**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 3000 |
| Co-Processor | - |
| Speed | 30MHz |
| Flash Size | 256KB |
| RAM Size | 128KB |
| Connector Type | 2 IDC Headers 2x17 |
| Size / Dimension | 1.85" x 2.73" (47mm x 69mm) |
| Operating Temperature | -40°C ~ 70°C |
| Purchase URL | https://www.e-xfl.com/product-detail/digi-international/101-0507 |

# RabbitCore RCM3000 User's Manual

Part Number 019-0110 • 070831–H • Printed in U.S.A.



## Trademarks

Rabbit and Dynamic C are registered trademarks of Rabbit Semiconductor Inc.

Rabbit 3000 and RabbitCore are trademarks of Rabbit Semiconductor Inc.

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

**Rabbit Semiconductor Inc.**

www.rabbit.com

# 1. INTRODUCTION

The RCM3000 RabbitCore module is designed to be the heart of embedded control systems. The RCM3000 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 RCM3000 refers to the complete series of RCM3000 RabbitCore modules unless other production models are referred to specifically.

The RCM3000 has a Rabbit 3000 microprocessor operating at 29.4 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 3000's internal real-time clock and the static RAM. Two 34-pin headers bring out the Rabbit 3000 I/O bus lines, parallel ports, and serial ports.

The RCM3000 receives its +3.3 V power from the customer-supplied motherboard on which it is mounted. The RabbitCore RCM3000 can interface with all kinds of CMOS-compatible digital devices through the motherboard.

## 1.1 RCM3000 Features

- Small size: 1.85" x 2.65"" x 0.86"
  (47 mm × 67 mm × 22 mm)
- Microprocessor: Rabbit 3000 running at 29.4 MHz
- 52 parallel 5 V tolerant I/O lines: 44 configurable for I/O, 4 fixed inputs, 4 fixed outputs
- Two additional digital inputs, two additional digital outputs
- External reset input
- Alternate I/O bus can be configured for 8 data lines and 6 address lines (shared with parallel I/O lines), I/O read/write
- Ten 8-bit timers (six cascadable) and one 10-bit timer with two match registers
- 256K–512K flash memory, 128K–512K SRAM
- Real-time clock
- Watchdog supervisor

- Provision for customer-supplied backup battery via connections on header J2
- 10/100-compatible RJ-45 Ethernet port with 10Base-T interface
- 10-bit free-running PWM counter and four width registers
- Two-channel Input Capture can be used to time input signals from various port pins
- Two-channel Quadrature Decoder accepts inputs from external incremental encoder modules
- Six CMOS-compatible serial ports: maximum asynchronous baud rate of 1.84 Mbps, maximum synchronous baud rate of 7.35 Mbps. Four ports are configurable as a clocked serial port (SPI), and two ports are configurable as SDLC/HDLC serial ports.
- Supports 1.15 Mbps IRDA transceiver

There are two production models in the RCM3000 series. If the standard models do not serve your needs, other variations can be specified and ordered in production quantities. Contact your Rabbit Semiconductor sales representative for details.

Table 1 below highlights the differences between the two models in the RCM3000 family.

***Table 1. RCM3000 Versions***

| Feature | RCM3000 | RCM3010 |
|---|---|---|
| Microprocessor | Rabbit 3000 running at 29.4 MHz | |
| Flash Memory | 512K | 256K |
| Static RAM | 512K | 128K |
| Serial Ports | 6 shared high-speed, CMOS-compatible ports:<br>6 are configurable as asynchronous serial ports;<br>4 are configurable as clocked serial ports (SPI);<br>2 are configurable as SDLC/HDLC serial ports;<br>1 asynchronous clocked serial port is dedicated for programming | |

**NOTE:** The RCM3010 is the RabbitCore module supplied with the Development Kit.

In addition, there is an RCM3100 series of RabbitCore modules that omits the RCM3000 series' Ethernet connectivity, but offer a much smaller footprint, which is about one-half that of the RCM3000 series.

The RabbitCore modules can be programed locally, remotely, or via a network using appropriate interface hardware.

