....	105
B.1.7 Other Prototyping Board Modules.....	107
B.1.8 Jumper Configurations.....	108
B.1.9 Use of Rabbit 3000 Parallel Ports.....	110
B.2 RCM3720 Prototyping Board.....	112
B.2.1 Features.....	113
B.2.2 Mechanical Dimensions and Layout.....	114
B.2.3 Power Supply.....	115
B.2.4 Using the RCM3720 Prototyping Board.....	116
B.2.4.1 Prototyping Area.....	118
B.2.5 Serial Communication.....	119
B.2.6 Use of Rabbit 3000 Parallel Ports.....	121
Appendix C. LCD/Keypad Module	123
C.1 Specifications.....	123
C.2 Contrast Adjustments for All Boards.....	125
C.3 Keypad Labeling.....	126
C.4 Header Pinouts.....	127
C.4.1 I/O Address Assignments.....	127
C.5 Install Connectors on Prototyping Board.....	128
C.6 Mounting LCD/Keypad Module on the Prototyping Board.....	129
C.7 Bezel-Mount Installation.....	130
C.7.1 Connect the LCD/Keypad Module to Your Prototyping Board.....	132
C.8 Sample Programs.....	133
C.9 LCD/Keypad Module Function Calls.....	134
C.9.1 LCD/Keypad Module Initialization.....	134
C.9.2 LEDs.....	134
C.9.3 LCD Display.....	135
C.9.4 Keypad.....	155
Appendix D. Power Supply	158
D.1 Power Supplies.....	158
D.1.1 Battery Backup.....	159
D.1.2 Battery-Backup Circuit.....	160
D.1.3 Reset Generator.....	160
Appendix E. Secure Embedded Web Application Kit	161
E.1 Sample Programs.....	162
E.2 Module Documentation.....	162
Index	163
Schematics	167

1. INTRODUCTION

The RCM3700 is a compact module that incorporates the latest revision of the powerful Rabbit® 3000 microprocessor, flash memory, onboard serial flash, static RAM, and digital I/O ports.

Throughout this manual, the term RCM3700 refers to the complete series of RCM3700 RabbitCore modules unless other production models are referred to specifically.

The RCM3700 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 RCM3700 receives its +5 V power from the customer-supplied motherboard on which it is mounted. The RCM3700 can interface with all kinds of CMOS-compatible digital devices through the motherboard.

The Development Kit and the Ethernet Connection Kit have what you need to design your own microprocessor-based system: a complete Dynamic C software development system with optional modules and a Prototyping Board that acts as a motherboard to allow you to evaluate the RCM3700 and to prototype circuits that interface to the RCM3700 module.

1.1 RCM3700 Features

- Small size: 1.20" x 2.95" x 0.98"
(30 mm x 75 mm x 25 mm)
- Microprocessor: latest revision of Rabbit 3000 running at 22.1 MHz supports Dynamic C Secure Sockets Layer (SSL) module for added security
- 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 and 512K SRAM (options for 256K flash memory and 128K SRAM)

- 1 Mbyte serial flash memory, which is required to run the optional Dynamic C FAT file system
- Real-time clock
- Watchdog supervisor
- Provision for customer-supplied backup battery via connections on header J1
- 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: 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 microprocessor 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 three RCM3700 production models. Table 1 below summarizes their main features.

Table 1. RCM3700 Features

Feature	RCM3700	RCM3710	RCM3720
Microprocessor	Rabbit 3000 [®] running at 22.1 MHz		
Flash Memory	512K	256K	512K
SRAM	512K	128K	256K
Serial Flash Memory	1 Mbyte		
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 RCM3700 is programmed over a standard PC serial port through a programming cable supplied with the Development Kit or the Ethernet Connection Kit, and can also be programmed through a USB port with an RS-232/USB converter or over an Ethernet with the RabbitLink (both available from Rabbit).

Mounting holes were introduced on opposite corners of the RCM3700 in 2009.

Appendix A provides detailed specifications for the RCM3700.

1.2 Advantages of the RCM3700

- 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.
- Integrated Ethernet port for network connectivity, with royalty-free TCP/IP software.
- Ideal for network-enabling security and access systems, home automation, HVAC systems, and industrial controls

1.3 Development and Evaluation Tools

1.3.1 Development Kit

The Development Kit contains the hardware and software needed to use the RCM3700.

