PRODUCT MANUAL

Wildcat (BL2000)

C-Programmable Single-Board Computer with Ethernet

User's Manual

019–0094 • 090529–O

Wildcat (BL2000) User's Manual

Part Number 019-0094 • 090529–O • Printed in U.S.A.

©2001–2009 Digi International Inc. • All rights reserved.

No part of the contents of this manual may be reproduced or transmitted in any form or by any means without the express written permission of Digi International.

Permission is granted to make one or more copies as long as the copyright page contained therein is included. These copies of the manuals may not be let or sold for any reason without the express written permission of Digi International.

Digi International reserves the right to make changes and improvements to its products without providing notice.

Trademarks

Rabbit and Dynamic C are registered trademarks of Digi International Inc.

Rabbit 2000 is a trademark of Digi International Inc.

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

Digi International Inc.

www.rabbit.com

TABLE OF CONTENTS

Chapter 1. Introduction **1**
1.1 BL2000 Description 1
1.2 BL2000 Features 1
1.2.1 Connector Options 2
1.3 Development and Evaluation Tools 3
1.3.1 Tool Kit 3
1.3.2 Software 4
1.4 CE Compliance 5
1.4.1 Design Guidelines 6
1.4.2 Interfacing the BL2000 to Other Devices 6

Chapter 2. Getting Started **7**
2.1 BL2000 Connections 7
2.2 Installing Dynamic C 10
2.3 Starting Dynamic C 11
2.4 PONG.C 12
2.5 Where Do I Go From Here? 12

Chapter 3. Subsystems **13**
3.1 BL2000 Pinouts 14
3.1.1 Headers and Screw Terminals 15
3.1.2 Power Supply Pins 16
3.2 Digital I/O 17
3.2.1 Digital Inputs 17
3.2.2 Digital Outputs 19
3.3 Relay Outputs 21
3.4 Serial Communication 22
3.4.1 RS-232 22
3.4.2 RS-485 22
3.4.3 Ethernet Port 24
3.4.4 Programming Port 25
3.5 A/D Converter Inputs 27
3.6 D/A Converter Outputs 28
3.7 Memory 30
3.7.1 SRAM 30
3.7.2 Flash Memory 30
3.8 Programming Cable 31
3.8.1 Changing Between Program Mode and Run Mode 31
3.9 Other Hardware 32
3.9.1 External Interrupts 32
3.9.2 Clock Doubler 32
3.9.3 Spectrum Spreader 33

Chapter 4. Software **35**
4.1 An Overview of Dynamic C 35
4.1.1 Upgrading Dynamic C 37

4.2 Sample Programs ... 38
4.2.1 General BL2000 Sample Programs ... 38
4.2.2 Digital I/O ... 38
4.2.3 Serial Communication ... 39
4.2.4 A/D Converter Inputs ... 39
4.2.5 D/A Converter Outputs ... 40
4.2.6 Real-Time Clock ... 40
4.2.7 TCP/IP Sample Programs ... 40
4.3 BL2000 Libraries ... 41
4.4 BL2000 Function Calls ... 42
4.4.1 Board Initialization ... 42
4.4.2 Digital I/O ... 44
4.4.3 Serial Communication ... 45
4.4.4 Relay and LED Outputs ... 46
4.4.5 A/D Converter Inputs ... 47
4.4.6 D/A Converter Outputs ... 50

Chapter 5. Using the TCP/IP Features 53
5.1 TCP/IP Connections ... 53
5.2 TCP/IP Sample Programs ... 55
5.2.1 How to Set IP Addresses in the Sample Programs ... 55
5.2.2 How to Set Up your Computer's IP Address for a Direct Connection ... 56
5.3 Run the PINGME.C Sample Program ... 57
5.4 Running More Sample Programs With a Direct Connection ... 58
5.5 Where Do I Go From Here? ... 58

