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 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/101-0446

TABLE OF CONTENTS

1.4 How to Use This Manual

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

1.4.1 Additional Product Information

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

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

1.4.2 Using Online Documentation

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

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

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

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.)

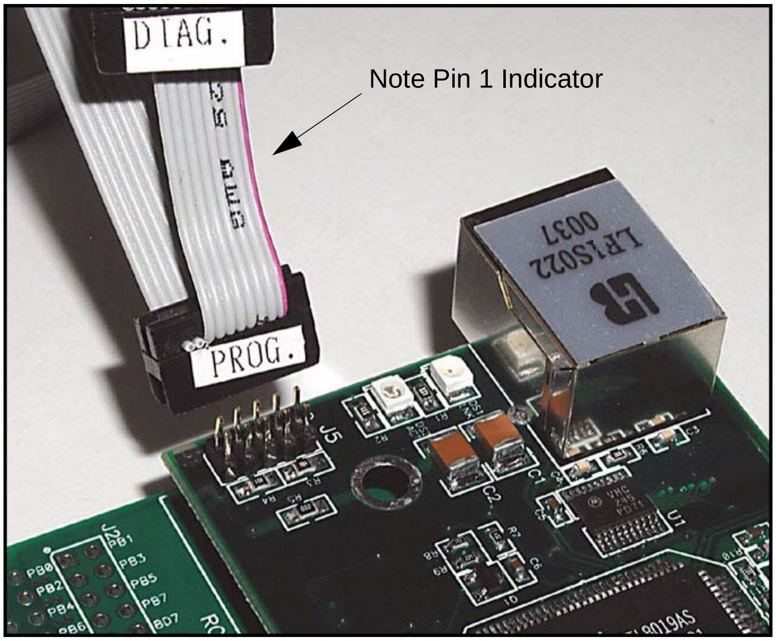


Figure 3. Attaching Programming Cable to the RCM2100

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.

4. HARDWARE REFERENCE

Chapter 3 describes the hardware components and principal hardware subsystems of the RabbitCore RCM2100. Appendix A, “RabbitCore RCM2100 Specifications,” provides complete physical and electrical specifications.

4.1 RCM2100 Digital Inputs and Outputs

Figure 5 shows the subsystems designed into the RCM2100 modules.

