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.

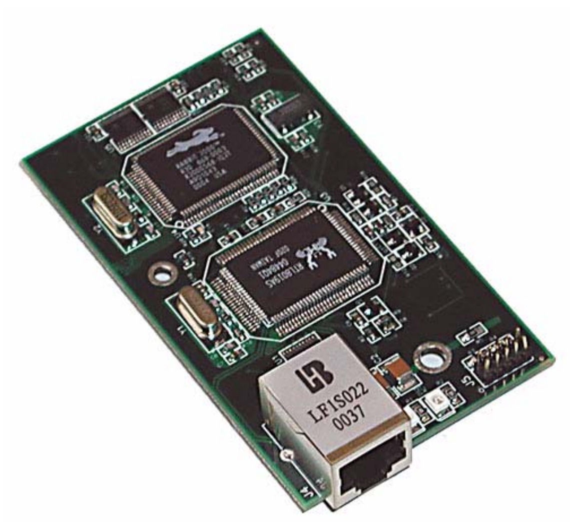
With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC 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

PRODUCT MANUAL



RabbitCore RCM2100

C-Programmable Module with Ethernet

User's Manual

019–0091 • 070831–K

RabbitCore RCM2100 User's Manual

Part Number 019-0091 • 070831–K • Printed in U.S.A.

Trademarks

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

Rabbit Semiconductor Inc.

www.rabbit.com

TABLE OF CONTENTS

Appendix C. Power Supply 81

Appendix D. Sample Circuits 85

Index 91

Schematics 93

1. Introduction

The RabbitCore RCM2100 series is a family of microprocessor modules designed to be the heart of embedded control systems. In addition to the array of I/O and addressing available on other Rabbit Semiconductor products, the RCM2100 series offers an optional integrated Ethernet port. These modules permit LAN and Internet-enabled systems to be built as easily as serial communications-only systems.

Throughout this manual, the term RCM2100 refers to the complete series of RCM2100 RabbitCore modules unless other production models are referred to specifically.

The RCM2100 is a microprocessor core module designed to be the heart of your own controller built around the plug-in module. Data processing is done by a Rabbit 2000 microprocessor operating at 22.1 MHz.

The RCM2100 has a Rabbit 2000 microprocessor, a static RAM, up to two flash memory chips, two quartz crystals (main oscillator and timekeeping), and the circuitry necessary for reset and management of battery backup of the Rabbit 2000's internal real-time clock and the static RAM. Two 40-pin headers bring out the Rabbit 2000 I/O bus, address lines, data lines, parallel ports, and serial ports.

The RCM2100 receives its +5 V power from the user board on which it is mounted. The RCM2100 can interface with all kinds of CMOS-compatible digital devices through the user board.

1.1 RCM2100 Features

- Small size: 2.0" × 3.5" × 0.80"
  (51 mm × 89 mm × 20 mm)
- Microprocessor: Rabbit 2000 running at 22.1 MHz
- 34 CMOS-compatible parallel I/O lines grouped in five 8-bit ports (shared with serial ports)
- 8 data lines (BD0–BD7)
- 13 address lines (BA0–BA12)
- I/0 read, write, buffer enable
- Status, watchdog and clock outputs
- Two startup mode inputs for booting and master/slave configuration
- External reset input
- Reset output
- Five 8-bit timers, two 10-bit timers; five timers are cascadable in pairs
- 2 × 256K flash memory, 512K SRAM
- Real-time clock
- Watchdog supervisor
- Provision for customer-supplied backup battery via connections on header J2
- Four CMOS-compatible serial ports: maximum asynchronous baud rate of 690,625 bps, maximum synchronous baud rate of 5.52 Mbps. Two ports are configurable as clocked ports.

Appendix A, "RabbitCore RCM2100 Specifications," provides detailed specifications for the RabbitCore RCM2100 modules.

Four versions of the RabbitCore RCM2100 are available. Their standard features are summarized in Table 1.

Table 1. RCM2100 Production Models

Model	Features
RCM2100	Full-featured module including 10/100-compatible Ethernet port with 10Base-T interface
RCM2110	RCM2100 with 128K SRAM, 256K flash memory
RCM2120	RCM2100 without Ethernet
RCM2130	RCM2110 without Ethernet

1.2 Advantages of the RCM2100

- Fast time to market using a fully engineered, “ready to run” microprocessor core.
- Competitive pricing when compared with the alternative of purchasing and assembling individual components.
- Easy C-language program development and debugging, including 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 (on selected models) for network connectivity, royalty-free TCP/IP software.
- Models with and without Ethernet for flexible production options.
- Small size and identical footprint and pinout for all models.

