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	512KB
RAM Size	512KB
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/101-0494

TABLE OF CONTENTS

2. GETTING STARTED

This chapter describes the RCM2200 hardware in more detail, and explains how to set up and use the accompanying Prototyping Board.

NOTE: This chapter (and this manual) assume that you have the RCM2200 Development Kit. If you purchased an RCM2200 module by itself, you will have to adapt the information in this chapter and elsewhere to your test and development setup.

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.

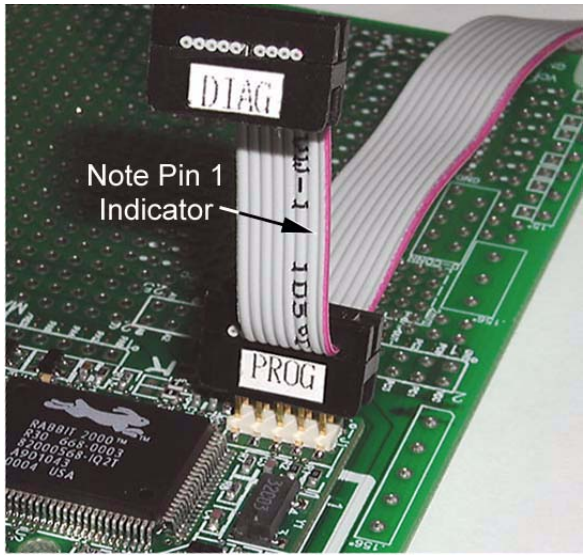

Figure 2. Connect Programming Cable to RCM2200

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.

2.1.3 Connect Power

When all other connections have been made, you can connect power to the RCM2200 Prototyping Board.

First, prepare the AC adapter for the country where it will be used by selecting the plug. The RCM2200 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 J5 on the Prototyping Board as shown in Figure 3 below. The connector may be attached either way as long as it is not offset to one side.

