

Welcome to E-XFL.COM

Understanding [Embedded - Microcontroller, Microprocessor, FPGA Modules](#)

Embedded - Microcontroller, Microprocessor, and FPGA Modules are fundamental components in modern electronic systems, offering a wide range of functionalities and capabilities. Microcontrollers are compact integrated circuits designed to execute specific control tasks within an embedded system. They typically include a processor, memory, and input/output peripherals on a single chip. Microprocessors, on the other hand, are more powerful processing units used in complex computing tasks, often requiring external memory and peripherals. FPGAs (Field Programmable Gate Arrays) are highly flexible devices that can be configured by the user to perform specific logic functions, making them invaluable in applications requiring customization and adaptability.

Applications of [Embedded - Microcontroller,](#)

Details

Product Status	Obsolete
Module/Board Type	MPU Core
Core Processor	Rabbit 2000
Co-Processor	-
Speed	22.1MHz
Flash Size	256KB
RAM Size	128KB
Connector Type	2 IDC Headers 2x13
Size / Dimension	1.6" x 2.3" (41mm x 58mm)
Operating Temperature	-40°C ~ 70°C
Purchase URL	https://www.e-xfl.com/product-detail/digi-international/101-0454

RabbitCore RCM2200 User's Manual

Part Number 019-0097 • 090417-G • Printed in U.S.A.

©2001–2009 Digi International Inc. • All rights reserved.

No part of the contents of this manual may be reproduced or transmitted in any form or by any means without the express written permission of Digi International.

Permission is granted to make one or more copies as long as the copyright page contained therein is included. These copies of the manuals may not be let or sold for any reason without the express written permission of Digi International.

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

Trademarks

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

Rabbit 2000 and RabbitCore are trademarks of Digi International Inc.

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

Digi International Inc.

www.rabbit.com

1.5 How to Use This Manual

This user's manual is intended to give users detailed information on the RCM2200 module. 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 RCM2200 module to effective use.

1.5.1 Additional Product Information

In addition to the product-specific information contained in the *RabbitCore RCM2200 User's Manual* (this manual), several higher level reference manuals are provided in HTML and PDF form on the accompanying CD-ROM. Advanced users will find these references valuable in developing systems based on the RCM2200 modules:

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

1.5.2 Online Documentation

The online documentation is installed along with Dynamic C, and an icon for the documentation menu is placed on the workstation's desktop. Double-click this icon to reach the menu. If the icon is missing, use your browser to find and load **default.htm** in the **docs** folder, found in the Dynamic C installation folder.

The latest versions of all documents are always available for free, unregistered download from our Web sites as well.

Although you can install a single module into either the **MASTER** or the **SLAVE** position on the Prototyping Board, all the Prototyping Board features (switches, LEDs, serial port drivers, etc.) are connected to the **MASTER** position. We recommend you install the module in the **MASTER** position.

NOTE: It is important that you line up the pins on headers J4 and J5 of the RCM2200 exactly with the corresponding pins of header sockets J1 and J2 on the Prototyping Board. The header pins may become bent or damaged if the pin alignment is offset, and the module will not work. Permanent electrical damage to the module may also result if a misaligned module is powered up.

Press the module's pins firmly into the Prototyping Board header sockets.

2.1.2 Connect Programming Cable

The programming cable connects the RCM2200 module to the PC running Dynamic C to download programs and to monitor the RCM2200 for debugging.

Connect the 10-pin connector of the programming cable labeled **PROG** to header J1 on the RCM2200 module as shown in Figure 2. Be sure to orient the marked (usually red) edge of the cable towards pin 1 of the connector. (Do not use the **DIAG** connector, which is used for a normal serial connection.)

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.