1.3 Development and Evaluation Tools

A complete Development Kit, including a Prototyping Board, accessory components and Dynamic C development software, is available to accompany the RCM2100 module. The Development Kit puts together the essentials you need to design an embedded microprocessor-based system rapidly and efficiently.

1.3.1 Development Software

The RCM2100 modules use the Dynamic C development environment for rapid creation and debugging of runtime applications. Dynamic C provides a complete development environment with integrated editor, compiler and source-level debugger. It interfaces directly with the target system, eliminating the need for complex and unreliable in-circuit emulators.

> **NOTE:** The RCM2100 modules require Dynamic C v7.04 or later for development. A compatible version is included on the Development Kit CD-ROM.

1.3.2 Development Kit Contents

The RCM2100 Development Kit contains the following items:

- RCM2100 module with 10Base-T Ethernet port, 512K flash memory and 512K SRAM.
- RCM2100 Prototyping Board with accessory hardware and components.
- Wall transformer power supply, 12 V DC, 1 A. (Included only with Development Kits sold for the North American market. Overseas users will have to substitute a power supply compatible with their local mains power.)
- 10-pin header to DB9 programming cable with integrated level-matching circuitry.
- ***Dynamic C*** CD-ROM, with complete product documentation on disk.
- ***Getting Started*** instructions.
- Registration card.

1.4 How to Use This Manual

This user's manual is intended to give users detailed information on the RCM2100 modules. It does not contain detailed information on the Dynamic C development environment or the TCP/IP software support for the integrated Ethernet port. Most users will want more detailed information on some or all of these topics in order to put the RCM2100 module to effective use.

1.4.1 Additional Product Information

In addition to the product-specific information contained in the ***RabbitCore RCM2100 User's Manual***, several higher level reference manuals are provided in HTML and PDF form on the accompanying CD-ROM. Advanced users will find these references valuable in developing systems based on the RCM2100 modules:

- ***Dynamic C User's Manual***
- ***An Introduction to TCP/IP***
- ***Dynamic C TCP/IP User's Manual***
- ***Rabbit 2000 Microprocessor User's Manual***

1.4.2 Using Online Documentation

We provide the bulk of our user and reference documentation in two electronic formats, HTML and Adobe PDF. We do this for several reasons.

We believe that providing all users with our complete library of product and reference manuals is a useful convenience. However, printed manuals are expensive to print, stock, and ship. Rather than include and charge for manuals that every user may not want, or provide only product-specific manuals, we choose to provide our complete documentation and reference library in electronic form with every Development Kit and with our Dynamic C development environment.

> **NOTE:** The most current version of Adobe Acrobat Reader can always be downloaded from Adobe's web site at **http://www.adobe.com**. We recommend that you use version 5.0 or later.

Providing this documentation in electronic form saves an enormous amount of paper by not printing copies of manuals that users don't need. It reduces the number of outdated manuals we have to discard from stock as well, and it makes providing a complete library of manuals an almost cost-free option. For one-time or infrequent reference, electronic documents are more convenient than printed ones—after all, they aren't taking up shelf or desk space!

Finding Online Documents

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.

Printing Electronic Manuals

We recognize that many users prefer printed manuals for some uses. Users can easily print all or parts of those manuals provided in electronic form. The following guidelines may be helpful:

- Print from the Adobe PDF versions of the files, not the HTML versions.
- Print only the sections you will need to refer to more than once.
- Print manuals overnight, when appropriate, to keep from tying up shared resources during the work day.
- If your printer supports duplex printing, print pages double-sided to save paper and increase convenience.
- If you do not have a suitable printer or do not want to print the manual yourself, most retail copy shops (e.g., Kinkos, AlphaGraphics, CopyMax) will print the manual from the PDF file and bind it for a reasonable charge—about what we would have to charge for a printed and bound manual.

2. Getting Started

This chapter describes the RCM2100 hardware in more detail, and explains how to set up and use the accompanying prototyping and development board.

NOTE: This chapter (and this manual) assume that you have the RabbitCore RCM2100 Development Kit. If you purchased an RCM2100 module by itself, you will have to adapt the information in this chapter and elsewhere to your test and development setup.

2.1 Connections

There are three steps to connecting the Prototyping Board for use with Dynamic C and the sample programs:

1. Attach the RCM2100 module to the Prototyping Board.
2. Connect the programming cable between the RCM2100 module and the workstation PC.
3. Connect the power supply to the Prototyping Board.

2.1.1 Attach Module to Prototyping Board

Turn the RCM2100 module so that the Ethernet connector is on the left, as shown in Figure 1 below. Align the module headers J1 and J2 on the bottom side of the RCM2100 into header sockets J1 and J3 on the Prototyping Board.