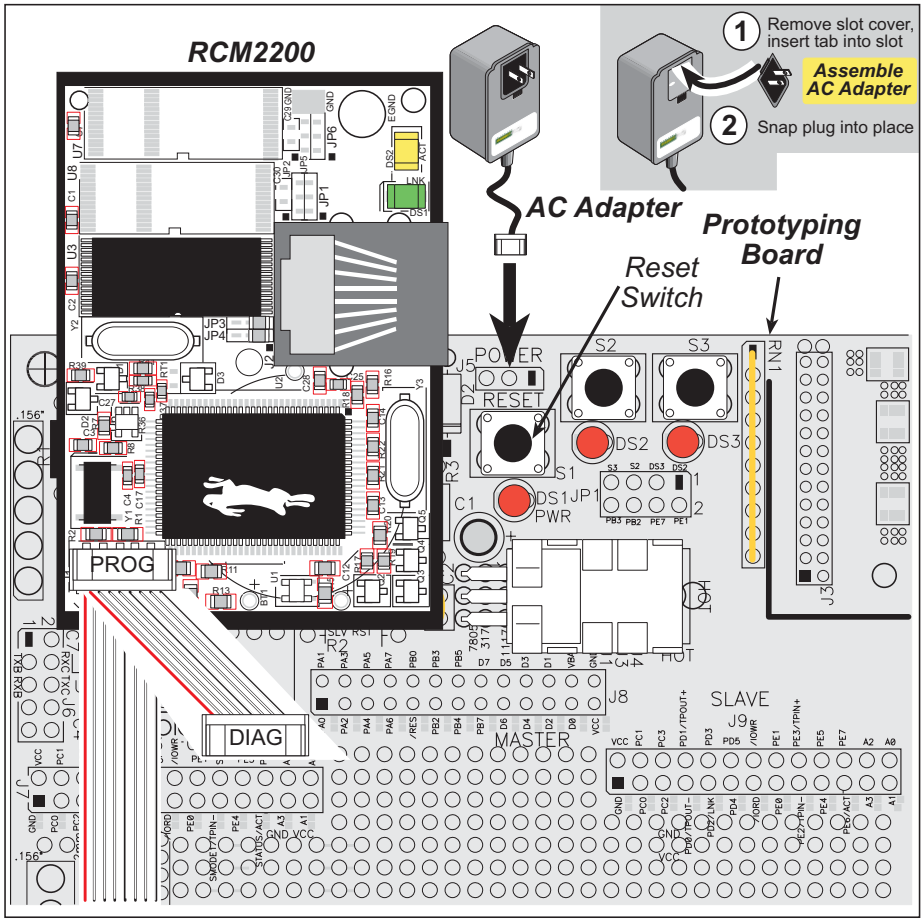

Figure 3. Power Supply Connections

Plug in the AC adapter. The power LED on the Prototyping Board should light up. The RCM2200 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 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.

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***.

4.2 Serial Communication

The RCM2200 board does not have an RS-232 or an RS-485 transceiver directly on the board. However, an RS-232 or RS-485 interface may be incorporated on the board the RCM2200 is mounted on. For example, the Prototyping Board supports a standard RS-232 transceiver chip.

4.2.1 Serial Ports

There are four serial ports designated as Serial Ports A, B, C, and D. All four serial ports can operate in an asynchronous mode up to the baud rate of the system clock divided by 64. An asynchronous port can handle 7 or 8 data bits. A 9th bit address scheme, where an additional bit is sent to mark the first byte of a message, is also supported. Serial Ports A and B can also be operated in the clocked serial mode. In this mode, a clock line synchronously clocks the data in or out. Either of the two communicating devices can supply the clock. When the Rabbit 2000 provides the clock, the baud rate can be up to 80% of the system clock frequency divided by 128, or 138,240 bps for a 22.1 MHz clock speed.

Serial Port A is available only on the programming port, and so is likely to be inconvenient to interface with.

4.2.2 Ethernet Port

Figure 6 shows the pinout for the RJ-45 Ethernet port (J2). Note that some Ethernet connectors are numbered in reverse to the order used here.

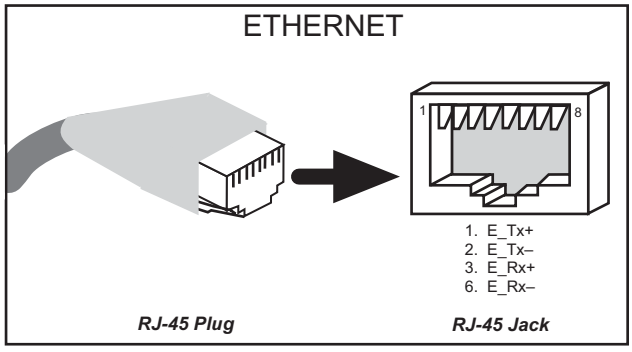

Figure 6. RJ-45 Ethernet Port Pinout

Two LEDs are placed next to the RJ-45 Ethernet jack, one to indicate an Ethernet link (**LNK**) and one to indicate Ethernet activity (**ACT**).

The Ethernet signals are also available on header J4. The **ACK** and **LNK** signals can be used to drive LEDs on the user board the RCM2200 is connected to.

The transformer/connector assembly ground is connected to the RCM2200 printed circuit board digital ground via a 0 Ω resistor, R29, as shown in Figure 7.

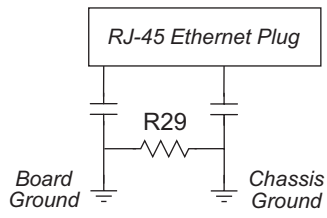

Figure 7. Isolation Resistor R29

5. SOFTWARE REFERENCE

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 single-board computers and other single-board computers based on the Rabbit microprocessor. Chapter 4 provides the libraries and function calls related to the RCM2200.

5.1 More About Dynamic C

Dynamic C has been in use worldwide since 1989. Dynamic C is specially designed for programming embedded systems, and features quick compile and interactive debugging. A complete reference to Dynamic C is contained in the ***Dynamic C User's Manual***.