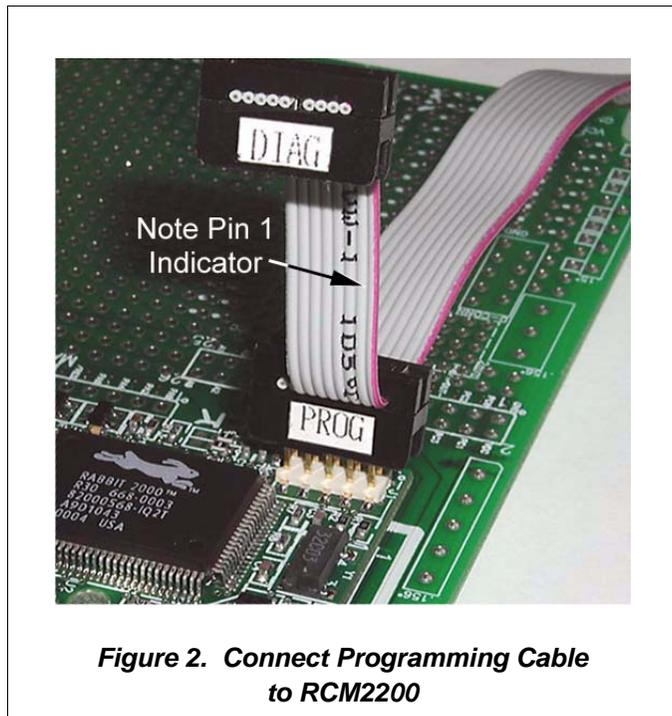


Figure 2. Connect Programming Cable to RCM2200

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. 20-151-0178) with the programming cable supplied with the RCM2200 Development Kit. Note that not all RS-232/USB converters work with Dynamic C.

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 RabbitCore module from the board.

2.1.4 Alternate Power Supply Connections

Development kits sold outside North America before 2009 included a header connector that could be connected to 3-pin header J5 on the Prototyping Board. 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 8 V–24 V DC at 8 W.

2.2 Run a Sample Program

Once the RCM2200 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. Dynamic C uses the serial port specified during installation.

If you are using a USB port to connect your computer to the RCM2200 module, choose **Options > Project Options** and select “Use USB to Serial Converter” under the **Communications** tab, then click **OK**.

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.

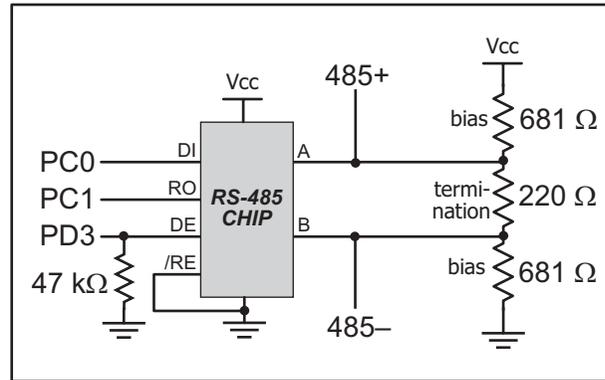
2.2.1 Troubleshooting

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

- Check that the RCM2200 is powered correctly — the red power LED on the Prototyping Board should be lit when the RCM2200 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 RCM2200 with the marked (colored) edge of the programming cable towards pin 1 of the programming header.
- Ensure that the RCM2200 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.

Two sample programs, **MASTER.C** and **SLAVE.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 PD3 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
- **MASTER.C**—This program demonstrates a simple RS-485 transmission of lower case letters to a slave RCM2200. The slave will send back converted upper case letters back to the master RCM2200 and display them in the **STDIO** window. Use **SLAVE.C** to program the slave RCM2200—reset the slave before you run **MASTER.C** on the master.
- **SLAVE.C**—This program demonstrates a simple RS-485 transmission of lower case letters to a master RCM2200. The slave will send back converted upper case letters back to the master RCM2200 and display them in the **STDIO** window. Compile and run this program on the slave before you use **MASTER.C** to program the master.

3.1.3 Other Sample Programs

Section 6.2 covers how to run the TCP/IP sample programs, which are then described in detail.

The RCM2200 has 26 parallel I/O lines grouped in five 8-bit ports available on headers J4 and J5. The 16 bidirectional I/O lines are located on pins PA0–PA7, PD3–PD5, and PE0–PE1, PE4, PE5, and PE7. The pinouts for headers J4 and J5 are shown in Figure 5.