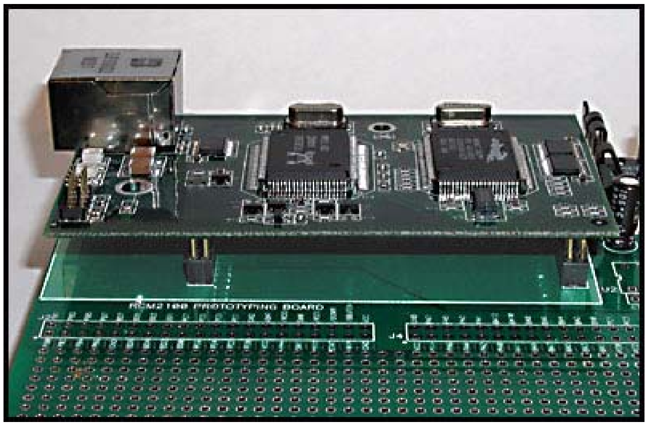


Figure 1. Installing the RCM2100 Module on the Prototyping Board. Note the orientation of the module.

NOTE: It is important that you line up the RCM2100 pins on headers J1 and J2 exactly with the corresponding pins of header sockets J1 and J3 on the Prototyping Board. The header pins may become bent or damaged if the pin alignment is offset, and the module will not work.

Press the module's pins firmly into the Prototyping Board header sockets. The installed module is shown in Figure 2 below.

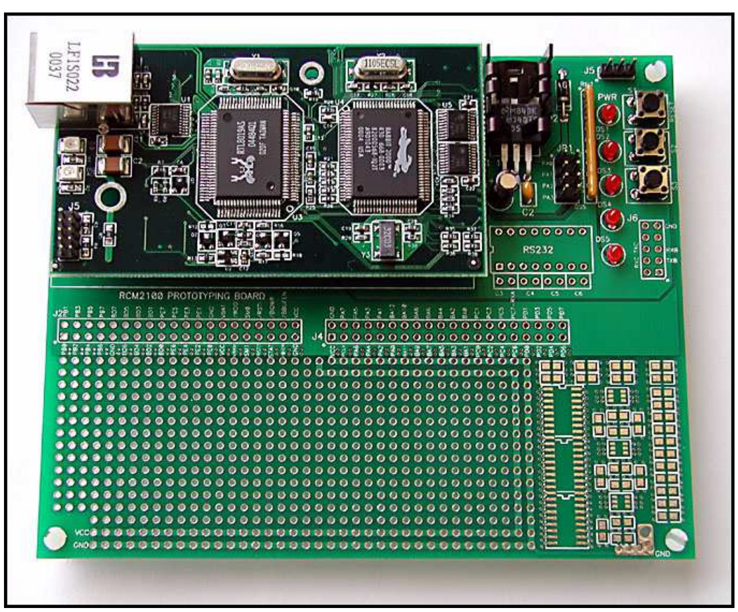


Figure 2. RCM2100 Installed and Seated on the Prototyping Board

2.1.2 Connect Programming Cable

The programming cable connects the RCM2100 module to the PC running Dynamic C, to download programs and to monitor the RCM2100 for debugging.

Connect the 10-pin connector of the programming cable labeled **PROG** to header J5 on the RCM2100 module as shown in Figure 3 below. Be sure to orient the 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.)

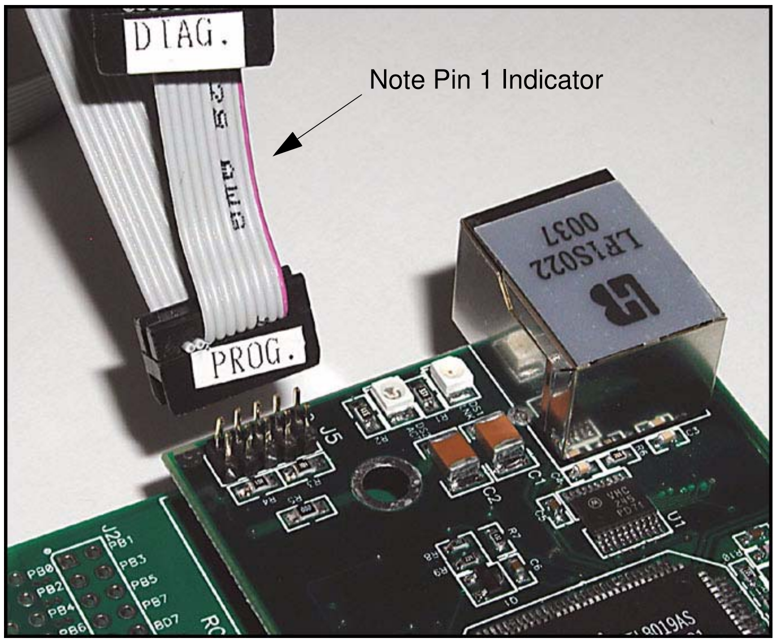


Figure 3. Attaching Programming Cable to the RCM2100

NOTE: The stripe on the cable is towards pin 1 of the header J5.

