Digi - 20-101-0488 Datasheet

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	0°C ~ 70°C
Purchase URL	https://www.e-xfl.com/product-detail/digi-international/20-101-0488

1. INTRODUCTION

The RCM2200 RabbitCore module is designed to be the heart of embedded control systems. The RCM2200 features an integrated Ethernet port and provides for LAN and Internet-enabled systems to be built as easily as serial-communication systems.

Throughout this manual, the term RCM2200 refers to the complete series of RCM2200 RabbitCore modules unless other production models are referred to specifically.

The RCM2200 has a Rabbit 2000 microprocessor operating at 22.1 MHz, static RAM, flash memory, two clocks (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 26-pin headers bring out the Rabbit 2000 I/O bus lines, address lines, data lines, parallel ports, and serial ports.

The RCM2200 receives its +5 V power from the user board on which it is mounted. The RabbitCore RCM2200 can interface with all kinds of CMOS-compatible digital devices through the user board.

1.1 RCM2200 Features

- Small size: 1.60" × 2.30" × 0.86"
  (41 mm × 58 mm × 22 mm)
- Microprocessor: Rabbit 2000 running at 22.1 MHz
- 26 parallel I/O lines: 16 configurable for input or output, 7 fixed inputs, 3 fixed outputs
- 8 data lines (D0–D7)
- 4 address lines (A0–A3)
- Memory I/0 read, write
- External reset input
- Five 8-bit timers (cascadable in pairs) and two 10-bit timers
- 256K–512K flash memory, 128K–512K SRAM
- Real-time clock
- Watchdog supervisor

- Provision for customer-supplied backup battery via connections on header J5
- 10/100-compatible RJ-45 Ethernet port with 10Base-T interface (Ethernet jack not installed on all models)
- Raw Ethernet and two associated LED control signals available on 26-pin header
- Three CMOS-compatible serial ports: maximum asynchronous baud rate of 691,200 bps, maximum synchronous baud rate of 5,529,600 bps. One port is configurable as a clocked port.
- Six additional I/O lines are located on the programming port, can be used as I/O lines when the programming port is not being used for programming or in-circuit debugging—one synchronous serial port can also be used as two general CMOS inputs and one general CMOS output, and there are two additional inputs and one additional output.

Appendix A, “RabbitCore RCM2200 Specifications,” provides detailed specifications for the RCM2200.

In addition, three different RCM2200 models are available. A variant of the RCM2200, the RCM2300, omits the Ethernet connectivity but offers a much smaller footprint, one-half the size of the RCM2200.

1.2 Advantages of the RCM2200

- 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 for network connectivity, royalty-free TCP/IP software.

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.

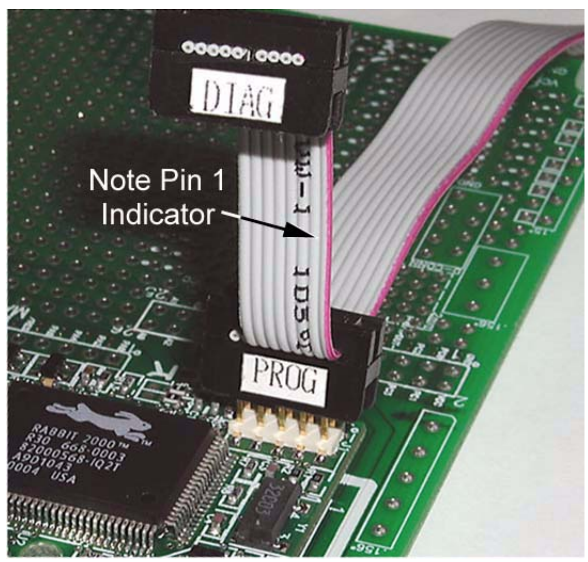


Figure 2. Connect Programming Cable to RCM2200

> **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.

3. Running Sample Programs

To develop and debug programs for the RCM2200 (and for all other Rabbit hardware), you must install and use Dynamic C. This chapter provides a tour of the sample programs for the RCM2200.

3.1 Sample Programs

To help familiarize you with the RCM2200 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 RC M2200'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 program, make sure that your RCM2200 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.