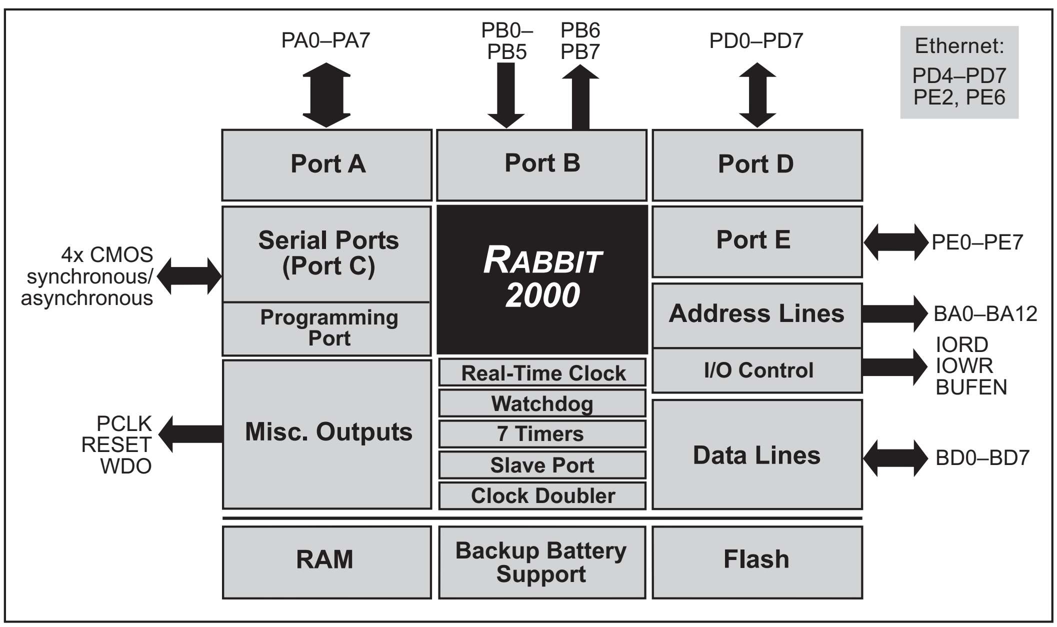


Figure 5. Rabbit Subsystems

4.1.1 Dedicated Inputs

PB0 and PB1 are designated as inputs because the Rabbit 2000 is operating in an asynchronous mode. Four of the input-only pins are located on PB2–PB5. These pins are 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 (SD0–SD7), which is the alternate assignment for parallel port A. When Port C is used as a parallel port, PC1, PC3, PC5, and PC7 are inputs only. These pins can alternately be selectively enabled to serve as the serial data inputs for Serial Ports D, C, B, and A.

4.1.2 Dedicated Outputs

Two of the output-only pins are located on PB6–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. When Port C is used as a parallel port, PC0, PC2, PC4 and PC6 are outputs only. These pins can alternately serve as the serial data outputs for Serial Ports D, C, B, and A.

4.1.3 Memory I/O Interface

Thirteen of the Rabbit 2000 buffered address lines (A0–A12) and all the buffered data lines (D0–D7) are available as outputs. I/0 write (/IOWR), I/0 read (/IORD), buffer enable (/BUFEN), and Watchdog Output (/WDO) are also available for interfacing to external devices.

The STATUS output has three different programmable functions:

1. It can be driven low on the first op code fetch cycle.
2. It can be driven low during an interrupt acknowledge cycle.
3. It can also serve as a general-purpose output.

The output clock is available on the PCLK pin. The primary function of PCLK is as a peripheral clock or a peripheral clock ÷ 2, but PCLK can instead be used as a digital output. PCLK can also be disabled by removing R20 if there is a need to reduce radiated emissions. Removing R20 will disable the PCLK output on pin 3 of header J1. Alternatively, PCLK can be disabled in software using Dynamic C version 7.03 or later.

4.1.4 Additional I/0

Two status mode pins, SMODE0 and SMODE1, are available as inputs. The logic state of these two pins determines the startup procedure after a reset.

/RES_IN is an external input used to reset the Rabbit 2000 microprocessor and the RabbitCore RCM2100 memory. /RES_OUT is an output from the reset circuitry that can be used to reset other peripheral devices.

4.2 Serial Communication

The RCM2100 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 RCM2100 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 32. 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 be operated alternately 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 1/4 of the system clock frequency, or 5.52 Mbps for a 22.1 MHz clock speed.

4.2.2 Ethernet Port

Figure 7 shows the pinout for the RJ-45 Ethernet port (J4). Note that there are two standards for numbering the pins on this connector—the convention used here, and numbering in reverse to that used here.

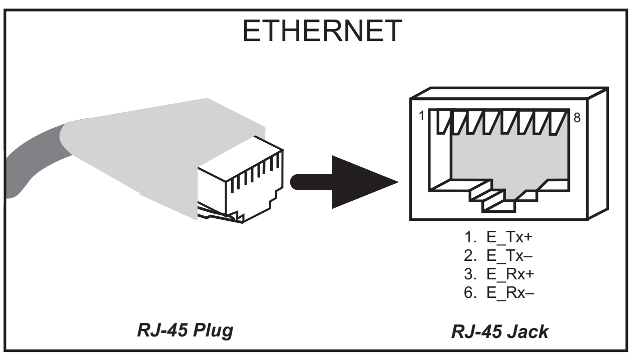


Figure 7. RJ-45 Ethernet Port Pinout

The transformer/connector assembly ground is connected to the RCM2100 printed circuit board digital ground via a 0 Ω resistor "jumper," R5, as shown in Figure 8.

The factory default is for the 0 Ω resistor "jumper" at R5 to be installed. In high-noise environments, it may be useful to ground the transformer/connector assembly directly through the chassis ground. This will be especially helpful to minimize EMI problems.