Connect the other end of the programming cable to a COM port on your PC. Make a note of the port to which you connect the cable, as Dynamic C needs to have this parameter configured when it is installed.

NOTE: COM 1 is the default port used by Dynamic C.

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 RCM2100 Development Kit. Note that not all RS-232/USB converters work with Dynamic C.

2.1.3 Connect Power

When all other connections have been made, you can connect power to the RCM2100 Prototyping Board.

Hook the connector from the wall transformer to header J5 on the Prototyping Board as shown in Figure 4 below. The connector may be attached either way as long as it is not offset to one side.

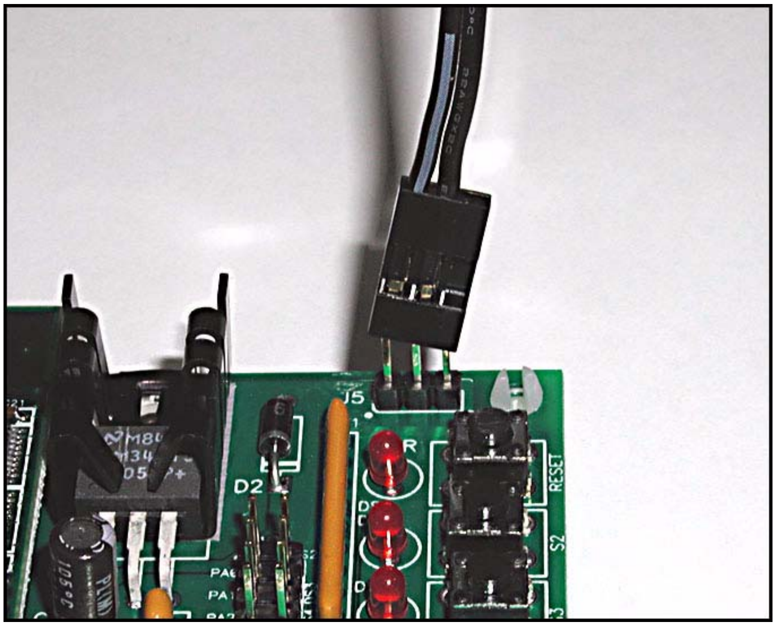


Figure 4. Power Supply Connections to Prototyping Board

Plug in the wall transformer. The power LED on the Prototyping Board should light up. The RCM2100 and the Prototyping Board are now ready to be used.

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

To power down the Prototyping Board, unplug the power connector from J5. You should disconnect power before making any circuit adjustments in the prototyping area, changing any connections to the board, or removing the RCM2100 module from the Prototyping Board.

2.2 Run a Sample Program

Once the RCM2100 is connected as described in the preceding pages, start Dynamic C by double-clicking on the Dynamic C icon or by double-clicking on **`dcrab_XXXX.exe`** in the Dynamic C root directory, where **`XXXX`** are version-specific characters. Dynamic C uses the serial port specified during installation.

If you are using a USB port to connect your computer to the RCM2100 module, choose **Options > Project Options** and select "Use USB to Serial Converter" under the **Communications** tab.

Find the file **`PONG.C`**, which is in the Dynamic C **`SAMPLES`** folder. To run the program, open it with the **File** menu (if it is not still open), then compile and run it by pressing **F9** or by selecting **Run** in the **Run** menu. The **STDIO** window will open and will display a small square bouncing around in a box.

2.2.1 Troubleshooting

If Dynamic C appears to compile the BIOS successfully, but you then receive a communication error message when you compile and load the sample program, it is possible that your PC cannot handle the higher program-loading baud rate. Try changing the maximum download rate to a slower baud rate as follows.

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

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

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

If there are any other problems:

- Check to make sure you are using the **PROG** connector, not the **DIAG** connector, on the programming cable.
- Check both ends of the programming cable to ensure that they are firmly plugged into the PC and the programming port on the RCM2100.
- Ensure that the RCM2100 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.

2.3 Where Do I Go From Here?

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

1. Run all of the sample programs described in Chapter 4 to get a basic familiarity with Dynamic C and the RabbitCore module's capabilities.
2. For further development, refer to the ***RabbitCore RCM2100 User's Manual*** for details of the module's hardware and software components.