- RCM3700 module.
- RCM3700 Prototyping Board.
- Universal AC adapter, 12 V DC, 1 A (includes Canada/Japan/U.S., Australia/N.Z., U.K., and European style plugs).
- Programming cable with 10-pin header and DE9 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.

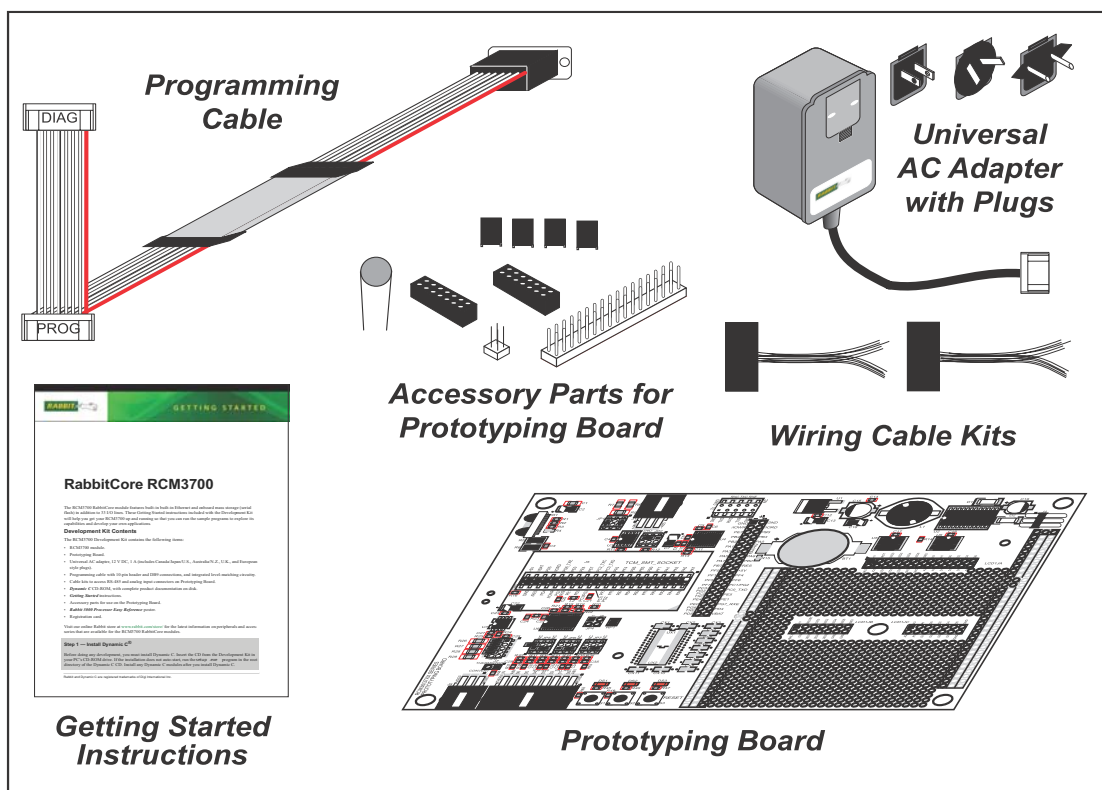


Figure 1. RCM3700 Development Kit

1.3.2 Software

The RCM3700 is programmed using version 8.11 or later of Dynamic C. Dynamic C v. 9.60 includes the popular μ C/OS-II real-time operating system, point-to-point protocol (PPP), FAT file system, RabbitWeb, and other select libraries that were previously sold as individual Dynamic C modules.

Rabbit also offers for purchase the Rabbit Embedded Security Pack featuring the Secure Sockets Layer (SSL) and a specific Advanced Encryption Standard (AES) library. In addition to the Web-based technical support included at no extra charge, a one-year telephone-based technical support subscription is also available for purchase. Visit our Web site at www.rabbit.com for further information and complete documentation, or contact your Rabbit sales representative or authorized distributor.

1.3.3 Application Kits

Rabbit also has application kits featuring the RCM3700 to provide the exact software and other tools that will enable to tailor your RCM3700 for specific applications.