Appendix A, "RCM3000 Specifications," provides detailed specifications for the RCM3000.

### 1.4.2 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.

# 2. HARDWARE SETUP

This chapter describes the RCM3000 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 RCM3000 Development Kit. If you purchased an RCM3000 module by itself, you will have to adapt the information in this chapter and elsewhere to your test and development setup.

## 2.1 Development Kit Contents

The RCM3000 Development Kit contains the following items:

- RCM3010 module with Ethernet port, 256K flash memory, and 128K SRAM.
- RCM30/31/32XX Prototyping Board.
- AC adapter, 12 V DC, 1 A. (Included only with Development Kits sold for the North American market. A header plug leading to bare leads is provided to allow overseas users to connect a power supply compatible with their local mains power.)
- 10-pin header to DB9 programming cable with integrated level-matching circuitry.
- ***Dynamic C*** CD-ROM, with complete product documentation on disk.
- ***Getting Started*** instructions.
- A bag of accessory parts for use on the Prototyping Board.
- ***Rabbit 3000 Processor Easy Reference*** poster.
- Registration card.

## 3.2 Sample Programs

Of the many sample programs included with Dynamic C, several are specific to the RCM3000. Sample programs illustrating the general operation of the RCM3000, and serial communication are provided in the **SAMPLES\RCM3000** folder. Each sample program has comments that describe the purpose and function of the program. Follow the instructions at the beginning of the sample program.

- **CONTROLLED.C**—uses the **STDIO** window to demonstrate digital outputs by toggling LEDs DS1 and DS2 on the Prototyping Board on and off.

  Parallel Port G bit 6 = LED DS1
  Parallel Port G bit 7 = LED DS2

  Once you have compile this program and it is running, you will be prompted via the Dynamic C **STDIO** window to select LED DS1 or DS2. Use your PC keyboard to make your selection.

  Once you have selected the LED, you will be prompted to select to turn the LED either ON or OFF. A logic low will light up the LED you selected.

- **FLASHLED1.C**—demonstrates the use of costatements to flash LEDs DS1 and DS2 on the Prototyping Board at different rates. Once you have compile this program and it is running, LEDs DS1 and DS2 will flash on/off at different rates.

- **FLASHLED2.C**—demonstrates the use of cofunctions and costatements to flash LEDs DS1 and DS2 on the Prototyping Board at different rates. Once you have compile this program and it is running, LEDs DS1 and DS2 will flash on/off at different rates.

- **TOGGLESWITCH.C**—demonstrates the use of costatements to detect switches using the press-and-release method of debouncing. LEDs DS1 and DS2 on the Prototyping Board are turned on and off when you press switches S2 and S3.

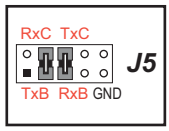
- **IR_DEMO.C**—Demonstrates sending Modbus ASCII packets between two Prototyping Board assemblies via the IrDA transceivers with the IrDA transceivers facing each other. Note that this sample program will only work with the RCM30/31/32XX Prototyping Board.

  First, compile and run this program on one Prototyping Board assembly, then remove the programming cable and press the **RESET** button on the Prototyping Board so that the first RabbitCore module is operating in the **Run** mode. Then connect the programming cable to the second Prototyping Board assembly with the RCM3000 and compile and run the same sample program. With the programming cable still connected to the second Prototyping Board assembly, press switch S2 on the second Prototyping Board to transmit a packet. Once the first Prototyping Board assembly receives a test packet, it will send back a response packet that will be displayed in the Dynamic C **STDIO** window. The test packets and response packets have different codes.

Once you have loaded and executed these sample programs and have an understanding of how Dynamic C and the RCM3000 modules interact, you can move on and try the other sample programs, or begin building your own.