   A documentation icon should have been installed on your workstation's desktop; click on it to reach the documentation menu. You can create a new desktop icon that points to **default.htm** in the **docs** folder in the Dynamic C installation folder.
3. For advanced development topics, refer to the ***Dynamic C User's Manual*** and the ***Dynamic C TCP/IP User's Manual***, also in the online documentation set.

2.3.1 Technical Support

> **NOTE:** If you purchased your RCM2100 through a distributor or through a Rabbit Semiconductor partner, contact the distributor or partner first for technical support.

If there are any problems at this point:

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

3. Running Sample Programs

To develop and debug programs for the RCM2100 (and for all other Rabbit Semiconductor hardware), you must install and use Dynamic C. Dynamic C is an integrated development system for writing embedded software. It runs on an IBM-compatible PC and is designed for use with Rabbit Semiconductor single-board computers and other single-board computers based on the Rabbit microprocessor. This chapter takes you through the installation of Dynamic C, and then provides a tour of the sample programs for the RCM2100.

3.1 Sample Programs

To help familiarize you with the RCM2100 modules, several sample Dynamic C programs have been included. Loading, executing and studying these programs will give you a solid hands-on overview of the RCM2100's capabilities, as well as a quick start with Dynamic C as an application development tool. These programs are intended to serve as tutorials, but then can also be used as starting points or building blocks for your own applications.

> **NOTE:** It is assumed in this section 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.

Each sample program has comments that describe the purpose and function of the program.

Before running any of these sample programs, make sure that your RCM2100 is connected to the Prototyping Board and to your PC as described in Section 2.1, "Connections." To run a sample program, open it with the **File** menu (if it is not already open), then compile and run it by pressing **F9** or by selecting **Run** in the **Run** menu.

Sample programs are provided in the Dynamic C `SAMPLES` folder. Two folders contain sample programs that illustrate features unique to the RabbitCore RCM2100.

- `RCM2100`—Demonstrates the basic operation and the Ethernet functionality of the RabbitCore RCM2100.
- `TCPIP`—Demonstrates more advanced TCP/IP programming for Rabbit Semiconductor's Ethernet-enabled Rabbit-based boards.

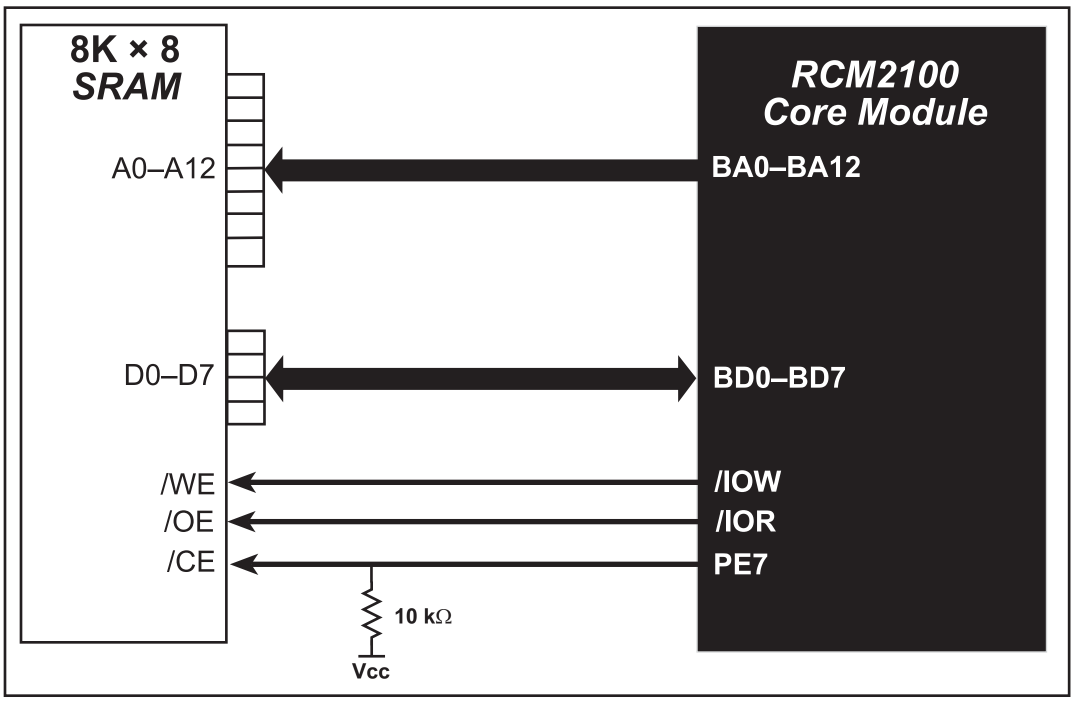
Complete information on Dynamic C is provided in the ***Dynamic C User's Manual***.