You have a choice of doing your software development in the flash memory or in the static RAM included on the RCM2200. The flash memory and SRAM options are selected with the **Options > Project Options > Compiler** menu.

The advantage of working in RAM is to save wear on the flash memory, which is limited to about 100,000 write cycles. The disadvantage is that the code and data might not both fit in RAM.

NOTE: An application can be developed in RAM, but cannot run standalone from RAM after the programming cable is disconnected. All standalone applications can only run from flash memory.

NOTE: Do not depend on the flash memory sector size or type. Due to the volatility of the flash memory market, the RCM2200 and Dynamic C were designed to accommodate flash devices with various sector sizes.

RCM2250 and RCM2260 RabbitCore modules have two 256K flash memories. By default, Dynamic C will use only the first flash memory for program code in these RCM2250 and RCM2260 RabbitCore modules. Uncomment he BIOS **USE_2NDFLASH_CODE** macro to allow the second flash memory to hold any program code that is in excess of the available memory in the first flash.

6.4 How IP Addresses are Used

The actual hardware connection via an Ethernet uses Ethernet adapter addresses (also called MAC addresses.) These are 48-bit addresses and are unique for every Ethernet adapter manufactured. In order to send a packet to another computer, given the IP address of the other computer, it is first determined if the packet needs to be sent directly to the other computer or to the gateway. In either case, there is an IP address on the local network to which the packet must be sent. A table is maintained to allow the protocol driver to determine the MAC address corresponding to a particular IP address. If the table is empty, the MAC address is determined by sending an Ethernet broadcast packet to all devices on the local network asking the device with the desired IP address to answer with its MAC address. In this way, the table entry can be filled in. If no device answers, then the device is nonexistent or inoperative, and the packet cannot be sent.

IP addresses are arbitrary and can be allocated as desired provided that they don't conflict with other IP addresses. However, if they are to be used with the Internet, then they must be numbers that are assigned to your connection by proper authorities, generally by delegation via your service provider.

6.9 Run the PINGME.C Sample Programs

Connect the crossover cable from your computer's Ethernet port to the RCM2200 board's RJ-45 Ethernet connector. Open this sample program from the **SAMPLES\TCPIP\ICMP** folder, compile the program, and start it running under Dynamic C. When the program starts running, the green **LNK** light on the RCM2200 board should be on to indicate an Ethernet connection is made. (Note: If the **LNK** light does not light, you may not have a crossover cable, or if you are using a hub perhaps the power is off on the hub.)

The next step is to ping the board from your PC. This can be done by bringing up the MS-DOS window and running the pingme program:

```
ping 10.10.6.100
```

or by **Start > Run**

and typing the entry

```
ping 10.10.6.100
```

Notice that the red **ACT** light flashes on the RCM2200 board while the ping is taking place, and indicates the transfer of data. The ping routine will ping the board four times and write a summary message on the screen describing the operation.

6.10 Running More Sample Programs With Direct Connect

The sample programs discussed here are in the Dynamic C **SAMPLES\RCM2200** folder.

- **CONSOLE.C**—Demonstrates the features of **ZCONSOLE.LIB** command-oriented console library. This program is also run in conjunction with **SERDCLIENT.C** or **SPCLIENT.C**.
- **ETHCORE1.C**—Creates two "devices" (lights) and two "buttons" in the Web browser to toggle them. Users can change the status of the lights. If the RCM2200 is plugged into the **MASTER** slot on the Prototyping Board, the lights on the Prototyping Board will track the ones in the Web browser. As long as you have not modified the **TCPCONFIG 1** macro in the sample program, enter the following server address in your Web browser to bring up the Web page served by the sample program.

  http://10.10.6.100

  Otherwise use the TCP/IP settings you entered in the **TCP_CONFIG.LIB** library.
- **MYECHO.C**—Operates RCM2200 as a basic server. When a client connects, echoes back any data sent by the client.
- **SERDCLIENT.C**—Demonstrates the ability of a Rabbit-based target board to update files on the Web server of the RCM2200 board it is connected to via Serial Port D. This program is run in conjunction with **CONSOLE.C**.
- **SPCLIENT.C**—Demonstrates the ability of a Rabbit-based target board to update files on the Web server of the RCM2200 board it is connected to via the slave port. This program is run in conjunction with **CONSOLE.C**.