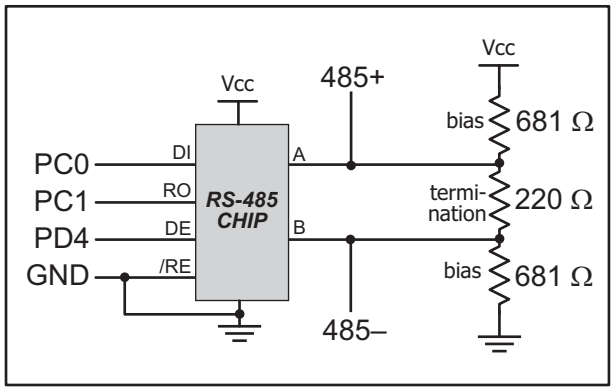
- **SWITCHCHAR.C**—This program demonstrates transmitting and then receiving an ASCII string on Serial Ports B and C. It also displays the serial data received from both ports in the **STDIO** window.

  To set up the Prototyping Board, you will need to tie TxB and RxC together on the RS-232 header at J5, and you will also tie RxB and TxC together using the jumpers supplied in the Development Kit as shown in the diagram.




  Once you have compiled and run this program, press and release S2 and S3 on the Prototyping Board. The data sent between the serial ports will be displayed in the **STDIO** window.

Two sample programs, **SIMPLE485MASTER.C** and **SIMPLE485SLAVE.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 RCM30/31/32XX Prototyping Board.




The diagram shows the connections. You will have to connect PC0 and PC1 (Serial Port D) on the RCM30/31/32XX Prototyping Board to the RS-485 transceiver, and you will connect PD4 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

- **SIMPLE485MASTER.C**—This program demonstrates a simple RS-485 transmission of lower case letters to a slave RCM3000. The slave will send back converted upper case letters back to the master RCM3000 and display them in the **STDIO** window. Use **SIMPLE485SLAVE.C** to program the slave RCM3000.

- **SIMPLE485SLAVE.C**—This program demonstrates a simple RS-485 transmission of lower case letters to a master RCM3000. The slave will send back converted upper case letters back to the master RCM3000 and display them in the **STDIO** window. Use **SIMPLE485MASTER.C** to program the master RCM3000.

### 3.2.2 Other Sample Programs

Section 6.11 describes the TCP/IP sample programs, and Appendix C.8 provides sample programs for the optional LCD/keypad module that can be installed on the Prototyping Board.

## 4.1 RCM3000 Digital Inputs and Outputs

The RCM3000 has 52 parallel I/O lines grouped in seven 8-bit ports available on headers J1 and J2. The 44 bidirectional I/O lines are located on pins PA0–PA7, PB0, PB2–PB7, PD2–PD7, PE0–PE1, PE3–PE7, PF0–PF7, and PG0–PG7.

Figure 5 shows the RCM3000 pinouts for headers J1 and J2.