3.1.1 Getting to Know the RCM2100

The following sample programs can be found in the `SAMPLES\RCM2100` folder.

- `EXTSRAM.C`—demonstrates the setup and simple addressing to an external SRAM. This program first maps the external SRAM to the I/O Bank 7 register with a maximum of 15 wait states, chip select strobe (PE7), and allows writes. The first 256 bytes of SRAM are cleared and read back. Values are then written to the same area and are read back. The Dynamic C **STDIO** window will indicate if writes and reads did not occur

  Connect an external SRAM as shown below before you run this sample program.



- `FLASHLED.C`—repeatedly flashes LED DS3 on the Prototyping Board on and off. LED DS3 is controlled by Parallel Port A bit 1 (PA1).
- `FLASHLED2.C`—repeatedly flashes LED DS3 on the Prototyping Board on and off. LED DS3 is controlled by Parallel Port A bit 1 (PA1).

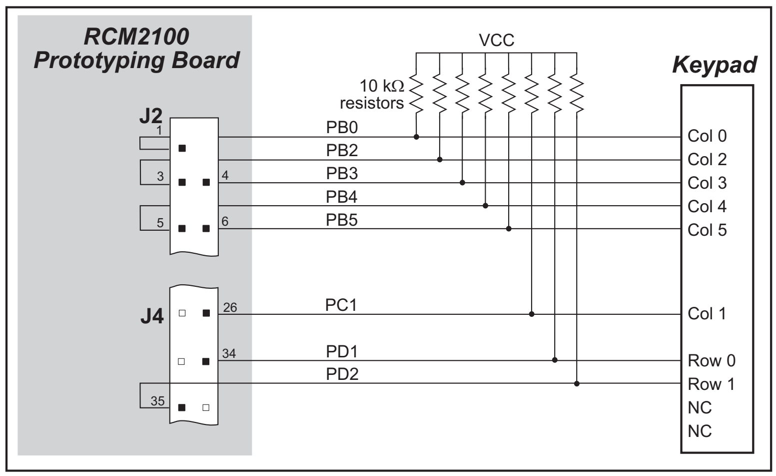
  This sample program also shows the use of the `runwatch()` function to allow Dynamic C to update watch expressions while running. The following steps explain how to do this.

  1. Add a watch expression for "k" in the **Inspect > Add Watch** dialog box.
  2. Click "Add" or "Add to top" so that it will be in the watch list permanently.
  3. Click **OK** to close the dialog box.
  4. Press **<Ctrl+U>** while the program is running. This will update the watch window.

- **`FLASHLEDS.C`**—demonstrates the use of coding with assembly instructions, cofunctions, and costatements to flash LEDs DS2 and DS3 on the Prototyping Board on and off. LEDs DS2 and DS3 are controlled by Parallel Port A bit 0 (PA0) and Parallel Port A bit 1 (PA1).Once you have compile this program and it is running, LEDs DS2 and DS3 will flash on/off at different rates.
- **`FLASHLEDS2.C`**—demonstrates the use of cofunctions and costatements to flash LEDs DS2 and DS3 on the Prototyping Board on and off. LEDs DS2 and DS3 are controlled by Parallel Port A bit 0 (PA0) and Parallel Port A bit 1 (PA1).Once you have compile this program and it is running, LEDs DS2 and DS3 will flash on/off at different rates.
- **`KEYLCD2.C`**—demonstrates a simple setup for a 2 × 6 keypad and a 2 × 20 LCD.

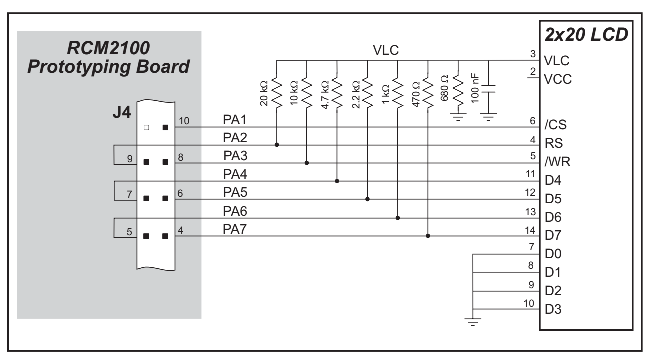
  Connect the keypad to Parallel Ports B, C, and D.

  PB0—Keypad Col 0
  PC1—Keypad Col 1
  PB2—Keypad Col 2
  PB3—Keypad Col 3
  PB4—Keypad Col 4
  PB5—Keypad Col 5
  PD1—Keypad Row 0
  PD2—Keypad Row 1



  Connect the LCD to Parallel Port A.