- Secure Embedded Web Application Kit (Rabbit Part No. 101-0898)—comes with three CD-ROMs that have the Dynamic C RabbitWeb, FAT File System, and Secure Sockets Layer (SSL) modules, and includes Dynamic C 8.51 or a later version and an RCM3700. This enhanced software bundle facilitates the rapid development of secure Web browser interfaces for embedded system control. Appendix E provides additional information about the Secure Embedded Web Application Kit.
- Ethernet Connection Kit (Rabbit Part No. 101-0964)—comes with one CD-ROM that includes Dynamic C 9.01 or a later version, an RCM3720 module, and an RCM3720 Prototyping Board. This kit is intended to demonstrate and help you develop Ethernet-based applications.

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.

Each Dynamic C module has complete documentation available with the online documentation described above.

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 RCM3700 hardware in more detail, and explains how to set up and use the accompanying Prototyping Board.

NOTE: It is assumed that you have the RCM3700 Development Kit. If you purchased an RCM3700 module by itself, you will have to adapt the information in this chapter and elsewhere to your test and development setup.

2.1 Step 1 — Install Dynamic C

To develop and debug programs for the RCM3700 (and for all other Rabbit 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 RCM3700 Development Kit in your PC's CD-ROM drive. If autorun is enabled, the CD installation will begin automatically.

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

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

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

1. Attach the RCM3700 module to the Prototyping Board.
2. Connect the programming cable between the RCM3700 and the COM port on the workstation PC.
3. Connect the power supply to the Prototyping Board.

The connections are shown for the RCM3700 Prototyping Board, and are similar for the RCM3720 Prototyping Board.

2.2.1 Step 1 — Attach Module to Prototyping Board

Turn the RCM3700 module so that the Ethernet jack is on the left as shown in Figure 2 below. Insert the module's J1 header into the TCM_SMT_SOCKET socket on the Prototyping Board. The shaded corner notch at the bottom right corner of the RCM3700 module should face the same direction as the corresponding notch below it on the Prototyping Board.

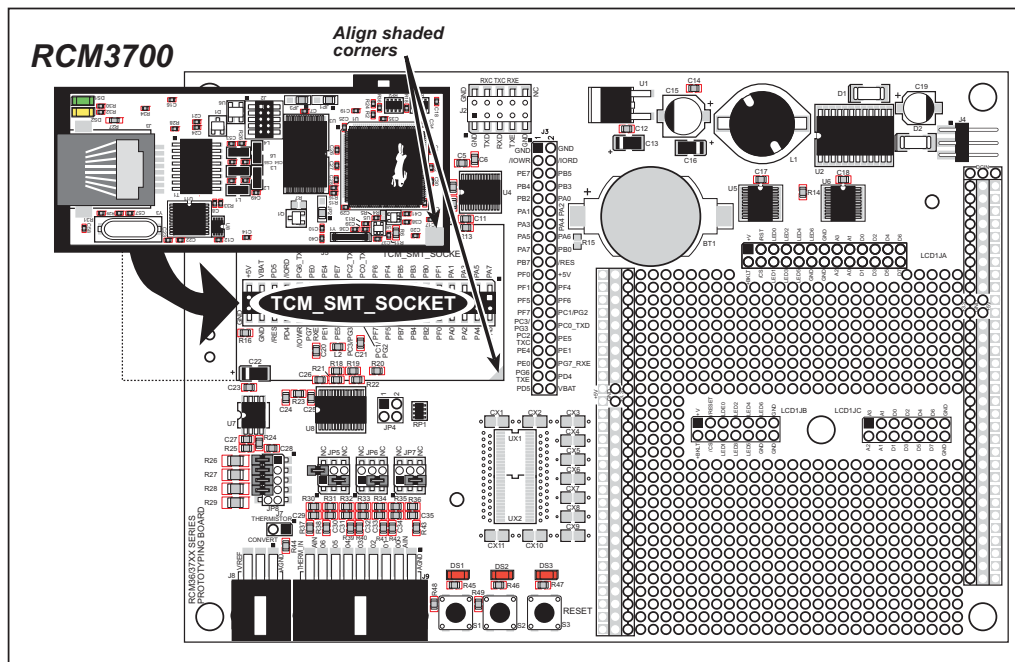


Figure 2. Install the RCM3700 Series on the Prototyping Board

NOTE: It is important that you line up the pins on header J1 of the RCM3700 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 Step 2 — Connect Programming Cable

The programming cable connects the RCM3700 to the PC running Dynamic C to download programs and to monitor the RCM3700 module during debugging.

