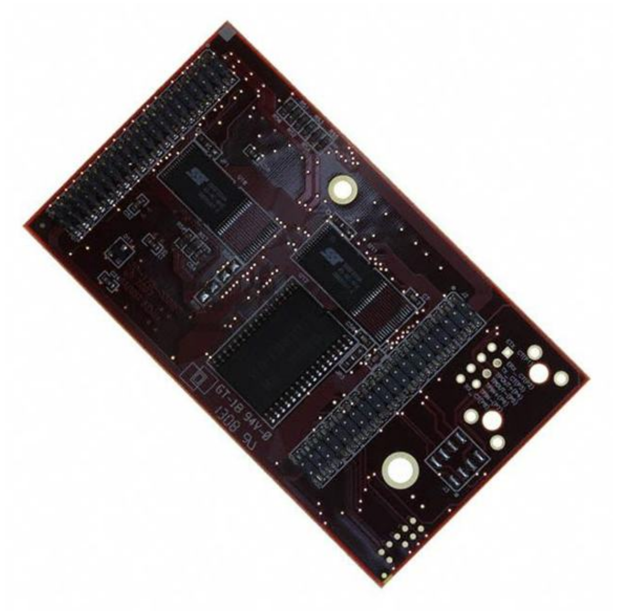


Welcome to E-XFL.COM

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 2x20
Size / Dimension	2" x 3.5" (51mm x 89mm)
Operating Temperature	-40°C ~ 85°C
Purchase URL	https://www.e-xfl.com/product-detail/digi-international/20-101-0436

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.

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 It reduces the number of outdated manuals we have to discard from stock as well, and it makes providing a complete library of manuals an almost cost-free option.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.

2.1.2 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.)

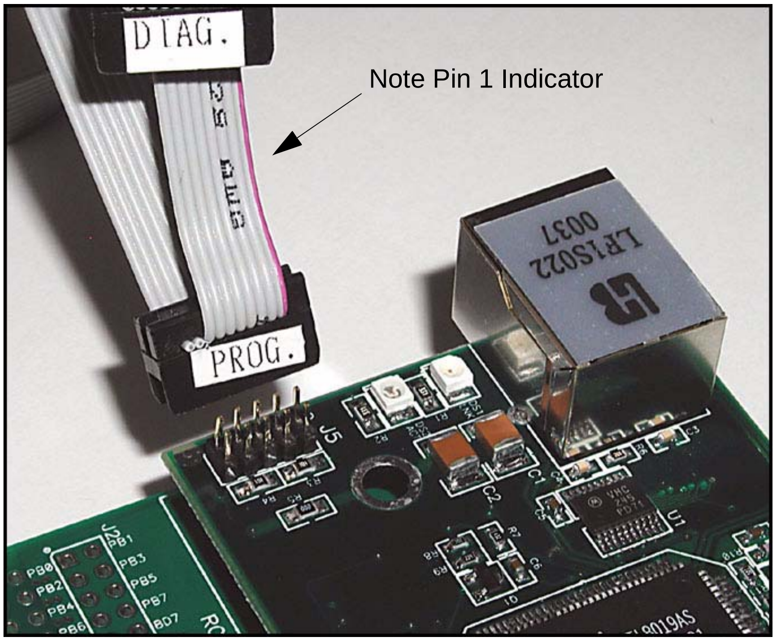


Figure 3. Attaching Programming Cable to the RCM2100

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.