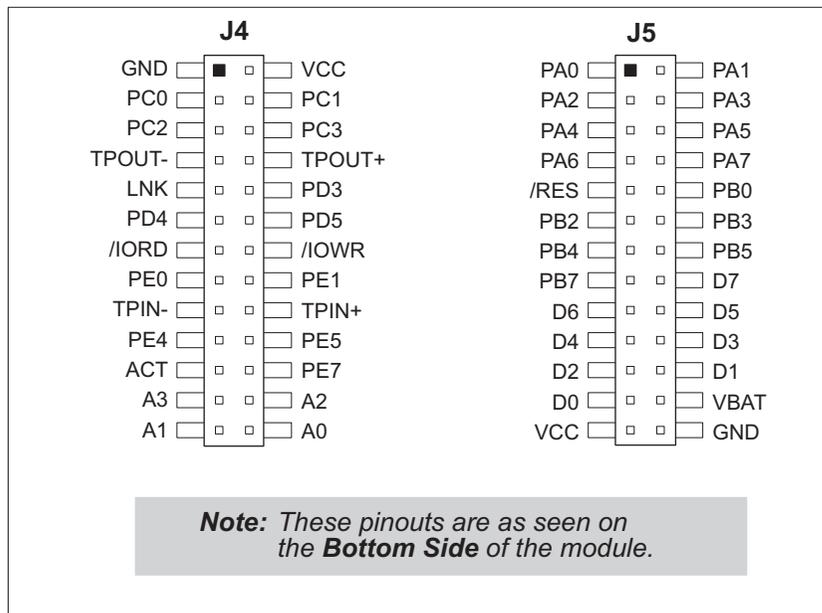


Figure 5. RCM2200 I/O Pinouts

4.1.1 Dedicated Inputs

PB0 is a general CMOS input when the Rabbit 2000 is either not using Serial Port B or is using Serial Port B in an asynchronous mode. Four other general CMOS input-only pins are located on PB2–PB5. These pins can also be used for the slave port. PB2 and PB3 are slave write and slave read strobes, while PB4 and PB5 serve as slave address lines SA0 and SA1, and are used to access the slave registers. PC1 and PC3 are general CMOS inputs only. These pins can instead be selectively enabled to serve as the serial data inputs for Serial Ports D and C.

4.1.2 Dedicated Outputs

One of the general CMOS output-only pins is located on PB7. PB7 can also be used with the slave port as the /SLAVEATTN output. This configuration signifies that the slave is requesting attention from the master. PC0 and PC2 are also output-only pins; PC0 and PC2 can instead serve as the serial data outputs for Serial Ports D and C.

4.1.3 Memory I/O Interface

Four of the Rabbit 2000 address lines (A0–A3) and all the data lines (D0–D7) are available. I/O write (/IOWR) and I/O read (/IORD) are also available for interfacing to external devices.

The ports on the Rabbit 2000 microprocessor used in the RCM2200 are configurable, and so the factory defaults can be reconfigured. Table 1 lists the Rabbit 2000 factory defaults and the alternate configurations.

Developing software with Dynamic C is simple. Users can write, compile, and test C and assembly code without leaving the Dynamic C development environment. Debugging occurs while the application runs on the target. Alternatively, users can compile a program to an image file for later loading. Dynamic C runs on PCs under Windows 95 or later. Programs can be downloaded at baud rates of up to 460,800 bps after the program compiles.

Dynamic C has a number of standard features:

- Full-feature source and/or assembly-level debugger, no in-circuit emulator required.
- Royalty-free TCP/IP stack with source code and most common protocols.
- Hundreds of functions in source-code libraries and sample programs:
 - ▶ Exceptionally fast support for floating-point arithmetic and transcendental functions.
 - ▶ RS-232 and RS-485 serial communication.
 - ▶ Analog and digital I/O drivers.
 - ▶ I²C, SPI, GPS, file system.
 - ▶ LCD display and keypad drivers.
- Powerful language extensions for cooperative or preemptive multitasking.
- Loader utility program to load binary images into Rabbit targets in the absence of Dynamic C.
- Provision for customers to create their own source code libraries and augment on-line help by creating “function description” block comments using a special format for library functions.
- Standard debugging features:
 - ▶ Breakpoints—Set breakpoints that can disable interrupts.
 - ▶ Single-stepping—Step into or over functions at a source or machine code level, μ C/OS-II aware.
 - ▶ Code disassembly—The disassembly window displays addresses, opcodes, mnemonics, and machine cycle times. Switch between debugging at machine-code level and source-code level by simply opening or closing the disassembly window.
 - ▶ Watch expressions—Watch expressions are compiled when defined, so complex expressions including function calls may be placed into watch expressions. Watch expressions can be updated with or without stopping program execution.
 - ▶ Register window—All processor registers and flags are displayed. The contents of general registers may be modified in the window by the user.
 - ▶ Stack window—shows the contents of the top of the stack.
 - ▶ Hex memory dump—displays the contents of memory at any address.
 - ▶ **STDIO** window—`printf` outputs to this window and keyboard input on the host PC can be detected for debugging purposes. `printf` output may also be sent to a serial port or file.