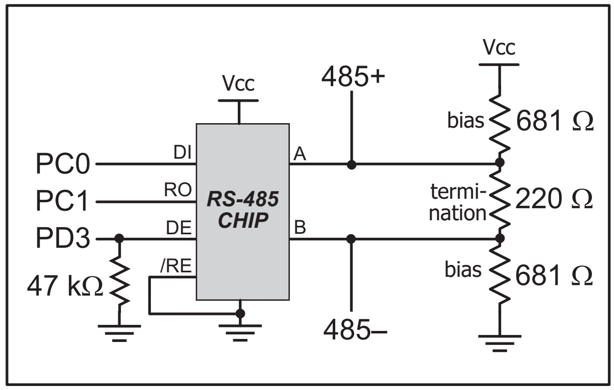
Sample programs are provided in the Dynamic C **`SAMPLES`** folder. Two folders contain sample programs that illustrate features unique to the RCM2200.

- **`RCM2200`**—Demonstrates the basic operation and the Ethernet functionality of the RCM2200.
- **`TCPIP`**—Demonstrates more advanced TCP/IP programming for Rabbit's Ethernet-enabled Rabbit-based boards.

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

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.

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.

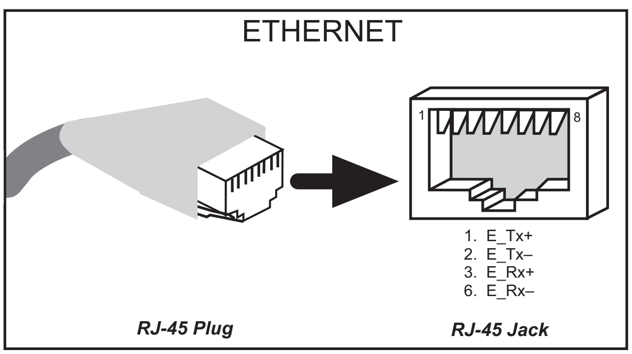


Figure 6. RJ-45 Ethernet Port Pinout

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.

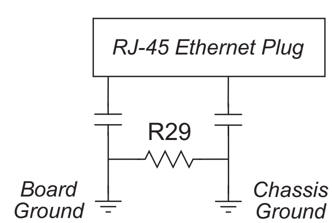


Figure 7. Isolation Resistor R29

6. USING THE TCP/IP FEATURES

6.1 TCP/IP Connections

Programming and development can be done with the RCM2200 RabbitCore modules without connecting the Ethernet port to a network. However, if you will be running the sample programs that use the Ethernet capability or will be doing Ethernet-enabled development, you should connect the RCM2200 module's Ethernet port at this time.

Before proceeding you will need to have the following items.

- If you don't have Ethernet access, you will need at least a 10Base-T Ethernet card (available from your favorite computer supplier) installed in a PC.
- Two RJ-45 straight through Ethernet cables and a hub, or an RJ-45 crossover Ethernet cable.

The Ethernet cables and Ethernet hub are available from Rabbit in a TCP/IP tool kit. More information is available at www.rabbit.com.

1. Connect the AC adapter and the programming cable as shown in Chapter 2, "Getting Started."
2. Ethernet Connections

   There are four options for connecting the RCM2200 module to a network for development and runtime purposes. The first two options permit total freedom in selecting network addresses and use of the "network," as no action can interfere with other users. We recommend one of these options for initial development.

- **No LAN —** The simplest alternative for desktop development. Connect the RCM2200's Ethernet port directly to the PC's network interface card using an RJ-45 *crossover cable*. A crossover cable is a special cable that flips some connections between the two connectors and permits direct connection of two client systems. A standard RJ-45 network cable will not work for this purpose.
- **Micro-LAN —** Another simple alternative for desktop development. Use a small Ethernet 10Base-T hub and connect both the PC's network interface card and the RCM2200's Ethernet port to it, using standard network cables.