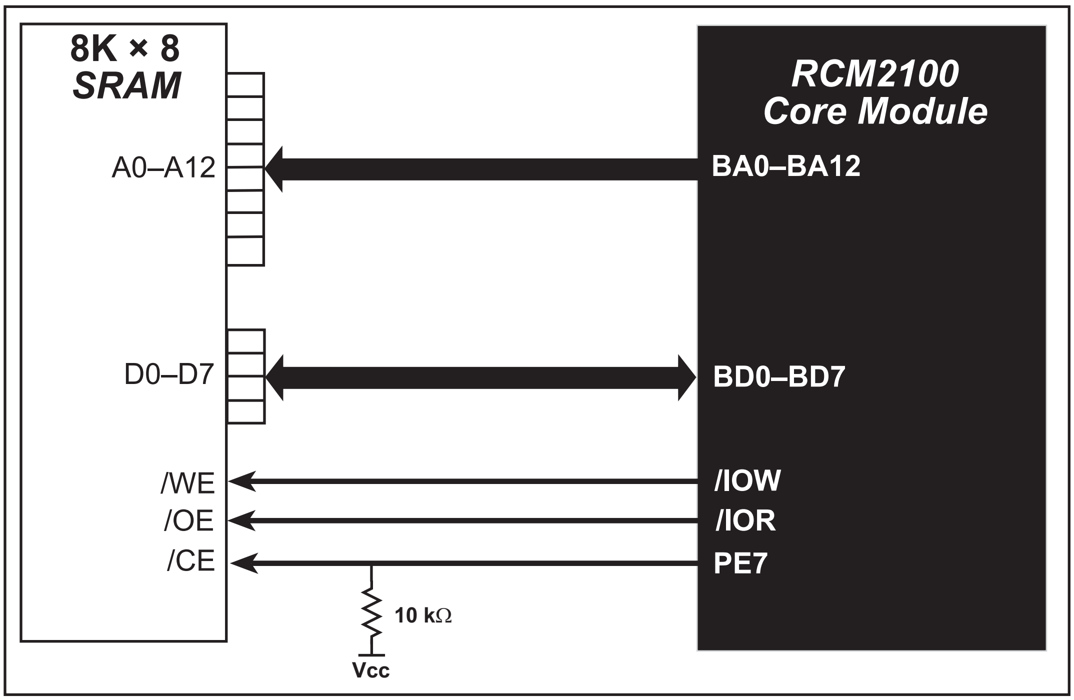
NOTE: Some PCs now come equipped only with a USB port. It may be possible to use an RS-232/USB converter (Part No. 540-0070) with the programming cable supplied with the RCM2100 Development Kit. Note that not all RS-232/USB converters work with Dynamic C.

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.

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***.

Table 2. RCM2100 Pinout Configurations

Pin		Pin Name	Default Use	Alternate Use	Notes
Header J1	1	VCC			
	2	GND			
	3	PCLK	Output (Internal Clock)	Output	Turned off in software
	4–11	PA[7:0]	Parallel I/O	Slave port data bus SD0–SD7	
	12–24	BA[12:0]	Output		Buffered Rabbit 2000 address bus
	25	PC0	Output	TXD	
	26	PC1	Input	RXD	
	27	PC2	Output	TXC	
	28	PC3	Input	RXC	
	29	PC4	Output	TXB	
	30	PC5	Input	RXB	
	31	PC6	Output	TXA	Connected to programming port
	32	PC7	Input	RXA	
	33–36	PD[0:3]	Bitwise or parallel programmable I/O, can be driven or open-drain output		16 mA sourcing and sinking current at full AC switching speed
	37	PD4		ATXB output	Ethernet chip RSTDRV
	38	PD5		ARXB input	Ethernet chip BD5
	39	PD6		ATXA output	Ethernet chip BD6
	40	PD7		ARXA input	Ethernet chip BD7

6.2 TCP/IP Primer on IP Addresses

Obtaining IP addresses to interact over an existing, operating, network can involve a number of complications, and must usually be done with cooperation from your ISP and/or network systems administrator. For this reason, it is suggested that the user begin instead by using a direct connection between a PC and the RCM2100 board using an Ethernet crossover cable or a simple arrangement with a hub. (A crossover cable should not be confused with regular straight through cables.)

In order to set up this direct connection, the user will have to use a PC without networking, or disconnect a PC from the corporate network, or install a second Ethernet adapter and set up a separate private network attached to the second Ethernet adapter. Disconnecting your PC from the corporate network may be easy or nearly impossible, depending on how it is set up. If your PC boots from the network or is dependent on the network for some or all of its disks, then it probably should not be disconnected. If a second Ethernet adapter is used, be aware that Windows TCP/IP will send messages to one adapter or the other, depending on the IP address and the binding order in Microsoft products. Thus you should have different ranges of IP addresses on your private network from those used on the corporate network. If both networks service the same IP address, then Windows may send a packet intended for your private network to the corporate network. A similar situation will take place if you use a dial-up line to send a packet to the Internet. Windows may try to send it via the local Ethernet network if it is also valid for that network.