5.2 I/O

The RCM2200 was designed to interface with other systems, and so there are no drivers written specifically for the I/O. The general Dynamic C read and write functions allow you to customize the parallel I/O to meet your specific needs. For example, use

```
WrPortI (PEDDR, &PEDDRShadow, 0x00);
```

to set all the port E bits as inputs, or use

```
WrPortI (PEDDR, &PEDDRShadow, 0xFF);
```

to set all the Port E bits as outputs.

The sample programs in the Dynamic C `SAMPLES\RCM2200` directory provide further examples.

5.2.1 PCLK Output

The PCLK output is controlled by bits 7 and 6 of the Global Output Register (GOOCR) on the Rabbit 2000 microprocessor, and so can be enabled or disabled in software. Starting with Dynamic C v 7.02, the PCLK output is disabled by default at compile time to minimize radiated emissions; the PCLK output is enabled in earlier versions of Dynamic C.

Use the following code to set the PCLK output as needed.

PCLK output driven with peripheral clock:

```
WrPortI (GOOCR, &GOOCRShadow, (GOOCRShadow&~0xc0));
```

PCLK output driven with peripheral clock ÷ 2:

```
WrPortI (GOOCR, &GOOCRShadow, ((GOOCRShadow&~0xc0) | 0x40));
```

PCLK output off (low):

```
WrPortI (GOOCR, &GOOCRShadow, ((GOOCRShadow&~0xc0) | 0x80));
```

PCLK output on (high):

```
WrPortI (GOOCR, &GOOCRShadow, (GOOCRShadow | 0xc0));
```

5.2.2 External Interrupts

The Rabbit 2000 microprocessor has four external interrupt inputs on Parallel Port E, which is accessed through pins PE0, PE1, PE4, and PE5 on header J4. These pins may be used either as I/O ports or as external interrupt inputs.

Earlier versions of the Rabbit 2000 microprocessor labeled *IQIT* or *IQ2T* would occasionally lose an interrupt request when one of the interrupt inputs was used as a pulse counter. See Technical Note 301, *Rabbit 2000 Microprocessor Interrupt Problem*, for further information on how to work around this problem if you purchased your RCM2200 before July, 2002, and the Rabbit 2000 microprocessor is labeled *IQIT* or *IQ2T*.

NOTE: Interrupts on RCM2000 series RabbitCore modules sold after July, 2002, work correctly and do not need this workaround.

6.7 How to Set IP Addresses in the Sample Programs

We have provided a number of sample programs demonstrating various uses of TCP/IP for networking embedded systems. These programs require that you connect your PC and the Coyote together on the same network. This network can be a local private network (preferred for initial experimentation and debugging), or a connection via the Internet.

With the introduction of Dynamic C 7.30 we have taken steps to make it easier to run many of our sample programs. You will see a **TCPCONFIG** macro. This macro tells Dynamic C to select your configuration from a list of default configurations. You will have three choices when you encounter a sample program with the **TCPCONFIG** macro.

1. You can replace the **TCPCONFIG** macro with individual **MY_IP_ADDRESS**, **MY_NETMASK**, **MY_GATEWAY**, and **MY_NAMESERVER** macros in each program.
2. You can leave **TCPCONFIG** at the usual default of 1, which will set the IP configurations to 10.10.6.100, the netmask to 255.255.255.0, and the nameserver and gateway to 10.10.6.1. If you would like to change the default values, for example, to use an IP address of 10.1.1.2 for the Coyote board, and 10.1.1.1 for your PC, you can edit the values in the section that directly follows the “General Configuration” comment in the **TCP_CONFIG.LIB** library. You will find this library in the **LIB\TCPIP** directory.
3. You can create a **CUSTOM_CONFIG.LIB** library and use a **TCPCONFIG** value greater than 100. Instructions for doing this are at the beginning of the **TCP_CONFIG.LIB** library in the **LIB\TCPIP** directory.

There are some other “standard” configurations for **TCPCONFIG** that let you select different features such as DHCP. Their values are documented at the top of the **TCP_CONFIG.LIB** library in the **LIB\TCPIP** directory. More information is available in the *Dynamic C TCP/IP User's Manual*.

IP Addresses Before Dynamic C 7.30

Most of the sample programs use macros to define the IP address assigned to the board and the IP address of the gateway, if there is a gateway. Instead of the **TCPCONFIG** macro, you will see a **MY_IP_ADDRESS** macro and other macros.

```
#define MY_IP_ADDRESS "10.10.6.170"  
#define MY_NETMASK "255.255.255.0"  
#define MY_GATEWAY "10.10.6.1"  
#define MY_NAMESERVER "10.10.6.1"
```

In order to do a direct connection, the following IP addresses can be used for the Coyote:

```
#define MY_IP_ADDRESS "10.1.1.2"  
#define MY_NETMASK "255.255.255.0"  
// #define MY_GATEWAY "10.10.6.1"  
// #define MY_NAMESERVER "10.10.6.1"
```

In this case, the gateway and nameserver are not used, and are commented out. The IP address of the board is defined to be 10.1.1.2. The IP address of your PC can be defined as 10.1.1.1.