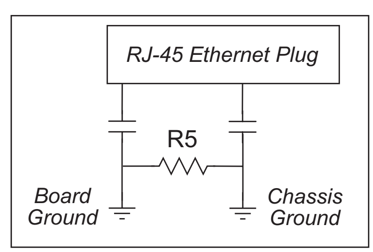


Figure 8. Isolation Resistor R5

4.5.2 Spectrum Spreader

RCM2100 modules that have a Rabbit 2000 microprocessor labeled ***IQ4T*** (or higher) are equipped with a Rabbit 2000 microprocessor that has a spectrum spreader, which helps to mitigate EMI problems. By default, the spectrum spreader is on automatically for RCM2100 modules that carry the ***IQ4T*** (or higher) marking when used with Dynamic C 7.30 or later versions, but the spectrum spreader may also be turned off or set to a stronger setting. The means for doing so is through a simple configuration macro as shown below.

1. Select the “Defines” tab from the Dynamic C **Options > Project Options** menu.
2. Normal spreading is the default, and usually no entry is needed. If you need to specify normal spreading, add the line

   ```
   ENABLE_SPREADER=1
   ```

   For strong spreading, add the line

   ```
   ENABLE_SPREADER=2
   ```

   To disable the spectrum spreader, add the line

   ```
   ENABLE_SPREADER=0
   ```

   NOTE: The strong spectrum-spreading setting is usually not necessary for the RCM2100.

3. Click **OK** to save the macro. The spectrum spreader will now remain off whenever you are in the project file where you defined the macro.

There is no spectrum spreader functionality for RCM2100 modules that have a Rabbit 2000 microprocessor labeled ***IQ1T***, ***IQ2T***, or ***IQ3T***, or when using any RCM2100 with a version of Dynamic C prior to 7.30.

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 Semiconductor single-board computers and other single-board computers based on the Rabbit microprocessor. Chapter 4 provides the libraries and function calls related to the RCM2100.

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 in the real environment. A complete reference to Dynamic C is contained in the ***Dynamic C User's Manual***.

Dynamic C for Rabbit® processors uses the standard Rabbit programming interface. This is a 10-pin connector that connects to the Rabbit Serial Port A. It is possible to reset and cold-boot a Rabbit processor via the programming port. No software needs to be present in the target system. More details are available in the ***Rabbit 2000 Microprocessor User's Manual***.

Dynamic C cold-boots the target system and compiles the BIOS. The BIOS is a basic program of a few thousand bytes in length that provides the debugging and communication facilities that Dynamic C needs. Once the BIOS has been compiled, the user can compile his own program and test it. If the BIOS fails because the program stops running, a new cold boot and BIOS compile can be done at any time.

The BIOS can be customized by using **`#define`** options.

Dynamic C does not use **`include`** files, rather it has libraries that are used for the same purpose, that is, to supply function prototypes to programs before they are compiled. See Section 4.24, "Modules," in the ***Dynamic C User's Manual*** for more information.

Dynamic C supports assembly language, either as separate functions or as fragments embedded in C programs. Interrupt routines may be written in Dynamic C or in assembly language.

5.1.1 Using Dynamic C

You have a choice of doing your software development in the flash memory or in the SRAM included on the RCM2100. There are 512K or 256K bytes of flash memory and 512K or 128K bytes of SRAM. 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 RCM2100 and Dynamic C were designed to accommodate flash devices with various sector sizes.

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, 98, 2000, NT, Me, and XP. 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.
  - ▶ I^2C, 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, µC/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.

5.1.2 Early Versions of Dynamic C

If you are using Dynamic C version 7.04 or earlier, modify the BIOS source code as follows. Skip these three steps if your version of Dynamic C is 7.05 or later.

1. Open the BIOS source code file named **`RABBITBIOS.C`**, which can be found in the **`BIOS`** directory.
2. Change the line

```
#define USE115KBAUD 1   // set to 0 to use 57600 baud
```

   to read as follows.

```
#define USE115KBAUD 0   // set to 0 to use 57600 baud
```

3. Save the changes using **File > Save**.

Now press **<Ctrl-Y>**. You should receive the **`"BIOS successfully compiled …"`** message indicating that the target is now ready to compile a program.