The following IP addresses are set aside for local networks and are not allowed on the Internet: 10.0.0.0 to 10.255.255.255, 172.16.0.0 to 172.31.255.255, and 192.168.0.0 to 192.168.255.255.

The RCM2100 board uses a 10Base-T type of Ethernet connection, which is the most common scheme. The RJ-45 connectors are similar to U.S. style telephone connectors, except they are larger and have 8 contacts.

An alternative to the direct connection using a crossover cable is a direct connection using a hub. The hub relays packets received on any port to all of the ports on the hub. Hubs are low in cost and are readily available. The RCM2100 board uses 10 Mbps Ethernet, so the hub or Ethernet adapter must be either a 10 Mbps unit or a 10/100 unit that adapts to either 10 or 100 Mbps.

In a corporate setting where the Internet is brought in via a high-speed line, there are typically machines between the outside Internet and the internal network. These machines include a combination of proxy servers and firewalls that filter and multiplex Internet traffic. In the configuration below, the RCM2100 board could be given a fixed address so any of the computers on the local network would be able to contact it. It may be possible to configure the firewall or proxy server to allow hosts on the Internet to directly contact the controller, but it would probably be easier to place the controller directly on the external network outside of the firewall. This avoids some of the configuration complications by sacrificing some security.

6.9 Run the `PINGME.C` Sample Program

Connect the crossover cable from your computer's Ethernet port to the RCM2100 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 RCM2100 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 RCM2100 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\RCM2100\`** folder.

6.10.1 Sample Program: PINGLED.C

One of the RCM2100's most important features is the availability of the built-in Ethernet port. This program makes the simplest possible use of the network port by "pinging" a remote system and using LEDs to report the status of the ping attempt and its return.

Compile & Run Program

Open the **`PINGLED.C`** sample program. Press **F9** to compile and run the program.

Each time the program sends a ping to the remote address, LED DS2 on the Prototyping Board will flash. Each time a successful return from a ping attempt is received, LED DS3 will flash.

If the ping return is unsuccessful (i.e., the remote system does not exist or does not acknowledge the ping within the timeout period), DS3 will not flash.

With short ping times, as will be encountered in most micro-LAN and LAN settings, the two LEDs should flash almost in parallel as pings are sent and returned.

You can modify the **`#define PING_DELAY`** statement to change the amount of time between the outgoing pings.

6.11 Where Do I Go From Here?

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/.

If the sample programs ran fine, you are now ready to go on.

Additional sample programs are described in the ***Dynamic C TCP/IP User's Manual***.

Please refer to the ***Dynamic C TCP/IP User's Manual*** to develop your own applications. ***An Introduction to TCP/IP*** provides background information on TCP/IP, and is available on the CD and on our Web site.

A.1 Electrical and Mechanical Characteristics

Figure A-1 shows the mechanical dimensions for the RCM2100.