Table A-1 lists the electrical, mechanical, and environmental specifications for the RCM2200.

Table A-1. RabbitCore RCM2200 Specifications

Parameter	RCM2200	RCM2210	RCM2250	RCM2260
Microprocessor	Rabbit 2000® at 22.1 MHz			
Ethernet Port (10/100-compatible with 10Base-T interface)	RJ-45, 2 LEDs	Raw signals only	RJ-45, 2 LEDs	Raw signals only
Flash Memory	One 256K		Two 256K	Two 256K
SRAM	128K		512K	512K
Backup Battery	Connection for user-supplied backup battery (to support RTC and SRAM)			
General-Purpose I/O	26 parallel I/0 lines grouped in five 8-bit ports (shared with serial ports): • 16 configurable I/O • 7 fixed inputs • 3 fixed outputs			
Additional Inputs	2 startup mode, reset			
Additional Outputs	Status, reset			
Memory, I/O Interface	4 address lines, 8 data lines, I/O read/write			
Serial Ports	Four 5 V CMOS-compatible ports. Two ports are configurable as clocked ports, one is a dedicated RS-232 programming port.			
Serial Rate	Maximum burst rate = CLK/32 Maximum sustained rate = CLK/64			
Slave Interface	A slave port allows the RCM2200 to be used as an intelligent peripheral device slaved to a master processor, which may either be another Rabbit 2000 or any other type of processor			
Real-Time Clock	Yes			
Timers	Five 8-bit timers cascadable in pairs, one 10-bit timer with 2 match registers that each have an interrupt			
Watchdog/Supervisor	Yes			
Power	4.75 V to 5.25 V DC, 134 mA			
Operating Temperature	–40°C to +70°C			
Humidity	5% to 95%, noncondensing			
Connectors	Two IDC headers 2 × 13, 2 mm pitch			
Board Size	1.60" × 2.30" × 0.86" (41 mm × 59 mm × 22 mm)			

A.2 Bus Loading

You must pay careful attention to bus loading when designing an interface to the RCM2200. This section provides bus loading information for external devices.

Table A-2 lists the capacitance for the various RCM2200 I/O ports.

Table A-2. Capacitance of Rabbit 2000 I/O Ports

I/O Ports	Input Capacitance (pF)	Output Capacitance (pF)
Parallel Ports A to E	12	14
Data Lines BD0–BD7	10	12
Address Lines BA0–BA12	4	8

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)	
Output Port Name	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 my 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.

A.6 Conformal Coating

The areas around the 32 kHz real-time clock crystal oscillator has had the Dow Corning silicone-based 1-2620 conformal coating applied. The conformally coated area is shown in Figure A-6. The conformal coating protects these high-impedance circuits from the effects of moisture and contaminants over time.

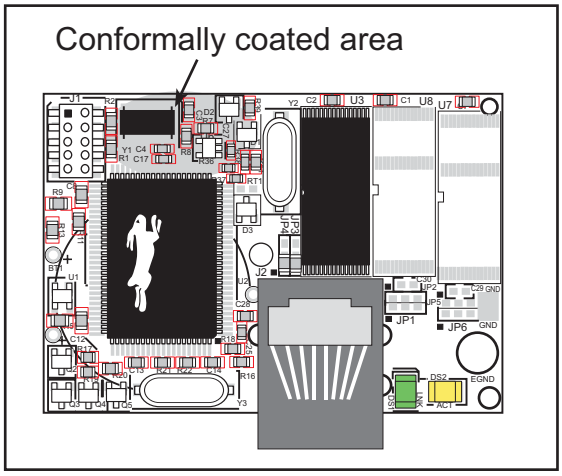

Figure A-6. RCM2200 Areas Receiving Conformal Coating

Any components in the conformally coated area may be replaced using standard soldering procedures for surface-mounted components. A new conformal coating should then be applied to offer continuing protection against the effects of moisture and contaminants.