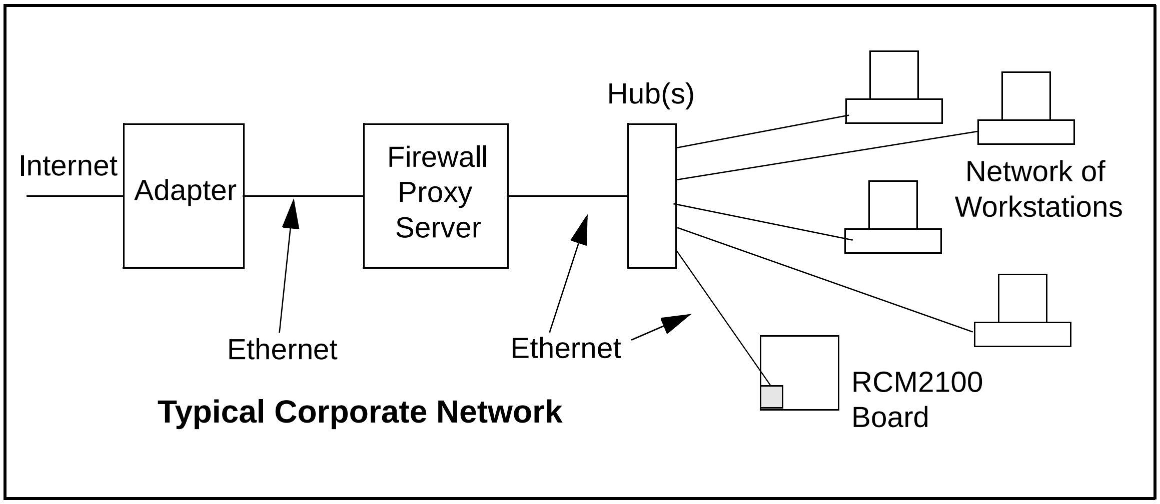
5.5 Upgrading Dynamic C

Dynamic C patches that focus on bug fixes are available from time to time. Check the Web site www.rabbit.com/support/ for the latest patches, workarounds, and bug fixes.

The default installation of a patch or bug fix is to install the file in a directory (folder) different from that of the original Dynamic C installation. Rabbit Semiconductor recommends using a different directory so that you can verify the operation of the patch without overwriting the existing Dynamic C installation. If you have made any changes to the BIOS or to libraries, or if you have programs in the old directory (folder), make these same changes to the BIOS or libraries in the new directory containing the patch. Do ***not*** simply copy over an entire file since you may overwrite a bug fix; of course, you may copy over any programs you have written. Once you are sure the new patch works entirely to your satisfaction, you may retire the existing installation, but keep it available to handle legacy applications.

5.5.1 Upgrades

Dynamic C installations are designed for use with the board they are included with, and are included at no charge as part of our low-cost kits. Dynamic C is a complete software development system, but does not include all the Dynamic C features. Rabbit Semiconductor also offers add-on Dynamic C modules containing the popular µC/OS-II real-time operating system, as well as PPP, Advanced Encryption Standard (AES), and other select libraries. In addition to the Web-based technical support included at no extra charge, a one-year telephone-based technical support module is also available for purchase.



If your system administrator can give you an Ethernet cable along with its IP address, the netmask and the gateway address, then you may be able to run the sample programs without having to set up a direct connection between your computer and the RCM2100 board. You will also need the IP address of the nameserver, the name or IP address of your mail server, and your domain name for some of the sample programs.

6.5 Dynamically Assigned Internet Addresses

In many instances, there are no fixed IP addresses. This is the case when, for example, you are assigned an IP address dynamically by your dial-up Internet service provider (ISP) or when you have a device that provides your IP addresses using the Dynamic Host Configuration Protocol (DHCP). The RCM2100 RabbitCore modules can use such IP addresses to send and receive packets on the Internet, but you must take into account that this IP address may only be valid for the duration of the call or for a period of time, and could be a private IP address that is not directly accessible to others on the Internet. These private addresses can be used to perform some Internet tasks such as sending e-mail or browsing the Web, but usually cannot be used to participate in conversations that originate elsewhere on the Internet. If you want to find out what this dynamically assigned IP address is, under Windows XP you can run the **`ipconfig`** program while you are connected and look at the interface used to connect to the Internet.