6.3 IP Addresses Explained

IP (Internet Protocol) addresses are expressed as 4 decimal numbers separated by periods, for example:

216.103.126.155

10.1.1.6

Each decimal number must be between 0 and 255. The total IP address is a 32-bit number consisting of the 4 bytes expressed as shown above. A local network uses a group of adjacent IP addresses. There are always 2^N IP addresses in a local network. The netmask (also called subnet mask) determines how many IP addresses belong to the local network. The netmask is also a 32-bit address expressed in the same form as the IP address. An example netmask is:

255.255.255.0

This netmask has 8 zero bits in the least significant portion, and this means that 2^8 addresses are a part of the local network. Applied to the IP address above (216.103.126.155), this netmask would indicate that the following IP addresses belong to the local network:

216.103.126.0

216.103.126.1

216.103.126.2

etc.

216.103.126.254

216.103.126.255

The lowest and highest address are reserved for special purposes. The lowest address (216.103.126.0) is used to identify the local network. The highest address (216.103.126.255) is used as a broadcast address. Usually one other address is used for the address of the gateway out of the network. This leaves 256 - 3 = 253 available IP addresses for the example given.

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**.

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.

B.1 Prototyping Board

The Prototyping Board included in the Development Kit makes it easy to connect an RCM2200 module to a power supply and a PC workstation for development. It also provides some basic I/O peripherals (switches and LEDs), as well as a prototyping area for more advanced hardware development.

For the most basic level of evaluation and development, the Prototyping Board can be used without modification.

As you progress to more sophisticated experimentation and hardware development, modifications and additions can be made to the board without modifying or damaging the RCM2200 module itself.

The Prototyping Board is shown below in Figure B-1, with its main features identified.