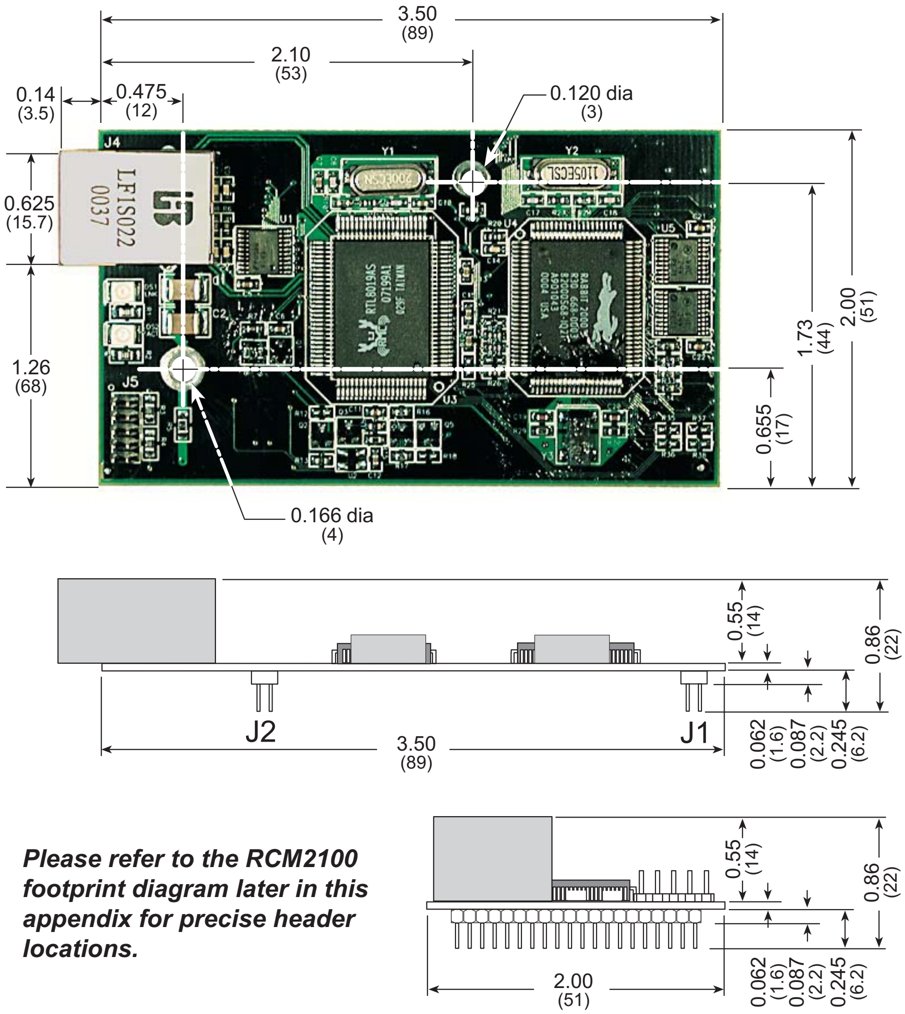


Figure A-1. RCM2100 Dimensions

NOTE: All measurements are in inches followed by millimeters enclosed in parentheses. All dimensions have a manufacturing tolerance of ±0.01" (0.25 mm).

Table A-1 lists the electrical, mechanical, and environmental specifications for the RCM2100.

Table A-1. RCM2100 Specifications

Parameter	RCM2100	RCM2110	RCM2120	RCM2130
Microprocessor	Rabbit® 2000 at 22.1 MHz			
Ethernet Port	10/100-compatible with 10Base-T interface, RJ-45, 2 LEDs		None	
Flash Memory	Two 256K × 8	One 256K × 8	Two 256K × 8	One 256K × 8
SRAM	512K × 8	128K × 8	512K × 8	128K × 8
Backup Battery	Connection for user-supplied backup battery (to support RTC and SRAM)			
General-Purpose I/O	34 parallel I/0 lines grouped in five 8-bit ports (and shared with serial ports): • 20 configurable I/O • 8 fixed inputs • 6 fixed outputs		40 parallel I/0 lines grouped in five 8-bit ports (and shared with serial ports): • 26 configurable I/O • 8 fixed inputs • 6 fixed outputs	
Additional Inputs	2 startup mode (for master/slave), reset			
Additional Outputs	Status, clock, watchdog, reset			
Memory, I/O Interface	13 address lines, 8 data lines, I/O read/write, buffer enable			
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 RCM2100 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, 140 mA			
Operating Temperature	–40°C to +70°C		–40°C to +85°C	
Humidity	5% to 95%, noncondensing			
Connectors	Two IDC headers 2 × 20, 2 mm pitch			
Board Size	2.00" × 3.50" × 0.86" (51 mm × 89 mm × 22 mm)		2.00" × 3.50" × 0.5" (51 mm × 89 mm × 13 mm)	

A.2 Bus Loading

You must pay careful attention to bus loading when designing an interface to the RabbitCore RCM2100. This section provides bus loading information for external devices.

Table A-2 lists the capacitance for the various RCM2100 I/O ports.

Table A-2. Capacitance of RCM2100 I/O Ports

I/O Ports	Input Capacitance		Output Capacitance	
	Typ.	Max.	Typ.	Max.
Parallel Ports A to E	6 pF	12 pF	10 pF	14 pF
Data Lines BD0–BD7	12 pF	18 pF	18 pF	22 pF
Address Lines BA0–BA12	—	—	8 pF	12 pF

Table A-3 lists the external capacitive bus loading for the various RCM2100 output ports. Be sure to add the loads for the devices you are using in your custom system and verify that they do not exceed the values in Table A-3.