Connect the 10-pin connector of the programming cable labeled **PROG** to header J2 on the RCM3700 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.)

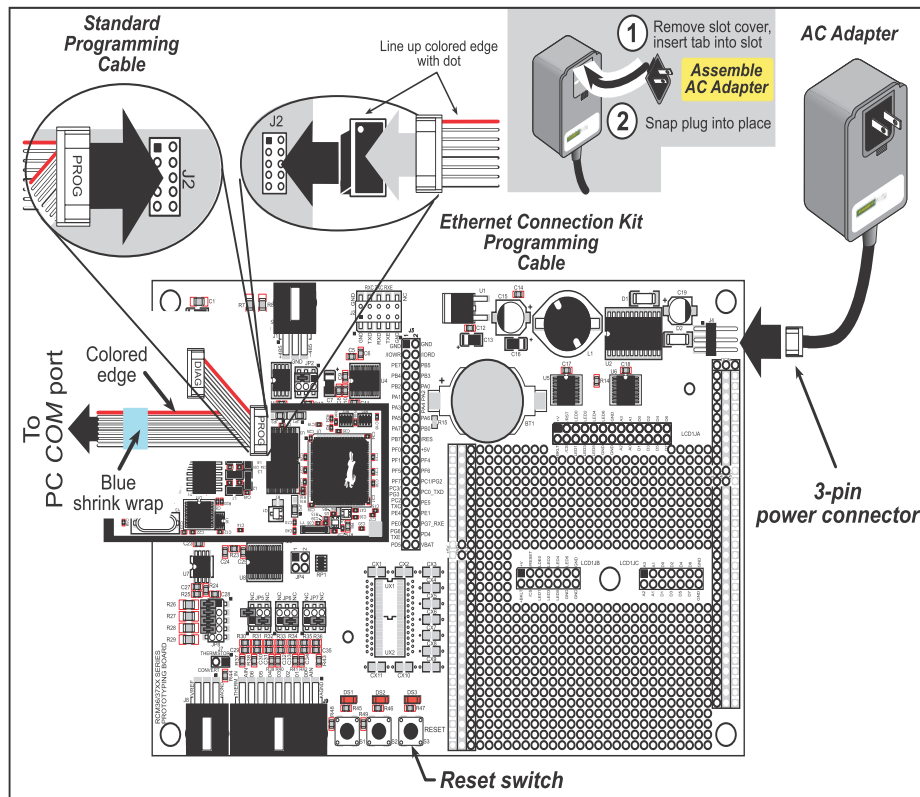


Figure 3. Connect Programming Cable and Power Supply

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

NOTE: Be sure to use the programming cable (part number 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. The simplified programming cable and adapter board that are supplied with the Ethernet Connection Kit may also be used as shown in the inset diagram above. Programming cables from other Rabbit kits are not designed to work with RCM3700 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. 20-151-0178) with the programming cables mentioned above. Note that not all RS-232/USB converters work with Dynamic C.

2.2.3 Step 3 — Connect Power

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

First, prepare the AC adapter for the country where it will be used by selecting the plug. The RCM3700 Development Kit presently includes Canada/Japan/U.S., Australia/N.Z., U.K., and European style plugs. Snap in the top of the plug assembly into the slot at the top of the AC adapter as shown in Figure 3, then press down on the spring-loaded clip below the plug assembly to allow the plug assembly to click into place.

Connect the AC adapter to 3-pin header 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 AC adapter. The LED above the **RESET** button on the Prototyping Board should light up. The RCM3700 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.

To power down the Prototyping Board, unplug the power connector from J4. You should disconnect power before making any circuit adjustments in the prototyping area, changing any connections to the board, or removing the RCM3700 from the Prototyping Board.

2.2.3.1 Overseas Development Kits

Development kits sold outside North America before 2009 included a header connector that could be connected to 3-pin header J4 on the Prototyping Board. The connector could be attached either way as long as it was not offset to one side. The red and black wires from the connector could 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 RCM3700 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 RCM3700, 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. Click **OK**.

2.4 Run a Sample Program

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

This program shows that the CPU is working.