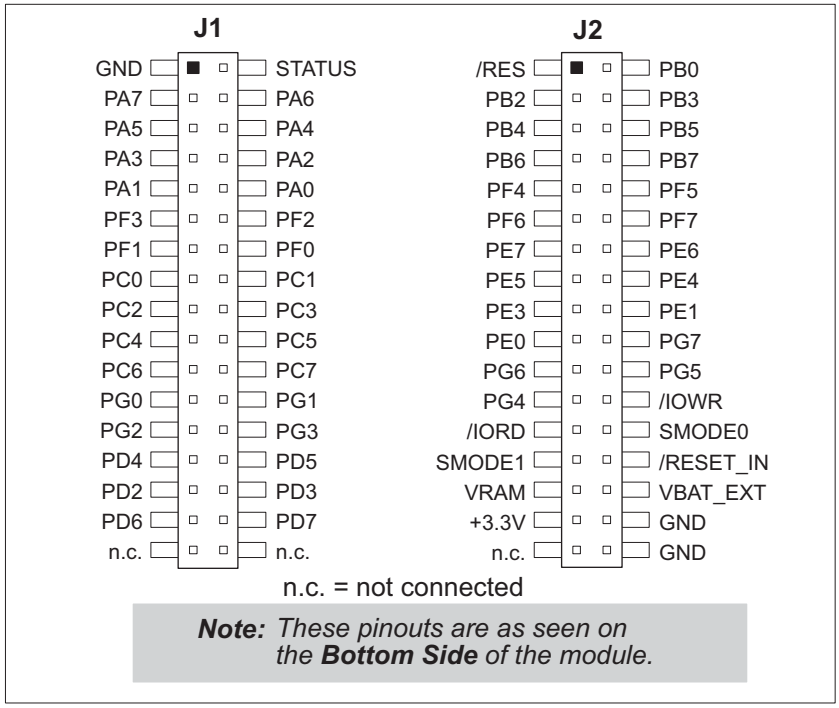



***Figure 5. RCM3000 Pinouts***

Headers J1 and J2 are standard 2 × 34 headers with a nominal 2 mm pitch. An RJ-45 Ethernet jack is also included with the RCM3000.

The signals labeled PD2, PD3, PD6, and PD7 on header J1 (pins 29–32) and the pins that are not connected (pins 33–34 on header J1 and pin 33 on header J2) are reserved for future use on other models in the RCM3000.

# 6. USING THE TCP/IP FEATURES

## 6.1 TCP/IP Connections

Programming and development can be done with the RCM3000 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 RCM3000 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 a 10Base-T Ethernet hub are available from Rabbit Semiconductor 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, "Hardware Setup."
2. Ethernet Connections

   There are four options for connecting the RCM3000 module to a network for development and runtime purposes. The first two options permit total freedom of action 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 RCM3000'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 RCM3000's Ethernet port to it, using standard network cables.

## 6.8 How to Set IP Addresses in the Sample Programs

With the introduction of Dynamic C 7.30 we have taken steps to make it easier to run many of our sample programs. Instead of the **MY_IP_ADDRESS** and other macros, 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 RCM3000 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** file.

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. 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 such as shown in the example below use macros to define the IP address assigned to the board and the IP address of the gateway, if there is a gateway.

```
#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 RCM3000:

```
#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**.

- **ENET_MENU.C**—This program demonstrates how to implement a menu system using a highlight bar on a graphic LCD display and to communicate it to another single-board computer via Ethernet.

  Use **ENET_AD.C** to program the other single-board computer with analog inputs and outputs.

- **MBOXDEMO.C**—Implements a Web server that allows e-mail messages to be entered and then shown on the LCD/keypad module.

- **SMTP.C**—This program allows you to send an E-mail when a switch on the Prototyping Board is pressed. Follow the instructions included with the sample program.

- **PINGLED.C**—This program demonstrates ICMP by pinging a remote host. It will flash LEDs DS1 and DS2 on the Prototyping Board when a ping is sent and received.

## 6.12 Where Do I Go From Here?

> **NOTE:** If you purchased your RCM3000 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.

It is recommended that you allow for an “exclusion zone” of 0.04" (1 mm) around the RCM3000 in all directions when the RCM3000 is incorporated into an assembly that includes other printed circuit boards. This “exclusion zone” that you keep free of other components and boards will allow for sufficient air flow, and will help to minimize any electrical or electromagnetic interference between adjacent boards. An “exclusion zone” of 0.08" (2  mm) is recommended below the RCM3000 when the RCM3000 is plugged into another assembly using the shortest connectors for headers J1 and J2. Figure A-2 shows this “exclusion zone.”