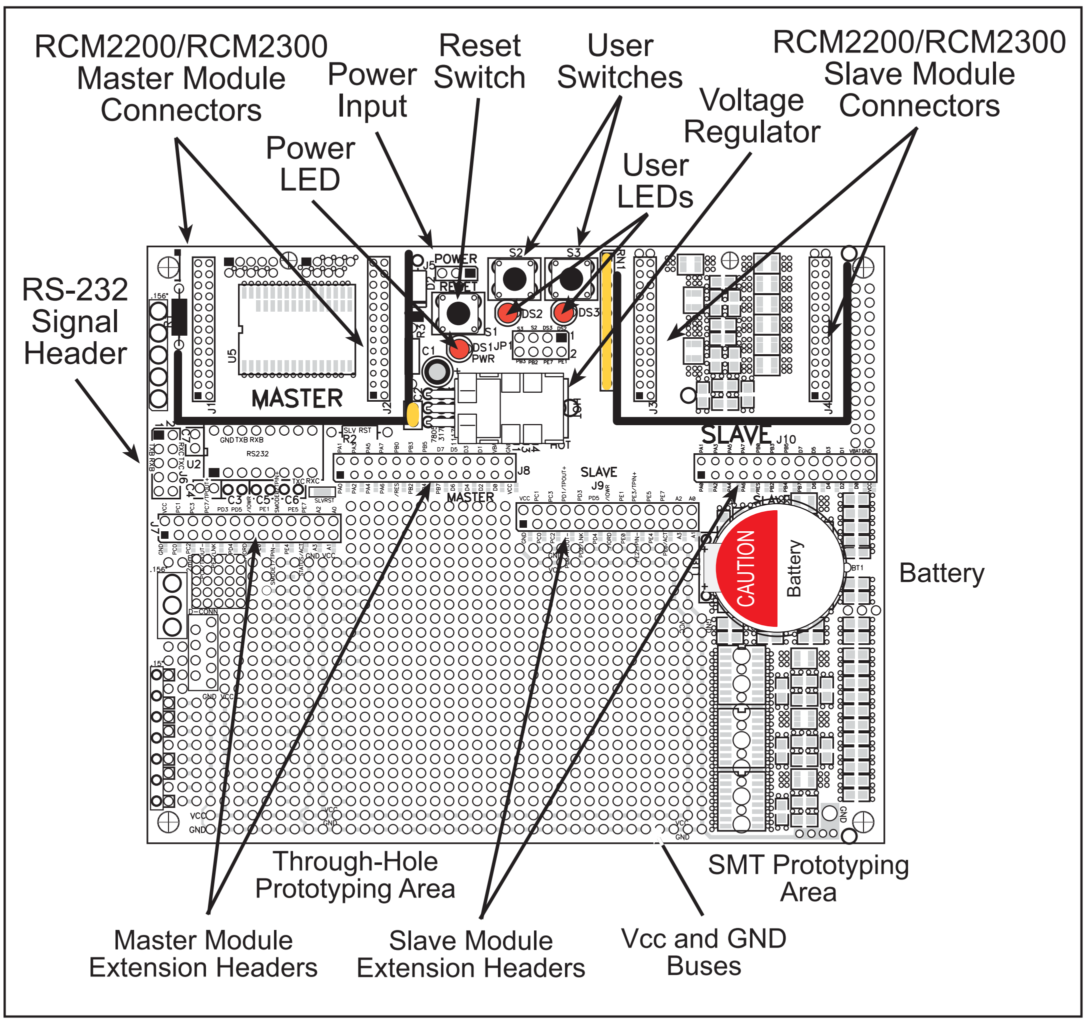


Figure B-1. RCM2200/RCM2300 Prototyping Board

B.2 Mechanical Dimensions and Layout

Figure B-2 shows the mechanical dimensions and layout for the RCM2200 Prototyping Board.

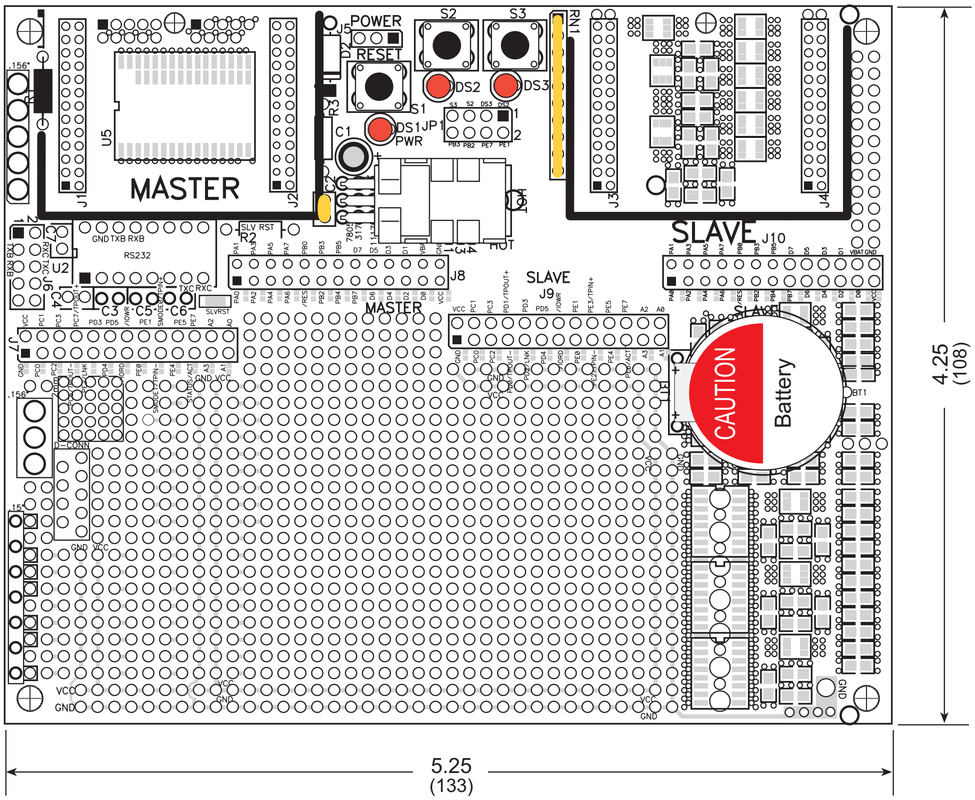


Figure B-2. RCM2200 Prototyping Board Dimensions

Table B-1 lists the electrical, mechanical, and environmental specifications for the Prototyping Board.