Table A-3. External Capacitive Bus Loading -40°C to +70°C

Output Port	Clock Speed (MHz)	Maximum External Capacitive Loading (pF)
A[12:0] D[7:0]	22.1	50
PD[3:0]	22.1	100
PA[7:0] PB[7,6] PC[6,4,2,0] PD[7:4]* PE[7:0]	22.1	90
All data, address, and I/O lines with clock doubler disabled	11.0592	100

* The Parallel Port D outputs (PD[7:4]) are available only on the RCM2120 and the RCM2130 models.

A.5 Jumper Configurations

Figure A-5 shows the header locations used to configure the various RCM2100 options via jumpers.

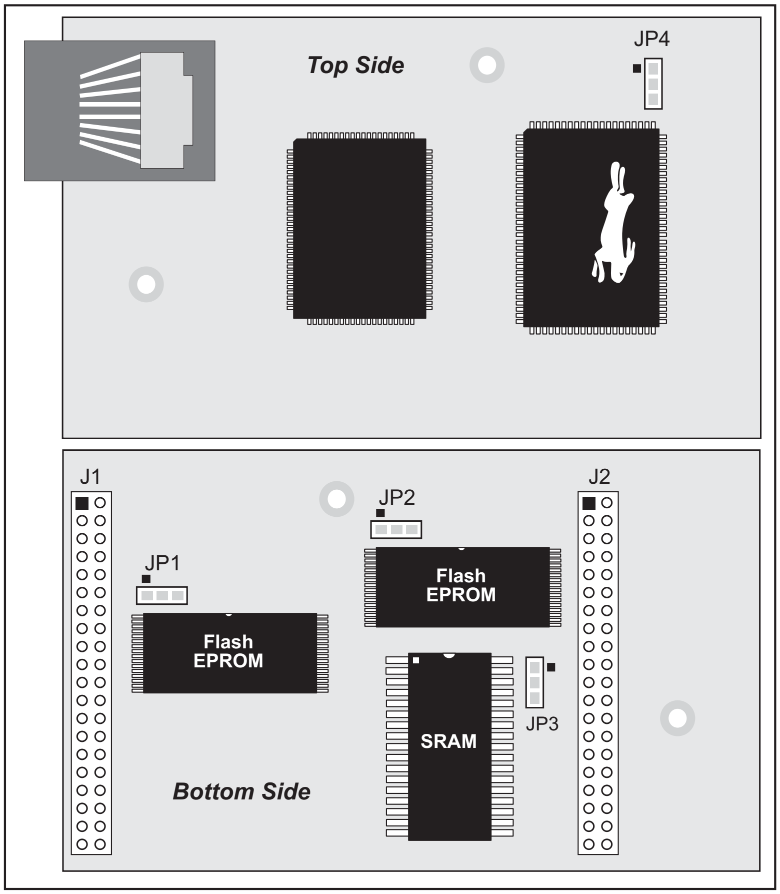


Figure A-5. Location of RCM2100 Configurable Positions

B.1.1 Prototyping Board Features

Power Connection. A 3-pin header is provided for connection of a power supply. Note that it is symmetrical, with both outer pins connected to ground and the center pin connected to the raw V+ input. The cable of the wall transformer provided with the North American version of the Development Kit ends in a connector that is correctly connected in either orientation.

Users providing their own power supply should ensure that it delivers 9–24 V DC at not less than 500 mA. The voltage regulator will get warm in use, but lower supply voltages will reduce thermal dissipation from the device.

Regulated Power Supply. The raw DC voltage provided at the POWER IN jack is routed to a 5 V linear voltage regulator, which provides stable power to the RCM2100 module and the Prototyping Board. A Shottky diode protects the power supply against damage from reversed raw power connections.

Power LED. The power LED lights whenever power is connected to the Prototyping Board.

Reset Switch. A momentary-contact, normally open switch is connected directly to the RCM2100's **/RES_IN** pin. Pressing the switch forces a hardware reset of the system.