6.8 How to Set Up Your Computer for Direct Connect

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.

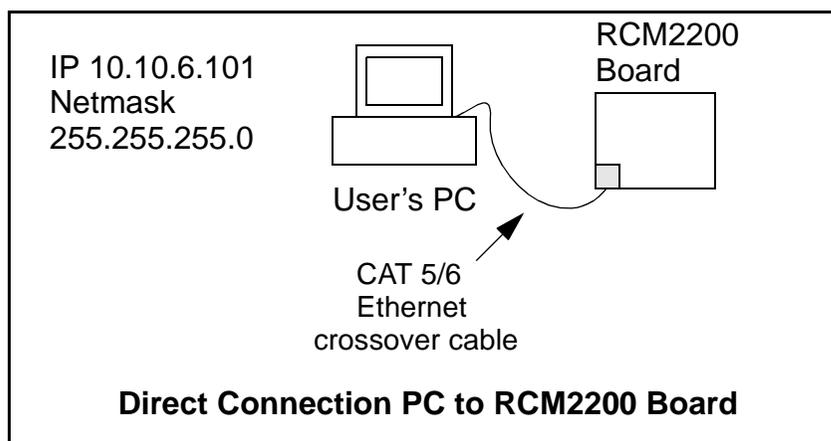
3. Now select the **IP Address** tab, and check **Specify an IP Address**, or select TCP/IP and click on “Properties” to assign an IP address to your computer (this will disable “obtain an IP address automatically”):

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.



A.1.1 Headers

The RCM2200 uses headers at J4 and J5 for physical connection to other boards. J4 and J5 are 2×13 SMT headers with a 2 mm pin spacing. J1, the programming port, is a 2×5 header with a 2 mm pin spacing.

Figure A-3 shows the layout of another board for the RCM2200 to be plugged into. These values are relative to the header connectors.

A.1.2 Physical Mounting

A $9/32''$ (7 mm) standoff with a 4-40 screw is recommended to attach the RCM2200 to a user board at the hole position shown in Figure A-3. Either use plastic hardware, or use insulating washers to keep any metal hardware from shorting out signals on the RCM2200.