Many networks use private IP addresses that are assigned using DHCP. When your computer comes up, and periodically after that, it requests its networking information from a DHCP server. The DHCP server may try to give you the same address each time, but a fixed IP address is usually not guaranteed.

If you are not concerned about accessing the RCM2100 from the Internet, you can place the RCM2100 on the internal network using a private address assigned either statically or through DHCP.

6.6 Placing Your Device on the Network

In many corporate settings, users are isolated from the Internet by a firewall and/or a proxy server. These devices attempt to secure the company from unauthorized network traffic, and usually work by disallowing traffic that did not originate from inside the network. If you want users on the Internet to communicate with your RCM2100, you have several options. You can either place the RCM2100 directly on the Internet with a real Internet address or place it behind the firewall. If you place the RCM2100 behind the firewall, you need to configure the firewall to translate and forward packets from the Internet to the RCM2100.

6.8.1 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.

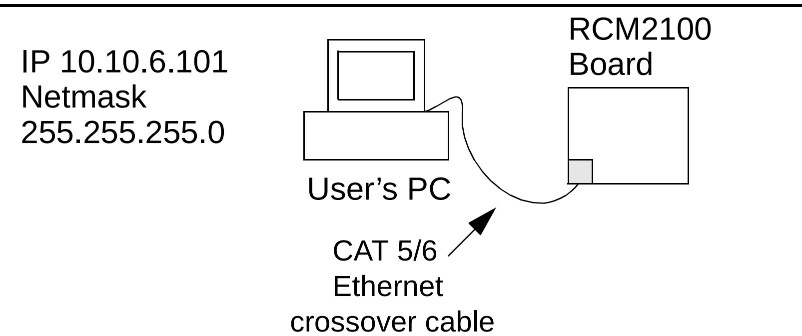
3. Now select the **IP Address** tab, and check **Specify an IP Address**, or select TCP/IP and click on "Properties" to assign an IP address to your computer (this will disable "obtain an IP address automatically"):

   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.



Direct Connection PC to RCM2100 Board

Program Description

For operation, network addresses must be correctly defined at the start of this program. The **`TCPCONFIG 1`** macro in the sample program provides default settings for **`MY_IP_ADDRESS`**, which is the address of the RCM2100 module, **`MY_NETMASK`**, and **`MY_GATEWAY`** (which needs to be defined if you wish to ping systems outside the local network). If you wish to ping systems using domain names instead of IP addresses, a valid DNS server address must be defined for **`MY_NAMESERVER`**. These TCP/IP settings can be changed as needed in the **`TCP_CONFIG.LIB`** library.

The IP address to be pinged is defined by **`PING_WHO`**. You will have to change this address and recompile the program to ping different addresses. (In most real-world applications, there should be some mechanism by which to dynamically define or select addresses.) This address may be defined as a numeric IP address. If a gateway to the Internet and a valid DNS server are specified, this definition may also be a fully-qualified domain name (such as "www.rabbit.com").

The program first defines three functions to control the LEDs—one to initialize them, and then one each to drive the "ping out" and "ping in" LEDs.

The program begins by calling the LED initialization function **`pingleds_setup( )`**. More importantly, it then calls **`sock_init( )`**, which initializes the packet driver and the TCP manager using the compiler defaults. This function must always be called before any other TCP/IP functions.

The program then resolves the address to be pinged into a numeric value. using the library function **`resolve()`**. If the defined address is numeric, it converts the define string into truly numeric form. If the address is a domain name, the function queries the indicated DNS server to obtain the numeric address. (If the function is unable to resolve the address—if, for example, the numeric address is incomplete or badly formed, or the DNS server is unable to identify the domain name—the program will print a message to the screen and terminate.)