2.4.1 Troubleshooting

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

- Check that the RCM3700 is powered correctly — the red power lamp on the Prototyping Board should be lit when the RCM3700 is mounted on the Prototyping Board and the AC adapter is plugged in.
- Check both ends of the programming cable to ensure that they are firmly plugged into the PC and the **PROG** connector, not the **DIAG** connector, is plugged in to the programming port on the RCM3700 with the marked (colored) edge of the programming cable towards pin 1 of the programming header.
- Ensure that the RCM3700 module is firmly and correctly installed in its connectors on the Prototyping Board.
- Dynamic C uses the COM port specified during installation. 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 COM port used by the programming cable.

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.

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 *RCM3700 User's Manual* also provides complete hardware reference information and describes the software function calls for the RCM3700, the Prototyping Board, and the optional LCD/keypad module.

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 RCM3700 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 RCM3700 (and for all other Rabbit hardware), you must install and use Dynamic C.

3.1 Introduction

To help familiarize you with the RCM3700 modules, Dynamic C includes several sample programs. Loading, executing and studying these programs will give you a solid hands-on overview of the RCM3700'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 RCM3700 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 RCM3700 to your PC.
4. Power must be applied to the RCM3700 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**. The RCM3700 must be connected to a PC using the programming cable.

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

The default I/O configuration in the sample programs is based on the RabbitCore module detected during compile time:

- Any RCM3700 RabbitCore module (except the RCM3720) will have its I/O ports configured for an RCM3700 Prototyping Board.
- An RCM3720 RabbitCore module will have its I/O ports configured for an RCM3720 Prototyping Board.

You may override these default settings to run an RCM3720 RabbitCore module on the RCM3700 Prototyping Board or to run another RCM3700 RabbitCore module on the RCM3720 Prototyping Board by adding the following macro to the sample program you will be running.

- To run an RCM3720 RabbitCore module on an RCM3700 Prototyping Board, add the following macro at the top of the sample program you will be running.

```
#define RCM3700_PROTOBOARD
```

Sample programs that are specifically designed for the RCM3700 Prototyping Board already have this macro included.

- To run an RCM3700 RabbitCore module (other than the RCM3720) on an RCM3720 Prototyping Board, add the following macro at the top of the sample program you will be running.

```
#define RCM3720_PROTOBOARD
```

3.2 Sample Programs

Of the many sample programs included with Dynamic C, several are specific to the RCM3700. Sample programs illustrating the general operation of the RCM3700, serial communication, and the A/D converter on the Prototyping Board are provided in the **SAMPLES\RCM3700** and the **SAMPLES\RCM3720** folders as shown in the table below. The sample programs use the features available on the two Prototyping Boards.

Feature	RCM3700 Prototyping Board	RCM3720 Prototyping Board
Sample Program Folder	SAMPLES\RCM3700	SAMPLES\RCM3720
Digital I/O	×	×
IrDA Transceivers	×	
Serial Flash	×	×
Serial Communication	×	×
TCP/IP	×	×
A/D Converter	×	
LCD/Keypad Module	×	
Dynamic C FAT File System, RabbitWeb, SSL Modules	×	×

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 RCM3700 must be installed on the Prototyping Board when using these sample programs.

TCP/IP sample programs are described in Chapter 6, “Using the TCP/IP Features.” Sample programs for the optional LCD/keypad module that is used on the RCM3700 Prototyping Board are described in Appendix C.

Additional sample programs are available online at www.rabbit.com/support/downloads/downloads_prod.shtml.

- **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.

If you are using the RCM3700 Prototyping Board, install a 2 x 2 header at JP4 and connect pins 1–2 and pins 3–4 on header JP4 before running this sample program.

- **FLASHLED.c**—Demonstrates assembly-language program by flashing LEDs DS1 and DS2 on the Prototyping Board at different rates.
- **TOGGLESWITCH.c**—Uses costatements to detect switches using debouncing. The corresponding LEDs (DS1 and DS2) will turn on or off.