)PA0—backlight (if connected
  PA1—LCD /CS
  PA2—LCD RS (High = Control, Low = Data) / LCD Contrast 0
  PA3—LCD /WR/ LCD Contrast 1
  PA4—LCD D4 / LCD Contrast 2
  PA5—LCD D5 / LCD Contrast 3
  PA6—LCD D6 / LCD Contrast 4
  PA7—LCD D7 / LCD Contrast 5

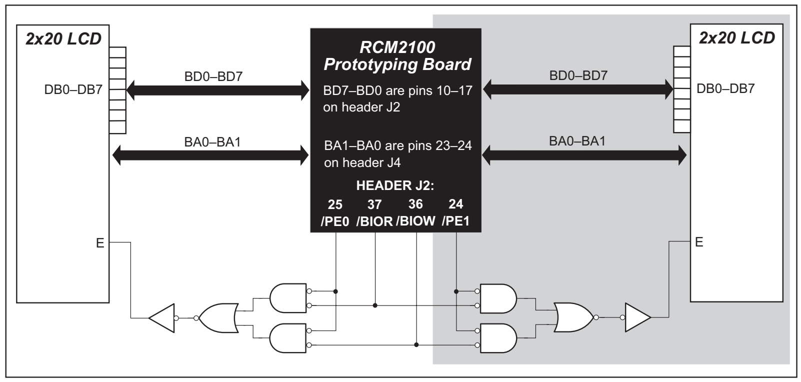


  Once the connections have been made and the sample program isrunning, the LCD will display two rows of 6 dots, each dot representing the corresponding key. When a key is pressed, the corresponding dot will become an asterisk.

- **`LCD_DEMO.C`**—demonstrates a simple setup for an LCD that uses the HD44780 controller or an equivalent.

  Connect the LCD to the RCM2100 address and data lines on the Prototyping Board.

  BD0—DB0
  BD1—DB1
  BD2—DB2
  BD3—DB3
  BD4—DB4
  BD5—DB5
  BD6—DB6
  BD7—DB7



  BA0—RS (Register Select: 0 = command, 1 = data)
  BA1—R/W (0=write, 1=read)
  *—E (normally low: latches on high-to-low transition)

- **`SWTEST.C`**—demonstrates the use of pushbutton switches S2 and S3 to toggle LEDs DS2 and DS3 on the Prototyping Board on and off.

  Parallel Port A bit 0 = LED DS2
  Parallel Port A bit 1 = LED DS3

  Parallel Port B bit 2 = switch S2
  Parallel Port B bit 3 = switch S3

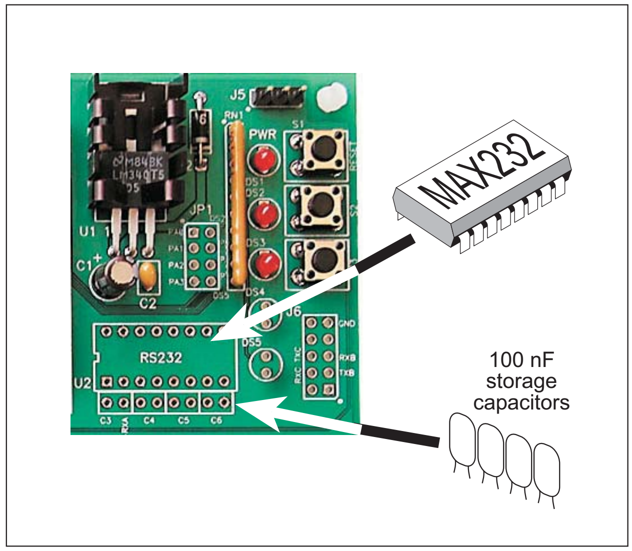
- **`TOGGLELED.C`**—demonstrates the use of costatements to detect switch presses using the press-and-release method of debouncing. As soon as the sample program starts running, LED DS3 on the Prototyping Board (which is controlled by PA1) starts flashing once per second. Press switch S2 on the Prototyping Board (which is connected to PB2) to toggle LED DS2 on the Prototyping Board (which is controlled by PA0). The pushbutton switch is debounced by the software.

3.1.2 Serial Communication

The following sample programs can be found in the **SAMPLES\RCM2100** folder.

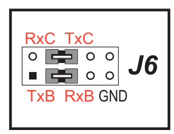
Two sample programs, **CORE_FLOWCONTROL.C** and **CORE_PARITY.C**, are available to illustrate RS-232 communication. To run these sample programs, you will have to add an RS-232 transceiver such as the MAX232 at location U2 and four 100 nF capacitors at C3–C6 on the Prototyping Board. Also install the 2 × 5 IDC header included with the Prototyping Board accessory parts at J6 to interface the RS-232 signals.