The program then begins an endless loop using **`for(;;)`**. Within this loop, the program executes the following steps:

1. Calls **`tcp_tick()`** to perform the basic housekeeping functions for the socket;
2. As a costatement, waits for the duration of **`PING_DELAY`** (defined by default as 500 ms or one-half second), issues a ping to the resolved address using the **`_ping()`** function, and flashes LED DS2;
3. As a second costatement, checks for a ping return using the **`_chk_ping()`** function. If the ping is successful, the costatement flashes LED DS3.

If you uncomment the **`#VERBOSE`** define near the beginning of the program, the ping return costatement will also print a message to the screen indicating each successful ping.

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 22.1 MHz CPU clock and capacitive loading on address and data lines of less than 100 pF per pin. Pins A0–A12 and D0–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 RCM2100.

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, 1, 0]	8/8	12/12
PC [6, 4, 2, 0]	8/8	12/12
PD [7:4]	8/8	12/12
PD [3:0]**	16/16	25/25
PE [7:0]	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 the first footnote.

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 RCM2100 and to build prototypes of your own circuits.

B.2 Mechanical Dimensions and Layout

Figure B-2 shows the mechanical dimensions and layout for the RCM2100 Prototyping Board.

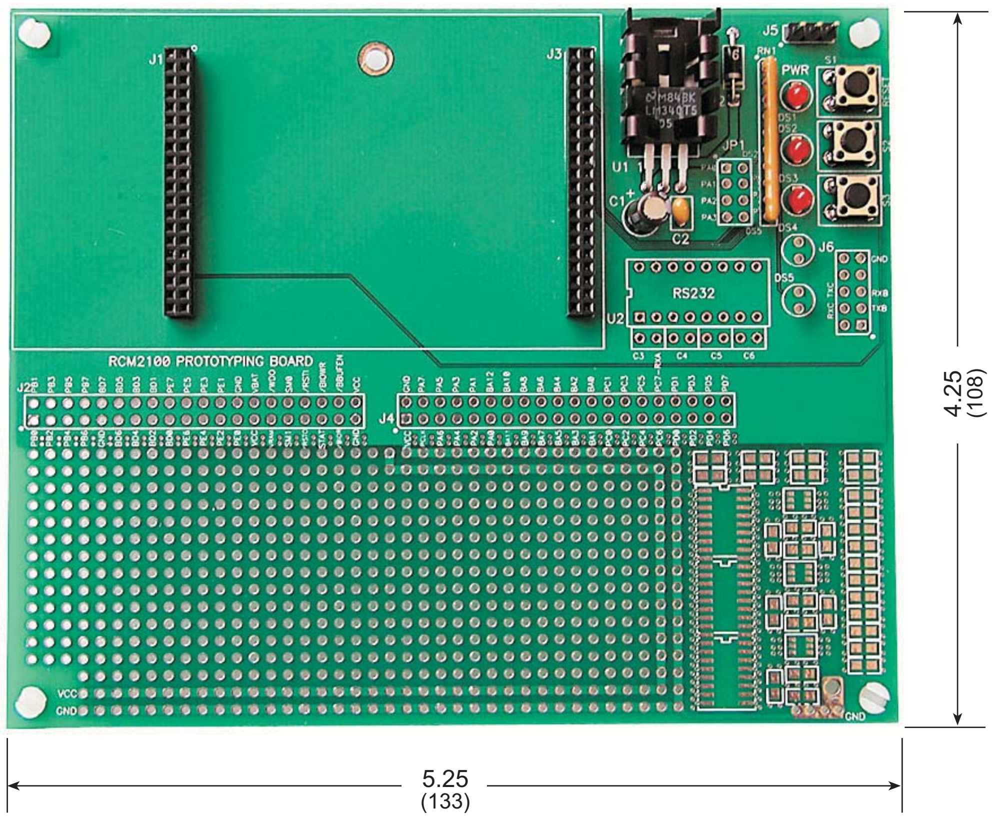


Figure B-2. RCM2100 Prototyping Board Dimensions

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

Table B-1. 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	1.7" × 4" (43 mm × 102 mm) throughhole, 0.1" spacing
Standoffs/Spacers	4, accept 6-32 × 3/8 screws

INDEX