Table B-1. RCM2200 Prototyping Board Specifications

Parameter	Specification
Board Size	4.25" × 5.25" × 1.00" (108 mm × 133 mm × 25 mm)
Operating Temperature	–40°C to +70°C
Humidity	5% to 95%, noncondensing
Input Voltage	7.5 V to 25 V DC
Maximum Current Draw (including user-added circuits)	1 A at 12 V and 25°C, 0.7 A at 12 V and 70ºC
Prototyping Area	2.4" × 4.0" (61 mm × 102 mm) throughhole, 0.1" spacing, additional space for SMT components
Standoffs/Spacers	4, accept 6-32 × 3/8 screws

To maximize the availability of RCM2200 resources, the demonstration hardware (LEDs and switches) on the Prototyping Board may be disconnected. This is done by cutting the traces below the silk-screen outline of header JP1 on the bottom side of the Prototyping Board. Figure B-4 shows the four places where cuts should be made. Cut the traces between the rows as shown. An exacto knife would work nicely to cut the traces. Alternatively, a small standard screwdriver may be carefully and forcefully used to wipe through the PCB traces.Use jumpers across the positions on JP1 if you need to reconnect any of the devices later on.

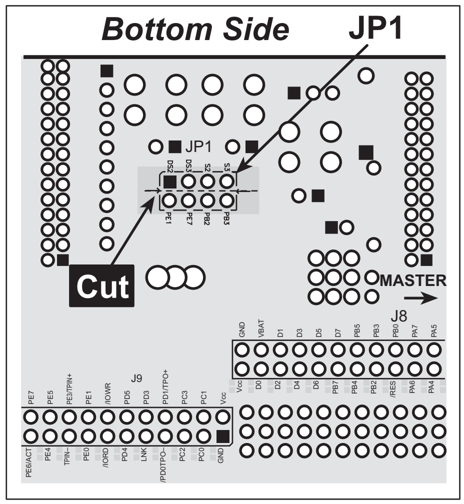


Figure B-4. Where to Cut Traces to Permanently Disable Demonstration Hardware on Prototyping Board

The power LED (PWR) and the RESET switch remain connected. Jumpers across the appropriate pins on header JP1 can be used to reconnect specific demonstration hardware later if needed.

Table B-2. Prototyping Board Jumper Settings

Header JP1	
Pins	**Description**
1–2	PE1 to LED DS2
3–4	PE7 to LED DS3
5–6	PB2 to Switch S2
7–8	PB3 to Switch S3

Note that the pinout at location JP1 on the bottom side of the Prototyping Board (shown in Figure B-4) is a mirror image of the top-side pinout.

The Prototyping Board provides the user with RCM2200 connection points brought out conveniently to labeled points at headers J7 and J8 on the Prototyping Board. Small to medium 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.

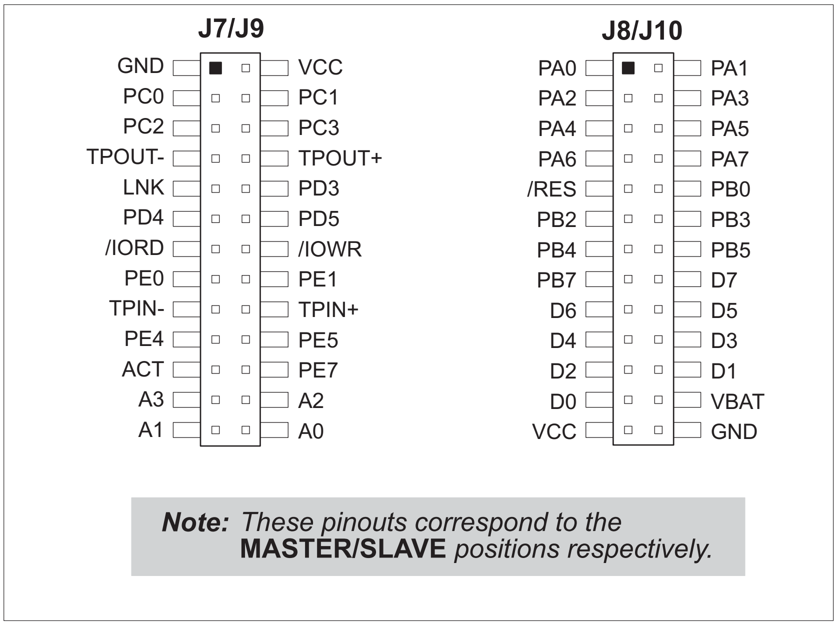


Figure B-5. RCM2200 Prototyping Board Pinout (Top View)

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.