The diagram shows the connections.



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

  To set up the Prototyping Board, you will need to tie PC4 and PC5 (TxB and RxB) together at header J4, and you will also tie PC2 and PC3 (TxC and RxC) together using the jumpers supplied in the Development Kit as shown in the diagram.

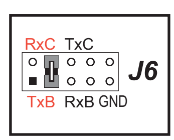
  

  A repeating triangular pattern should print out in the **STDIO** window.
  The program will periodically switch flow control on or off to demonstrate the effect of no flow control.

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

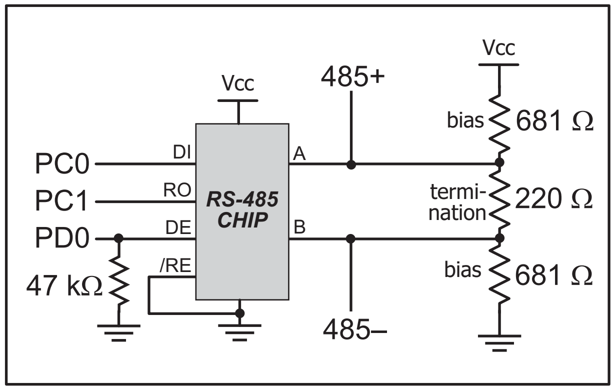
- **CORE_PARITY.C**—This program demonstrates the use of parity modes by repeatedly sending byte values 0–127 from Serial Port B to Serial Port C. The program will switch between generating parity or not on Serial Port B. Serial Port C will always be checking parity, so parity errors should occur during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

  To set up the Prototyping Board, you will need to tie PC4 and PC3 (TxB and RxC) together at header J4 using the jumpers supplied in the Development Kit as shown in the diagram.

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

Two sample programs, **`MASTER2.C`** and **`SLAVE2.C`**, are available to illustrate RS-485 master/slave communication. To run these sample programs, you will need a second Rabbit-based system with RS-485, and you will also have to add an RS-485 transceiver such as the SP483E and bias resistors to the Prototyping Board.



The diagram shows the connections. You will have to connect PC0 and PC1 (Serial Port D) on the Prototyping Board to the RS-485 transceiver, and you will connect PD0 to the RS-485 transceiver to enable or disable the RS-485 transmitter.

The RS-485 connections between the slave and master devices are as follows.

- RS485+ to RS485+
- RS485– to RS485–
- GND to GND

- **`MASTER2.C`**—This program demonstrates a simple RS-485 transmission of lower case letters to a slave RCM2100. The slave will send back converted upper case letters back to the master RCM2100 and display them in the **STDIO** window. Use **`SLAVE2.C`** to program the slave RCM2100.
- **`SLAVE2.C`**—This program demonstrates a simple RS-485 transmission of lower case letters to a master RCM2100. The slave will send back converted upper case letters back to the master RCM2100 and display them in the **STDIO** window. Use **`MASTER2.C`** to program the master RCM2100.

3.1.3 Other Sample Programs

Section 6.7 covers how to run the TCP/IP sample programs, which are then described in detail.

3.1.4 Sample Program Descriptions

3.1.4.1 FLASHLED.C

This program is about as simple as a Dynamic C application can get—the equivalent of the traditional “Hello, world!” program found in most basic programming tutorials. If you are familiar with ANSI C, you should have no trouble reading through the source code and understanding it.

The only new element in this sample application should be Dynamic C’s handling of the Rabbit microprocessor’s parallel ports. The program:

4. Initializes the pins of Port A as outputs.
5. Sets all of the pins of Port A high, turning off the attached LEDs.
6. Starts an endless loop with a `for(;;)` expression, and within that loop:
   - Writes a bit to turn bit 1 off, lighting LED DS3;
   - Waits through a delay loop;
   - Writes a bit to turn bit 1 on, turning off the LED;
   - Waits through a second delay loop;

   These steps repeat as long as the program is allowed to run.

You can change the flash rate of the LED by adjusting the loop values in the two `for` expressions. The first loop controls the LED’s “off” time; the second loop controls its “on” time.

> **NOTE:** Since the variable `j` is defined as type `int`, the range for `j` must be between 0 and 32767. To permit larger values and thus longer delays, change the declaration of `j` to `unsigned int` or `long`.

More Information

See the section on primitive data types, and the entries for the library functions `WrPortI( )` and `BitWrPortI( )` in the ***Dynamic C User’s Manual***.