I/O Switches & LEDs. Two momentary-contact, normally open switches are connected to the PB2 and PB3 pins of the RCM2100 module, and may be read as inputs by sample applications.

Two LEDs are connected to the PA0 and PA1 pins of the module, and may be driven as output indicators by sample applications. (Two more LEDs, driven by PA2 and PA3, may be added to the Prototyping Board for additional outputs.)

All the LEDs are connected through JP1, which has traces shorting adjacent pads together. These traces may be cut to disconnect the LEDs, and an 8-pin header soldered into JP1 to permit their selective reconnection with jumpers. See Figure B-4 for details.

Expansion Areas. The Prototyping Board is provided with several unpopulated areas for expansion of I/O and interfacing capabilities. See the next section for details.

Prototyping Area. A generous prototyping area has been provided for the installation of through-hole components. Vcc (5 V DC) and Ground buses run around the edge of this area. An area for surface-mount devices is provided to the right of the through-hole area. (Note that there are SMT device pads on both top and bottom of the Prototyping Board.)

To maximize the availability of RCM2100 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. 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.

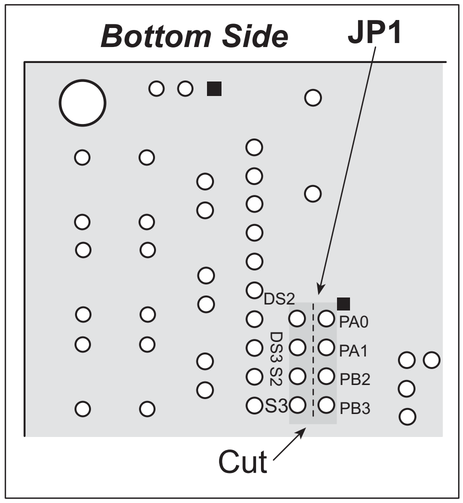


Figure B-4. Where to Cut Traces to Permanently Disable Demonstration Hardware on Prototyping Board

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 JP2	
Pins	**Description**
1–2	PA0 to LED DS2
3–4	PA1 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 RCM2100 connection points brought out conveniently to labeled points at headers J2 and J4 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 J2 and J4. The holes are spaced at 0.1" (2.5 mm),

APPENDIX C. POWER SUPPLY

Appendix C provides information on the current and power supply requirements of the RCM2100, and some background on the chip select and battery-backup circuits used in power management.

C.1 Power Supplies

The RCM2100 requires a regulated 5 V ± 0.25 V DC power source. The RCM2100 design presumes that the voltage regulator is on the user board, and that the power is made available to the RCM2100 board through headers J1 and J2.

An RCM2100 with no loading at the outputs operating at 22.1 MHz typically draws 140 mA. The RCM2100 will consume 13 mA to 15 mA of additional current when the programming cable is used to connect J5 to a PC.

C.1.1 Batteries and External Battery Connections

The RCM2100 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 J2, 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 RCM2100 powered down.

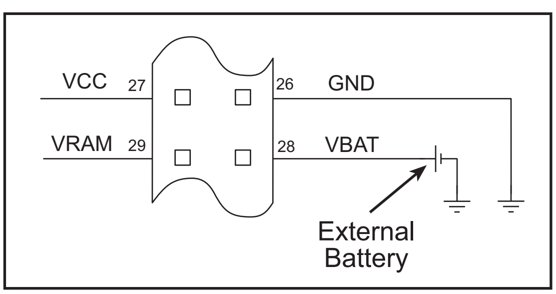


Figure C-1. External Battery Connections at Header J2

D.3 External Memory

The sample circuit can be used with an external 64K memory device. Larger SRAMs can be written to using this scheme by using other available Rabbit 2000 ports (parallel ports A to E) as address lines.