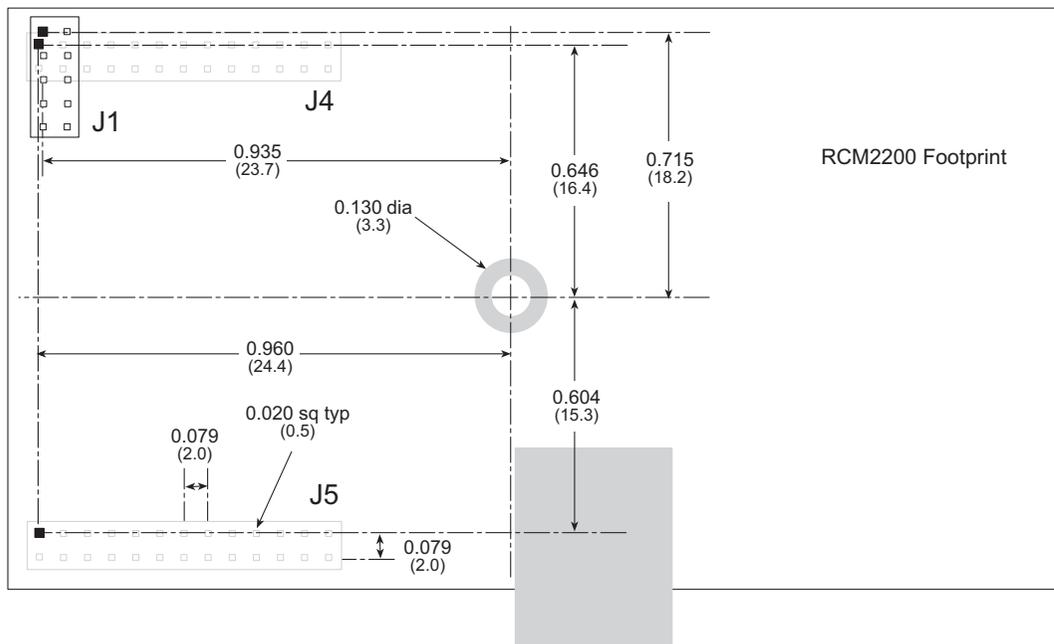


Figure A-3. User Board Footprint for RCM2200

A.4 I/O Buffer Sourcing and Sinking Limit

Unless otherwise specified, the Rabbit I/O buffers are capable of sourcing and sinking 8 mA of current per pin at full AC switching speed. Full AC switching assumes a 25.8 MHz CPU clock and capacitive loading on address and data lines of less than 100 pF per pin. Address pin A0 and data pin D0 are rated at 16 mA each. Pins A1–A12 and D1–D7 are each rated at 8 mA. The absolute maximum operating voltage on all I/O is $V_{DD} + 0.5$ V or 5.5 V.

Table A-5 shows the AC and DC output drive limits of the parallel I/O buffers when the Rabbit 2000 is used in the RCM2200.

Table A-5. I/O Buffer Sourcing and Sinking Capability

Pin Name	Output Drive Sourcing [*] /Sinking [†] Limits (mA)	
	Full AC Switching SRC/SNK	Maximum [‡] DC Output Drive SRC/SNK
PA [7:0]	8/8	12/12
PB [7:6]	8/8	12/12
PC [6, 2, 0]	8/8	12/12
PD [5:4]	8/8	12/12
PD [3:0]**	16/16	25/25
PE [7, 5, 4, 1, 0]	8/8	12/12

* The maximum DC sourcing current for I/O buffers between V_{DD} pins is 112 mA.

† The maximum DC sinking current for I/O buffers between V_{SS} pins is 150 mA.

‡ The maximum DC output drive on I/O buffers must be adjusted to take into consideration the current demands made by AC switching outputs, capacitive loading on switching outputs, and switching voltage.

The current drawn by all switching and nonswitching I/O must not exceed the limits specified in the first two footnotes.

** The combined sourcing from Port D [7:0] may need to be adjusted so as not to exceed the 112 mA sourcing limit requirement specified in Note 1.



APPENDIX B. PROTOTYPING BOARD

Appendix B describes the features and accessories of the Prototyping Board, and explains the use of the Prototyping Board to demonstrate the RCM2200 and to build prototypes of your own circuits.

circuits can be prototyped using point-to-point wiring with 20 to 30 AWG wire between the prototyping area and the holes at locations J7 and J8. The holes are spaced at 0.1" (2.5 mm), and 40-pin headers or sockets may be installed at J7 and J8. The pinouts for locations J7 and J8, which correspond to headers J1 and J2, are shown in Figure B-5.

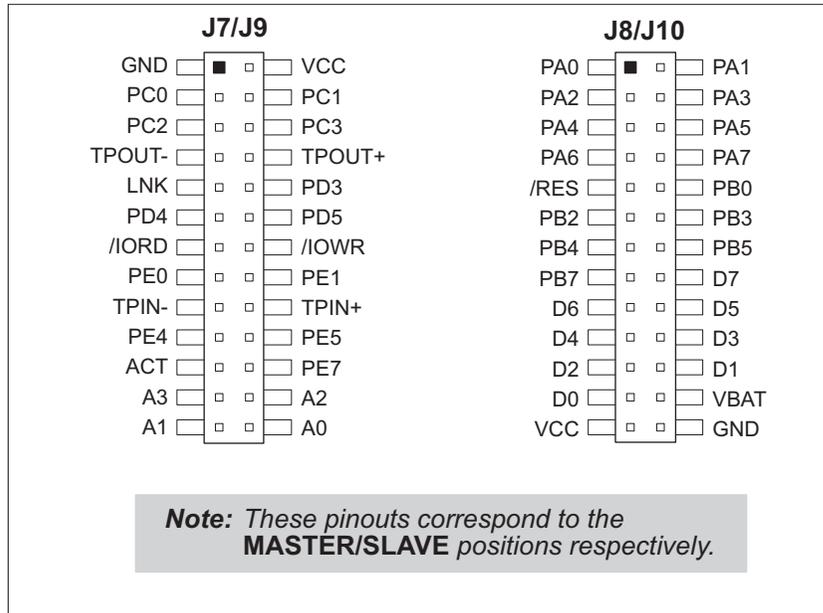


Figure B-5. RCM2200 Prototyping Board Pinout (Top View)

The small holes are also provided for surface-mounted components that may be installed to the right of the prototyping area.