NOTE: For more information on conformal coatings, refer to Technical Note TN303, ***Conformal Coatings***.

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.

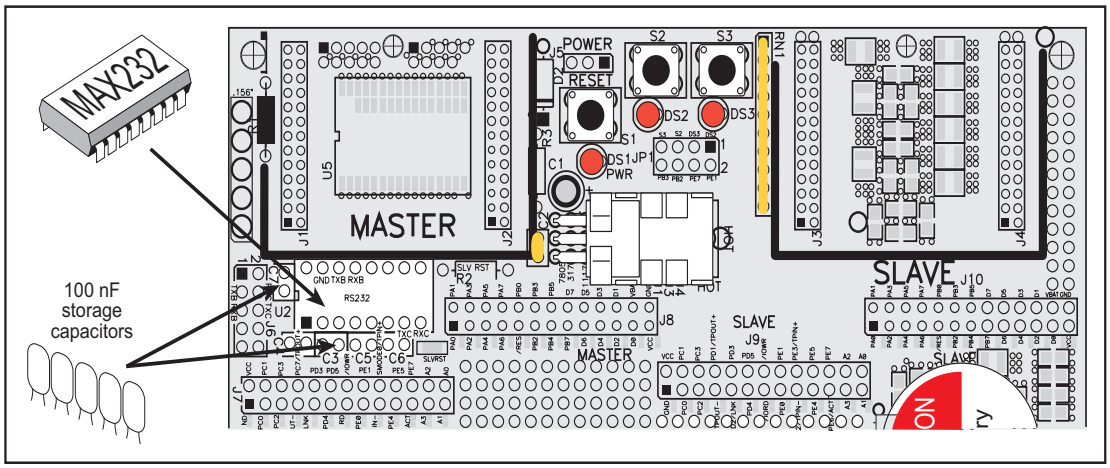

Figure B-7. Location for User-Supplied RS-232 Transceiver and Charge Storage Capacitors on Back Side of Prototyping Board

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 V ± 0.25 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.

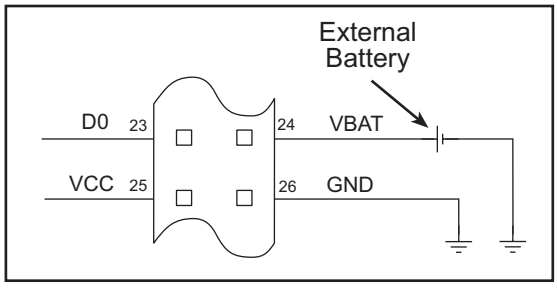

Figure C-1. External Battery Connections at Header J5

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.