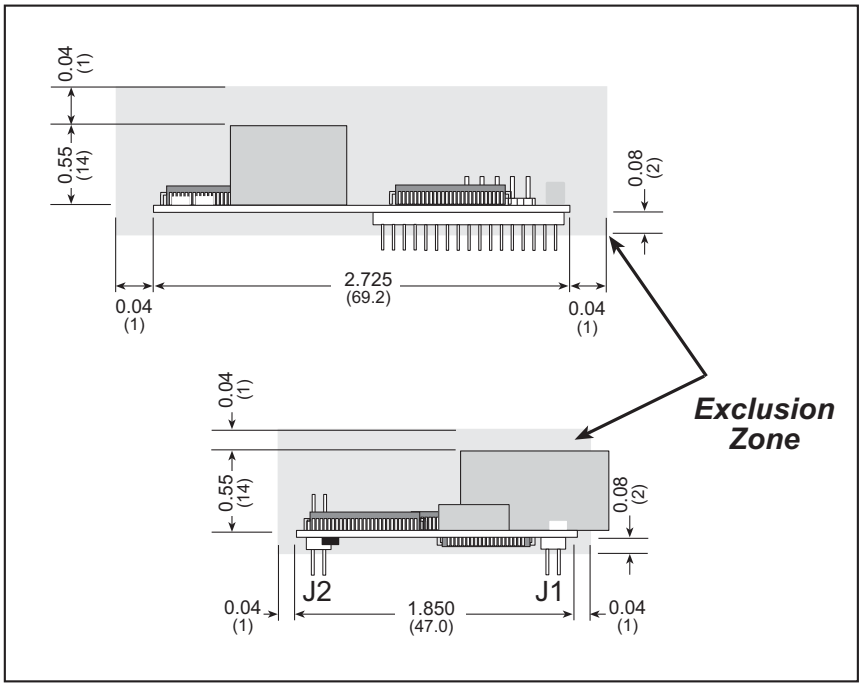



***Figure A-2.  RCM3000 “Exclusion Zone”***

*Table A-1. RabbitCore RCM3000 Specifications (continued)*

| Feature | RCM3000 | RCM3010 |
|---|---|---|
| Operating Temperature | –40°C to +70°C | |
| Humidity | 5% to 95%, noncondensing | |
| Connectors (for connection to headers J4 and J5) | Two 2 × 17, 2 mm pitch | |
| Board Size | 1.850" × 2.725" × 0.86"<br>(47 mm × 69 mm × 22 mm) | |

### A.1.1 Headers

The RCM3000 uses headers at J1 and J2 for physical connection to other boards. J1 and J2 are 2 × 17 SMT headers with a 2 mm pin spacing. J3, the programming port, is a 2 × 5 header with a 2 mm pin spacing.

Figure A-3 shows the layout of another board for the RCM3000 to be plugged into. These values are relative to the mounting hole.

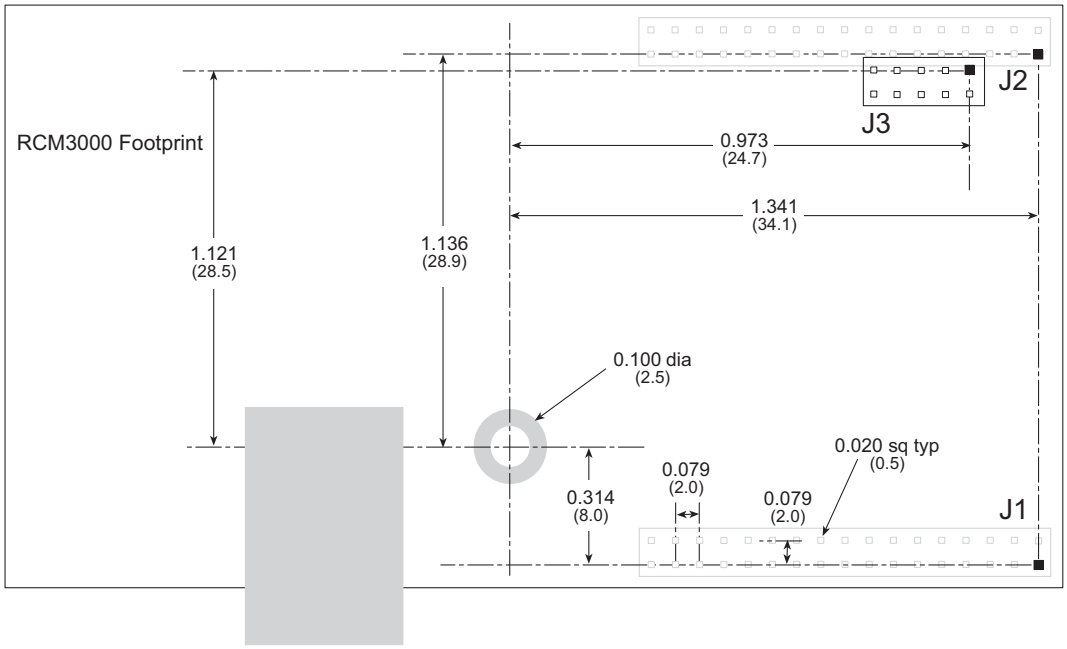



*Figure A-3. User Board Footprint for RCM3000*

### A.1.2 Physical Mounting

A 9/32” (7 mm) standoff with a 2-56 screw is recommended to attach the RCM3000 to a user board at the hole position shown in Figure A-3. Either use plastic hardware, or use insulating washers to keep any metal hardware from shorting out signals on the RCM3000.