- **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 LEDs >>>
DS2   DS3   DS4   DS5
---   ---   ---   ---
OFF   OFF   OFF   OFF

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

< Press 'Q' To Quit >

```

Press “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.

- **IR_DEMO . c**—Demonstrates sending Modbus ASCII packets between two RCM3700 Prototyping Board assemblies with IrDA transceivers via the IrDA transceivers. Note that this sample program will only work with the RCM3700 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 RCM3700 and compile and run the same sample program. With the programming cable still connected to the second Prototyping Board assembly, press switch S1 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 five programs and have an understanding of how Dynamic C and the RCM3700 modules interact, you can move on and try the other sample programs, or begin building your own.

3.2.1 Use of Serial Flash

The following sample programs can be found in the **SAMPLES\RCM3700\SerialFlash** and the **SAMPLES\RCM3720\SerialFlash** folders.

- **SERIAL_FLASHLOG.C**—This program runs a simple Web server and stores a log of hits on the home page of the serial flash “server.” This log can be viewed and cleared from a browser at <http://10.10.6.100/>. You may need to first “configure” your PC for a “10Base-T Half-Duplex” or an “Auto-Negotiation” connection from the “Advanced” tab, which is accessed from the control panel (**Start > Settings > Control Panel**) by choosing **Network Connections**.
- **SFLASH_INSPECT.C**—This program is a handy utility for inspecting the contents of a serialflash chip. When the sample program starts running, it attempts to initialize a serial flash chip on Serial Port B. Once a serial flash chip is found, the user can perform two different commands to either print out the contents of a specified page or clear (set to zero) all the bytes in a specified page.

3.2.2 Serial Communication

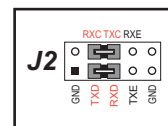
The following sample programs can be found in the **SAMPLES\RCM3700\SERIAL** and the **SAMPLES\RCM3720\SERIAL** folders.

NOTE: PE5 is set up to enable/disable the RS-232 chip on the RCM3700 Prototyping Board. This pin will also be toggled when you run RS-232 sample programs on an RCM3700 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 hardware flow control by configuring Serial Port C for CTS/RTS with serial data coming from Serial Port D. The serial data received are displayed in the **STDIO** window.

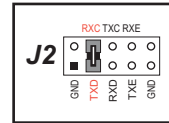
To set up the Prototyping Board, you will need to tie TxC and RxC together on the RS-232 header at J2, and you will also tie TxD and RxD together using the jumpers supplied in the Development Kit as shown in the diagram.



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.

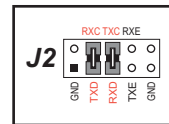
- **PARITY.C**—This program demonstrates the use of parity modes by repeatedly sending byte values 0–127 from Serial Port D to Serial Port C. The program will switch between generating parity or not on Serial Port D. Serial Port C will always be checking parity, so parity errors should occur during every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

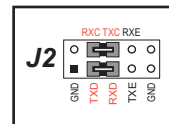
- **SIMPLE3WIRE.C**—This program demonstrates basic RS-232 serial communication. Lower case characters are sent by TxC, and are received by RxD. The characters are converted to upper case and are sent out by TxD, are received by RxC, and are displayed in the Dynamic C **STDIO** window.



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.

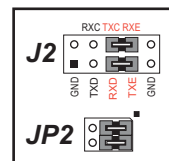
To set up the Prototyping Board, you will need to tie TxD and RxD together on the RS-232 header at J2, and you will also tie TxC and RxC together using the 0.1" jumpers supplied in the Development Kit as shown in the diagram.



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 on the Prototyping Board 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 RCM3700 Prototyping Board.

- **SIMPLE485MASTER.C**—This program demonstrates a simple RS-485 transmission of lower case letters to a slave RCM3700. The slave will send back converted upper case letters back to the master RCM3700 and display them in the **STDIO** window. Use **SIMPLE485SLAVE.C** to program the slave RCM3700, 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 RCM3700. The slave will send back converted upper case letters back to the master RCM3700 and display them in the **STDIO** window. Use **SIMPLE485MASTER.C** to program the master RCM3700, 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.



3.2.3 A/D Converter Inputs

The following sample programs are found in the **SAMPLES\RCM3700\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 use this sample program as a template for your own program, be aware that PE4 is also used for the IrDA FIR_SEL on the Prototyping Board. You will need to use another parallel port line for the analog input if you are also using the IrDA transceiver.

- **THERMISTOR.C**—Demonstrates how to use analog input THERM_IN7 to calculate temperature for display to the **STDIO** 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 **UPLoadCALIB.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.

You will need to run a serial utility such as Tera Term on your PC. You may download Tera Term from hp.vector.co.jp/authors/VA002416/teraterm.html. 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.