There is a 2.4" × 4" through-hole prototyping space available on the Prototyping Board. VCC and GND traces run along the edge of the Prototyping Board for easy access. A GND pad is also provided at the lower right for alligator clips or probes.

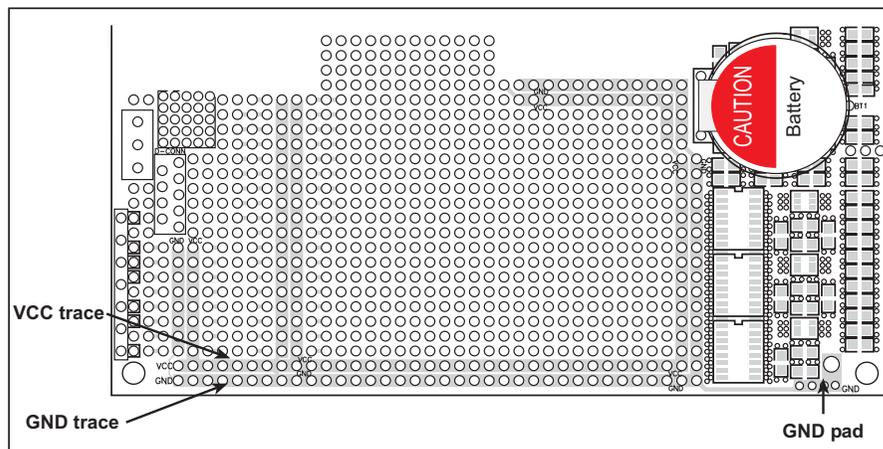


Figure B-6. VCC and GND Traces Along Edge of Prototyping Board

B.4.1 Adding Other Components

There is room on the Prototyping Board for a user-supplied RS-232 transceiver chip at location U2 and a 10-pin header for serial interfacing to external devices at location J6. A Maxim MAX232 transceiver is recommended. When adding the MAX232 transceiver at position U2, you must also add 100 nF charge storage capacitors at positions C3–C7 as shown in Figure B-7.

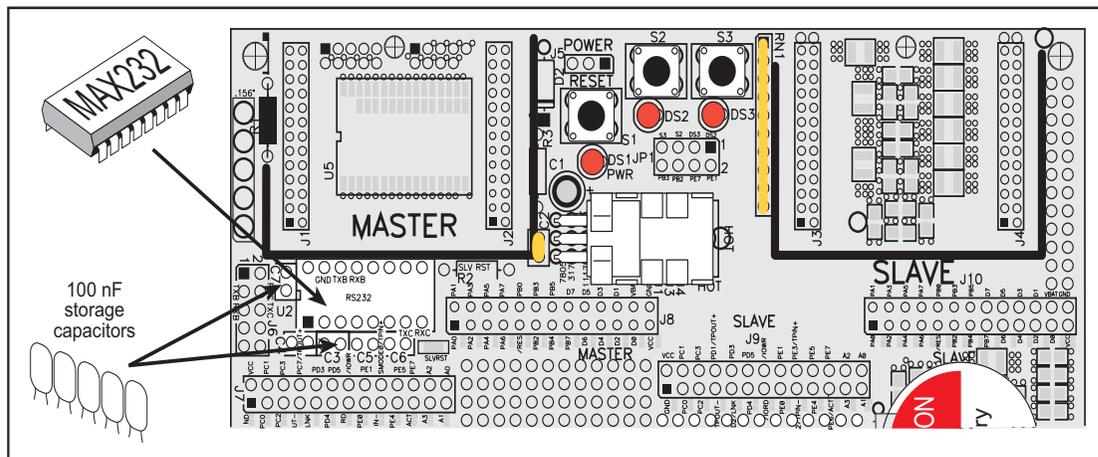


Figure B-7. Location for User-Supplied RS-232 Transceiver and Charge Storage Capacitors on Back Side of Prototyping Board

NOTE: The board that is supplied with the DeviceMate Development Kit already has the RS-232 chip and the storage capacitors installed, and is called the DeviceMate Demonstration Board.

There are two sets of pads that can be used for surface mount prototyping SOIC devices. The silk screen layout separates the rows into six 16-pin devices (three on each side). However, there are pads between the silk screen layouts giving the user two 52-pin (2×26) SOIC layouts with 50 mil pin spacing. There are six sets of pads that can be used for 3- to 6-pin SOT23 packages. There are also 60 sets of pads that can be used for SMT resistors and capacitors in an 0805 SMT package. Each component has every one of its pin pads connected to a hole in which a 30 AWG wire can be soldered (standard wire wrap wire can be soldered in for point-to-point wiring on the Prototyping Board). Because the traces are very thin, carefully determine which set of holes is connected to which surface-mount pad.