- **RS-232**—Two 3-wire or one 5-wire RS-232 serial port are available on the Prototyping Board. Refer to the Prototyping Board schematic (090-0137) for additional details.

  A 10-pin 0.1-inch spacing header strip is installed at J5 to permit connection of a ribbon cable leading to a standard DE-9 serial connector.

- **Current Measurement Option**—Jumpers across pins 1–2 and 5–6 on header JP1 can be removed and replaced with an ammeter across the pins to measure the current drawn from the +5 V or the +3.3 V supplies, respectively.

- **Motor Encoder**—A motor/encoder header is provided at header J6 for future use.

- **LCD/Keypad Module**—our's LCD/keypad module may be plugged in directly to headers J7, J8, and J10.

## B.2 Mechanical Dimensions and Layout

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

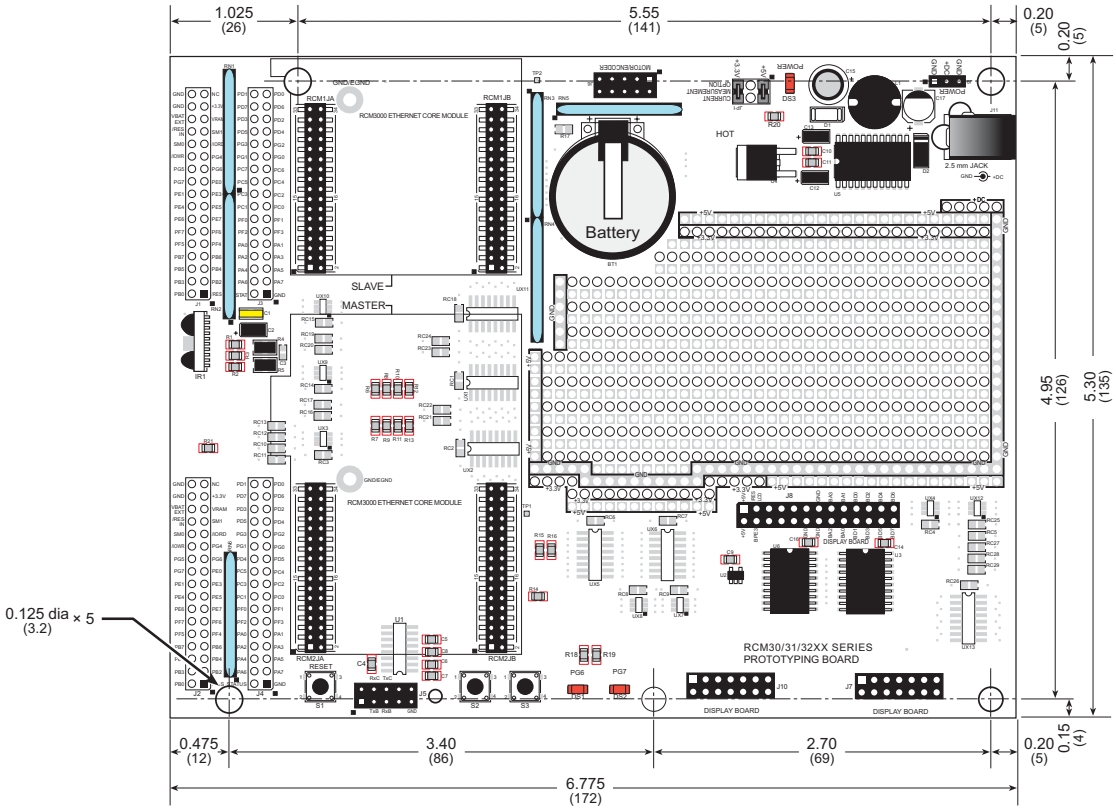



***Figure B-2. RCM30/31/32XX Prototyping Board Dimensions***

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

***Table B-1. RCM30/31/32XX Prototyping Board Specifications***

| Parameter | Specification |
|---|---|
| Board Size | 5.30" × 6.775" × 1.00" (135 mm × 172 mm × 25 mm) |
| Operating Temperature | –20°C to +60°C |
| Humidity | 5% to 95%, noncondensing |
| Input Voltage | 8 V to 24 V DC |