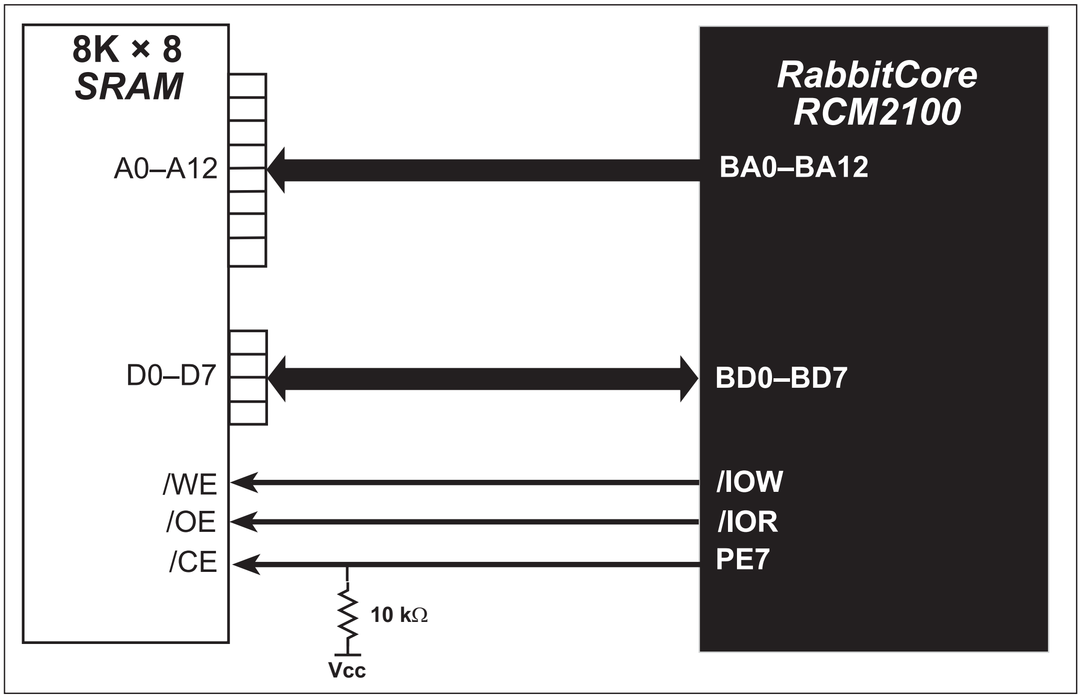


Figure D-4. Sample External Memory Connections

Sample Program: **`EXTSRAM2.C`** in **`SAMPLES\RCM2100`**.

D.4 D/A Converter

The output will initially be 0 V to -10.05 V after the first inverting op-amp, and 0 V to +10.05 V after the second inverting op-amp. All lows produce 0 V out, FF produces 10 V out. The output can be scaled by changing the feedback resistors on the op-amps. For example, changing 5.11 kΩ to 2.5 kΩ will produce an output from 0 V to -5 V. Op-amps with a very low input offset voltage are recommended.

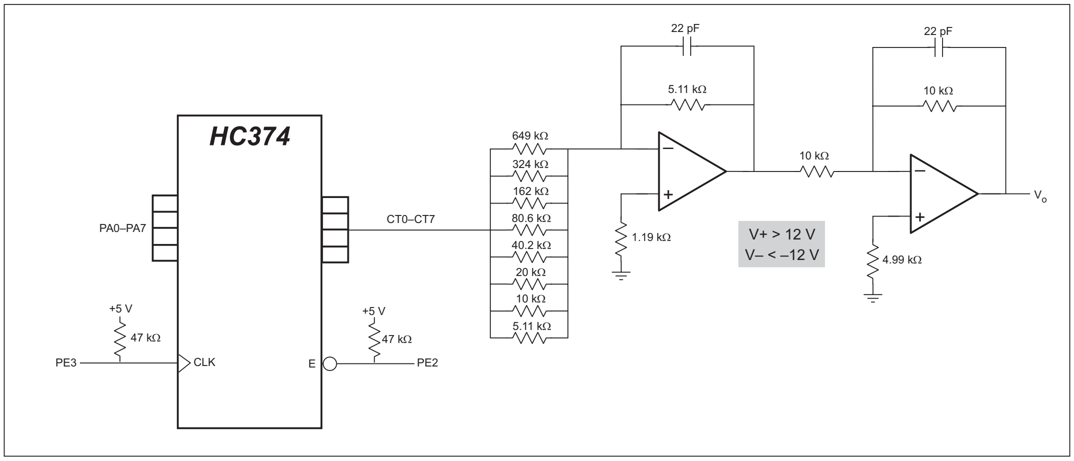


Figure D-5. Sample D/A Converter Connections