There is also a space above the space for the RS-232 transceiver that can accommodate a large surface-mounted SOIC component.

APPENDIX C. POWER SUPPLY

Appendix C provides information on the current requirements of the RCM2200, and includes some background on the chip select circuit used in power management.

C.1 Power Supplies

The RCM2200 requires a regulated $5\text{ V} \pm 0.25\text{ V}$ DC power source. The RabbitCore design presumes that the voltage regulator is on the user board, and that the power is made available to the RabbitCore board through headers J4 and J5.

An RCM2200 with no loading at the outputs operating at 22.1 MHz typically draws 134 mA. The RCM2200 will consume an additional 10 mA when the programming cable is used to connect the programming header, J1, to a PC.

C.1.1 Battery-Backup Circuits

The RCM2200 does not have a battery, but there is provision for a customer-supplied battery to back up SRAM and keep the internal Rabbit 2000 real-time clock running.

Header J5, shown in Figure C-1, allows access to the external battery. This header makes it possible to connect an external 3 V power supply. This allows the SRAM and the internal Rabbit 2000 real-time clock to retain data with the RCM2200 powered down.

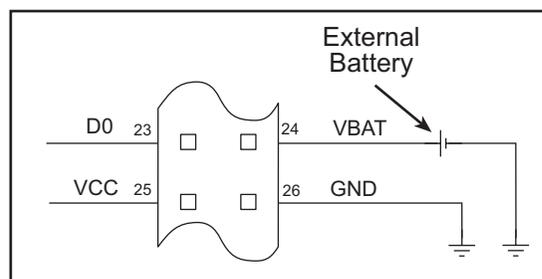


Figure C-1. External Battery Connections at Header J5

A lithium battery with a nominal voltage of 3 V and a minimum capacity of 165 mA-h is recommended. A lithium battery is strongly recommended because of its nearly constant nominal voltage over most of its life.

INDEX

A

additional information
 online documentation 4

B

backup-battery circuit
 external battery connections .
 71
battery life 72
battery-backup circuit
 reset generator 72
bus loading 54

C

clock doubler 28
conformal coating 59

D

Development Kit 3
 DeviceMate 69
 RCM2200 3
digital I/O 19
 I/O buffer sourcing and sink-
 ing limits 57
 memory interface 20
 SMODE0 24
 SMODE1 24
digital inputs 20
digital outputs 20
dimensions
 Prototyping Board 65
 RCM2200 50
Dynamic C 3, 31
 add-on modules 35
 Rabbit Embedded Security
 Pack 35
 sample programs 11
 standard features 32
 debugging 32
 telephone-based technical
 support 35
 upgrades and patches 35
 USB port settings 9

E

EMI
 spectrum spreader feature . 29
Ethernet cables 37
Ethernet connections 37, 40
 10Base-T 40
 10Base-T Ethernet card 37
 additional resources 48
 Ethernet cables 40
 Ethernet hub 37
 IP addresses 39, 41
 steps 37, 38
Ethernet port 23
 pinout 23
exclusion zone 51
external interrupts 33

F

features 1, 2
 Prototyping Board 62, 63
flash memory
 using second 256K flash
 memory 31
flash memory addresses
 user blocks 27

H

hardware connections
 install RCM2200 on Prototyp-
 ing Board 6
 power supply 8
 programming cable 7
hardware reset 8

I

I/O buffer sourcing and sinking
 limits 57
IP addresses 41
 how to set 45
 how to set PC IP address ... 46

J

JP1
 Prototyping Board 67
jumper configurations 58
 JP1 (U8 flash memory size)
 58
 JP2 (U8 flash memory bank
 select) 27, 58
 JP3 (U3 flash memory bank
 select) 27, 58
 JP4 (U3 flash memory size)
 58
 JP5 (U7 flash memory bank
 select) 27, 58
 JP6 (U7 flash memory size)
 58
 JP7 (SRAM size) 58
 jumper locations 58

M

manuals 4

P

PCLK output 33
physical mounting 53
pinout
 Ethernet port 23
 Prototyping Board 68
 RCM2200 20
 alternate configurations
 21, 22
power supplies 71
 chip select circuit 71
power supply
 connections 8
Program Mode 25
 switching modes 25
programming cable
 PROG connector 25
 RCM2200 connections 7
programming port 24