| Maximum Current Draw (including user-added circuits) | 800 mA max. for +3.3 V supply,<br>1 A total +3.3 V and +5 V combined |
| Prototyping Area | 2.0" × 3.5" (50 mm × 90 mm) throughhole, 0.1" spacing, additional space for SMT components |
| Standoffs/Spacers | 5, accept 4-40 × 3/8 screws |

## B.3 Power Supply

The RCM3000 requires a regulated 3.3 V ± 0.15 V DC power source to operate. Depending on the amount of current required by the application, different regulators can be used to supply this voltage.

The Prototyping Board has an onboard +5 V switching power regulator from which a +3.3 V linear regulator draws its supply. Thus both +5 V and +3.3 V are available on the Prototyping Board.

The Prototyping Board itself is protected against reverse polarity by a Shottky diode at D2 as shown in Figure B-3.

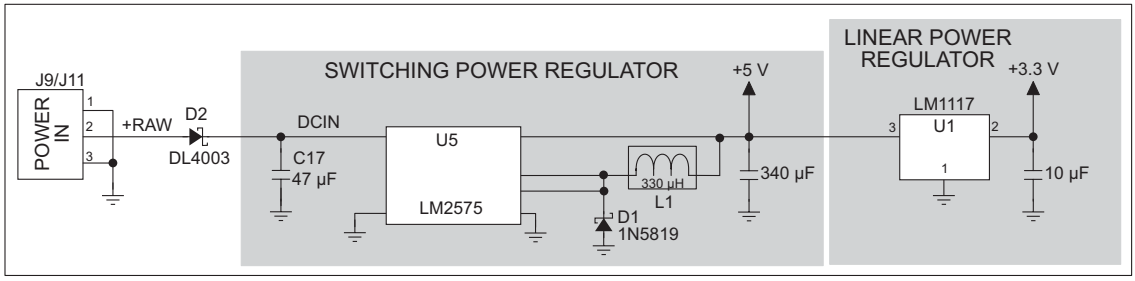



***Figure B-3. Prototyping Board Power Supply***

# APPENDIX C. LCD/KEYPAD MODULE

An optional LCD/keypad is available for the Prototyping Board. Appendix C describes the LCD/keypad and provides the software function calls to make full use of the LCD/keypad.

## C.1 Specifications

Two optional LCD/keypad modules—with or without a panel-mounted bezel—are available for use with the Prototyping Board. They are shown in Figure C-1.