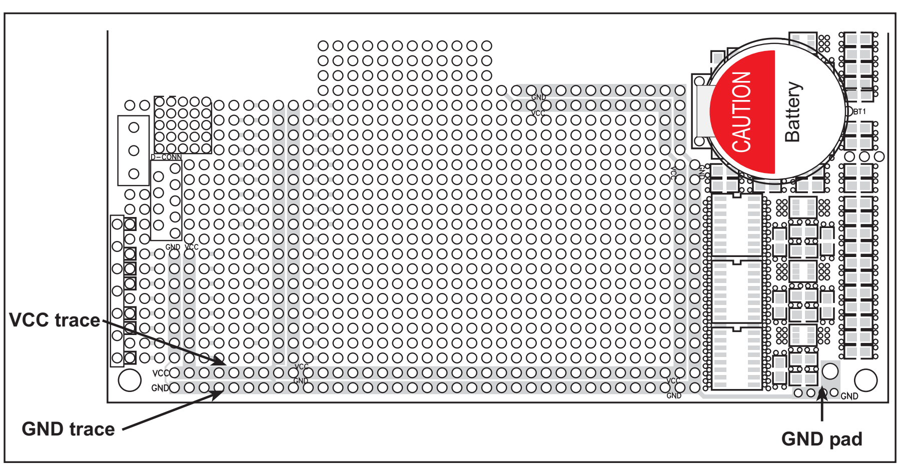


Figure B-6. VCC and GND Traces Along Edge of Prototyping Board

C.2 Chip Select Circuit

The RCM2200 has provision for battery backup, which kicks in to keep VRAM from dropping below 2 V.

When the RCM2200 is not powered, the battery keeps the SRAM memory contents and the real-time clock (RTC) going. The SRAM has a powerdown mode that greatly reduces power consumption. This powerdown mode is activated by raising the chip select (CS) signal line. Normally the SRAM requires Vcc to operate. However, only 2 V is required for data retention in powerdown mode. Thus, when power is removed from the circuit, the battery voltage needs to be provided to both the SRAM power pin and to the CS signal line. The CS control switch accomplishes this task for the CS signal line.

Figure C-3 shows a schematic of the chip select control switch.

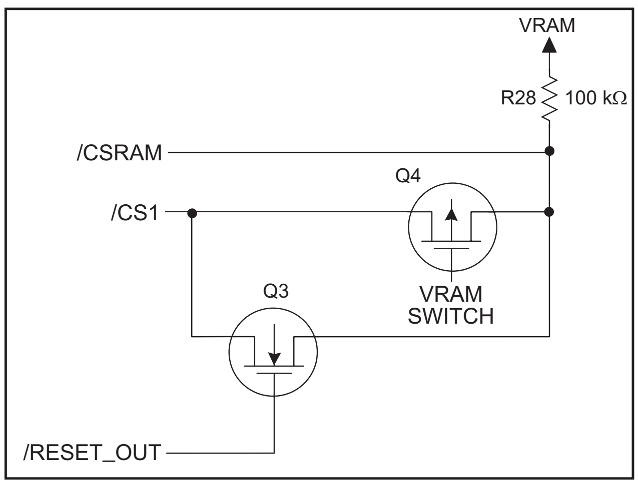


Figure C-3. Chip Select Control Switch

In a powered-up condition, the CS control switch must allow the processor's chip select signal /CS1 to control the SRAM's CS signal /CSRAM. So, with power applied, /CSRAM must be the same signal as /CS1, and with power removed, /CSRAM must be held high (but only needs to be as high as the battery voltage). Q3 and Q4 are MOSFET transistors with opposing polarity. They are both turned on when power is applied to the circuit. They allow the CS signal to pass from the processor to the SRAM so that the processor can periodically access the SRAM. When power is removed from the circuit, the transistors will turn off and isolate /CSRAM from the processor. The isolated /CSRAM line has a 100 kΩ pullup resistor to VRAM (R28). This pullup resistor keeps /CSRAM at the VRAM voltage level (which under no power condition is the backup battery's regulated voltage at a little more than 2 V).

Transistors Q3 and Q4 are of opposite polarity so that a rail-to-rail voltages can be passed. When the /CS1 voltage is low, Q3 will conduct. When the /CS1 voltage is high, Q4 will conduct. It takes time for the transistors to turn on, creating a propagation delay. This delay is typically very small, about 10 ns to 15 ns.