Appendix A. Specifications 59
A.1 Electrical and Mechanical Specifications ... 60
A.1.1 Headers ... 63
A.2 Conformal Coating ... 64
A.3 Jumper Configurations ... 65
A.4 Use of Rabbit 2000 Parallel Ports ... 67

Appendix B. Plastic Enclosure 69
B.1 Assembly ... 70
B.2 Dimensions ... 72

Appendix C. Power Supply 73
C.1 Power Supplies ... 73
C.1.1 Power for Analog Circuits ... 74
C.2 Batteries and External Battery Connections ... 74
C.2.1 Replacing the Backup Battery ... 75
C.2.2 Battery-Backup Circuit ... 75
C.2.3 Power to VRAM Switch ... 76
C.2.4 Reset Generator ... 77
C.3 Chip Select Circuit ... 78

Appendix D. Demonstration Board 81
D.1 Connecting Demonstration Board ... 81

Index 85

Schematics 89

1. INTRODUCTION

The BL2000 is a high-performance, C-programmable single-board computer that offers built-in digital and analog I/O combined with Ethernet connectivity in a compact form factor. A Rabbit® 2000 microprocessor operating at 22.1 MHz provides fast data processing. An optional plastic enclosure is available, and may be wall-mounted or panel-mounted.

1.1 BL2000 Description

The BL2000 is an advanced single-board computer that incorporates the powerful Rabbit 2000 microprocessor, flash memory, static RAM, digital I/O ports, A/D converter inputs, D/A converter outputs, an SPDT relay output, and a 10Base-T Ethernet port.

1.2 BL2000 Features

- Rabbit® 2000 microprocessor operating at 22.1 MHz.
- 128K static RAM and 256K flash memory.
- Up to 28 digital I/O:
  11 protected digital inputs (plus up to 7 dual-purpose unbuffered analog inputs that may be software-configured for use as digital inputs) and 10 high-current digital sinking outputs that may be factory-configured as sourcing outputs.
- 11 analog channels: nine 12-bit A/D converter inputs, two 12-bit D/A converter outputs.
- Onboard SPDT relay.
- One RJ-45 Ethernet port compliant with IEEE 802.3 standard for 10Base-T Ethernet protocol.
- Eight status LEDs.
- 4 serial ports (2 RS-232 or 1 RS-232 with RTS/CTS, 1 RS-485, and 1 CMOS-compatible programming port).
- Real-time clock.
- Watchdog supervisor.
- Voltage regulator.

- Backup battery.
- Ability to send e-mail and serve Web pages containing embedded data from single-board computer.
- Remote program downloading and debugging capability via RabbitLink.
- Boards with the CE mark are CE-compliant.
- Optional plastic enclosure (can be wall-mounted or panel-mounted) and LED light pipes (enclosure and light pipes are included with the Tool Kit, and are also sold separately).

Appendix A provides detailed specifications.

Four models of the BL2000 are available. Their standard features are summarized in Table 1.

Table 1. BL2000 Series Features

Model	Features
BL2000	Full-featured single-board computer.
BL2010	BL2000 with eleven 10-bit A/D converter inputs (no D/A converter outputs).
BL2020	BL2000 *without* Ethernet interface, only 6 LEDs.
BL2030	BL2010 *without* Ethernet interface, only 6 LEDs.

1.2.1 Connector Options

In addition to the standard screw-terminal connectors supplied on BL2000 boards, IDC headers, bottom-mount connectors, and polarized friction-lock terminals may be factory-installed instead. Visit our Web site at www.rabbit.com or contact your Rabbit sales representative or authorized distributor for further information.