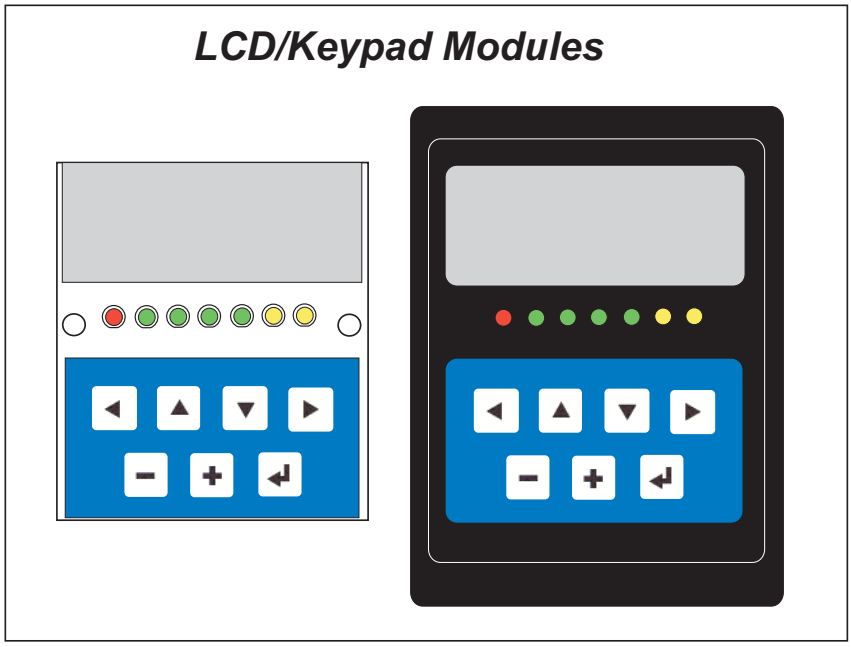



***Figure C-1. LCD/Keypad Modules Models***

Only the version without the bezel can mount directly on the Prototyping Board; if you have the version with a bezel, you will have to remove the bezel to be able to mount the LCD/keypad module on the Prototyping Board. Either version of the LCD/keypad module can be installed at a remote location up to 60 cm (24") away. Contact your Rabbit Semiconductor sales representative or your authorized Rabbit Semiconductor distributor for further assistance in purchasing an LCD/keypad module.

### *Table E-2. Parallel Port F Registers*

| Register Name | Mnemonic | I/O Address | R/W | Reset Value |
|---|---|---|---|---|
| Port F Data Register | PFDR | 00111000 (0x38) | R/W | xxxxxxxx |
| **Bits** | **Value** | **Description** | | |
| 0:7 | Read | Current state of pins | | |
| | Write | Port buffer. Value transferred to O/P register on next rising edge of transfer clock. | | |
| Port F Control Register | PFCR | 00111100 (0x3C) | W only | xx00xx00 |
| **Bits** | **Value** | **Description** | | |
| 0:1 | 00 | Lower nibble transfer clock is CLK/2 | | |
| | 01 | Lower nibble transfer clock is Timer A1 | | |
| | 10 | Lower nibble transfer clock is Timer B1 | | |
| | 11 | Lower nibble transfer clock is Timer B2 | | |
| 2:3 | xx | These bits are ignored | | |
| 4:5 | 00 | Upper nibble transfer clock is CLK/2 | | |
| | 01 | Upper nibble transfer clock is Timer A1 | | |
| | 10 | Upper nibble transfer clock is Timer B1 | | |
| | 11 | Upper nibble transfer clock is Timer B2 | | |
| 6:7 | xx | These bits are ignored | | |
| Port F Function Register | PFFR | 00111101 (0x3D) | W | xxxxxxxx |
| **Bits** | **Value** | **Description** | | |
| 0:7 | 0 | Corresponding port bits function normally | | |
| 0 | 1 | Bit 0 carries SCLK_D | | |
| 1 | 1 | Bit 1 carries SCLK_C | | |
| 2:3 | x | No effect | | |
| 4 | 1 | Bit 4 carries PWM[0] output | | |
| 5 | 1 | Bit 5 carries PWM[1] output | | |
| 6 | 1 | Bit 6 carries PWM[2] output | | |
| 7 | 1 | Bit 7 carries PWM[3] output | | |
| Port F Drive Control Register | PFDCR | 00111110 (0x3E) | W | xxxxxxxx |
| **Bits** | **Value** | **Description** | | |
| 0:7 | 0 | Corresponding port bit is active high or low | | |
| | 1 | Corresponding port bit is open drain | | |

*Table E-2. Parallel Port F Registers (continued)*

| Register Name | Mnemonic | I/O Address | R/W | Reset Value |
|---|---|---|---|---|
| Port F Data Direction Register | PFDDR | 00111111 (0x3F) | W | 00000000 |
| **Bits** | **Value** | **Description** | | |
| 0:7 | 0 | Corresponding port bit is an input | | |
| | 1 | Corresponding port bit is an output | | |