D.2 Keypad and LCD Connections

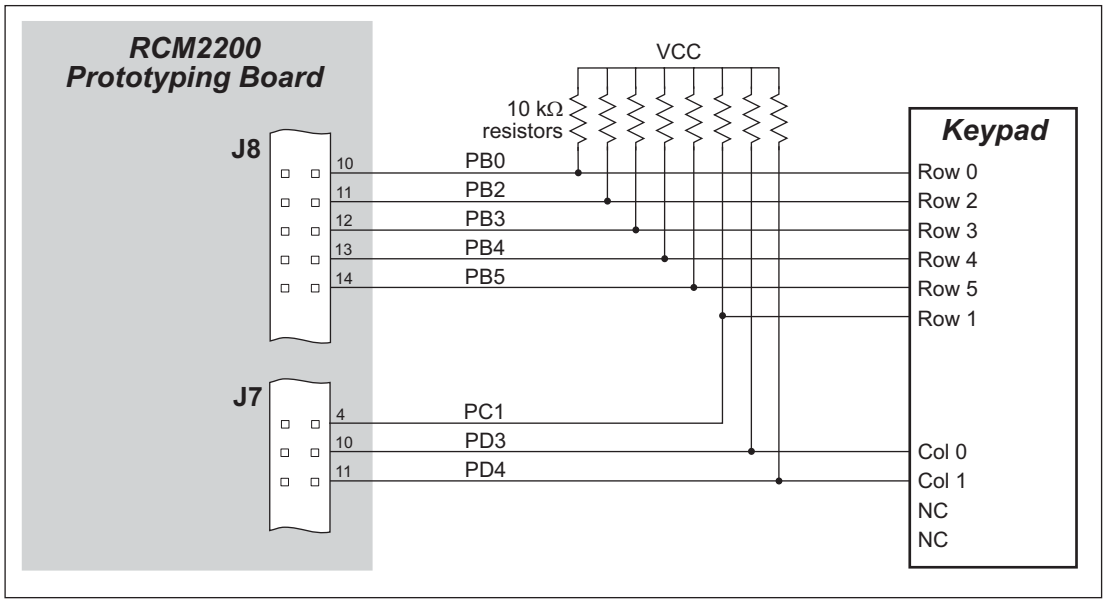

Figure D-2. Sample Keypad Connections

Sample Program: **KEYLCD.C** in **SAMPLES\RCM2200**.

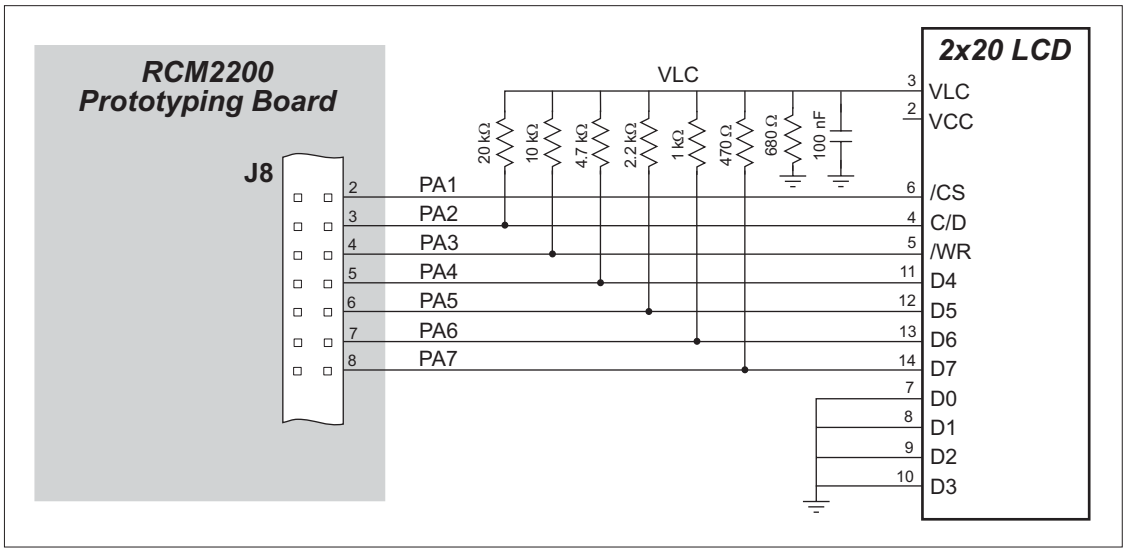

Figure D-3. Sample LCD Connections

Sample Program: **KEYLCD.C** in **SAMPLES\RCM2200**.

SCHEMATICS

090-0120 RCM2200 Schematic

www.rabbit.com/documentation/schemat/090-0120.pdf

090-0122 RCM2200 Prototyping Board Schematic

www.rabbit.com/documentation/schemat/090-0122.pdf

090-0128 Programming Cable Schematic

www.rabbit.com/documentation/schemat/090-0128.pdf

You may use the URL information provided above to access the latest schematics directly.