D.1 RS-232/RS-485 Serial Communication

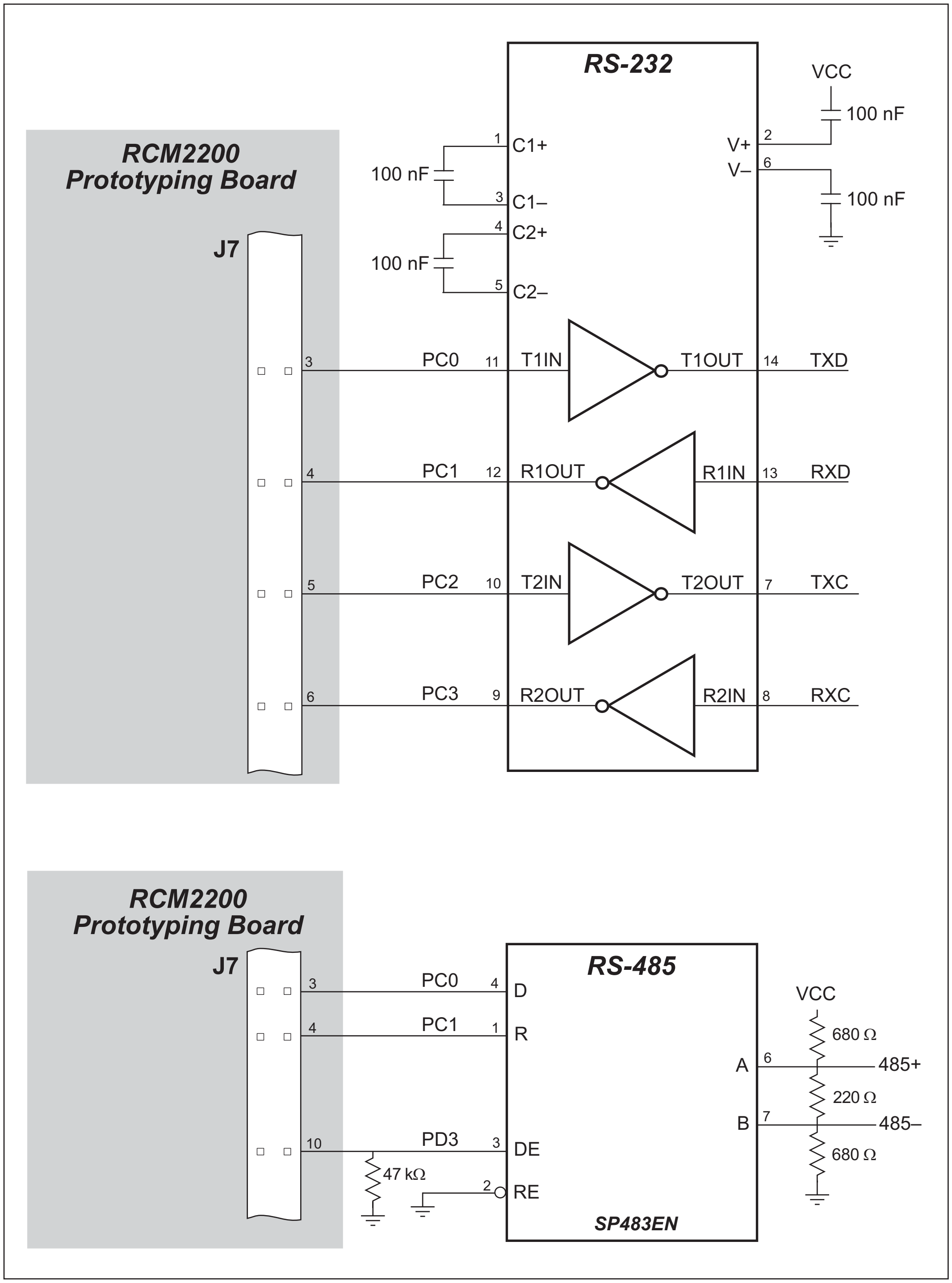


Figure D-1. Sample RS-232 and RS-485 Circuits

Sample Program: **`PUTS.C`** in **`SAMPLES\RCM2200`**.