Standard screw terminals, accept up to 14 AWG (1.5 mm^2) wire		“Bottom-mount connector” to mount BL2000 directly on 0.1" pitch pins located on motherboard	
2 × 17 IDC headers, 0.1" pitch		Polarized friction-lock terminals, 0.1" pitch	

1.3 Development and Evaluation Tools

1.3.1 Tool Kit

A Tool Kit contains the hardware essentials you will need to use your own BL2000 single-board computer. The items in the Tool Kit and their use are as follows:

- ***Getting Started*** instructions.
- ***Dynamic C*** CD-ROM, with complete product documentation on disk.
- Programming cable, used to connect your PC serial port to the BL2000.
- Universal AC adapter, 12 V DC, 1 A (includes Canada/Japan/U.S., Australia/N.Z., U.K., and European style plugs).
- Demonstration Board with pushbutton switches and LEDs. The Demonstration Board can be hooked up to the BL2000 to demonstrate the I/O.
- Wire assembly to connect Demonstration Board to BL2000.
- Plastic enclosure with four screws and eight customer-installable light pipes.
- Screwdriver.
- ***Rabbit 2000 Processor Easy Reference*** poster.
- Registration card.

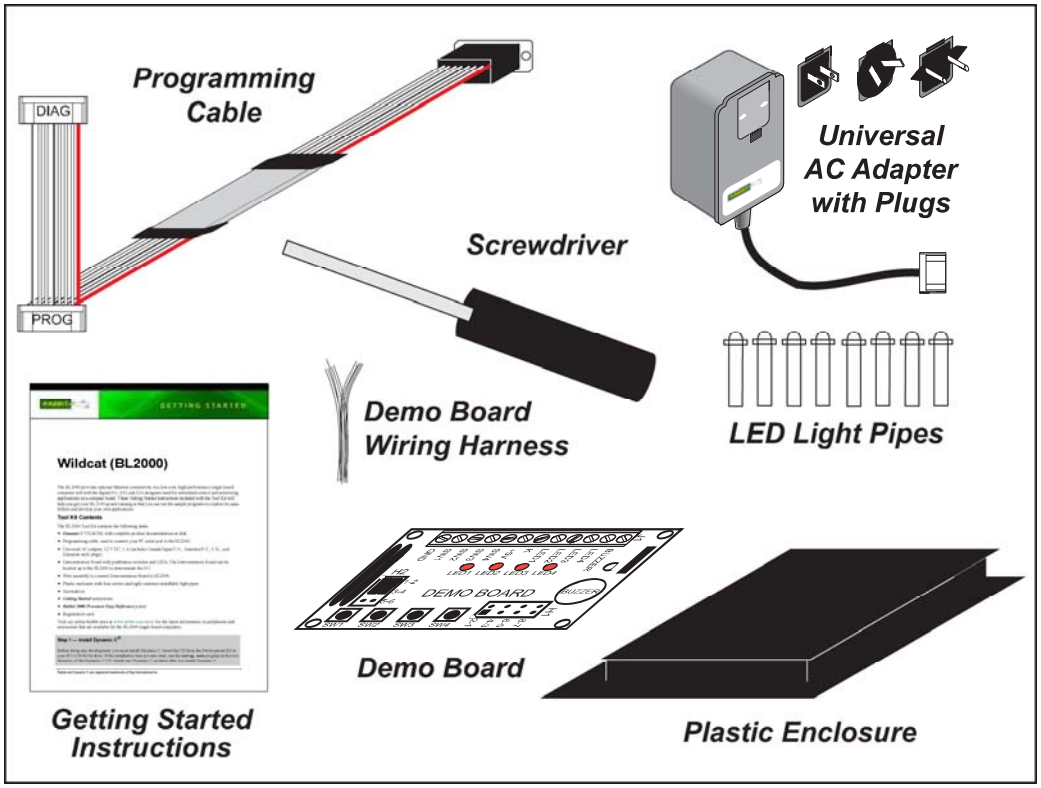


Figure 1. BL2000 Tool Kit

1.3.2 Software

The BL2000 is programmed using version 7.04 or later of Rabbit's Dynamic C. A compatible version is included on the Tool Kit CD-ROM. Dynamic C v. 9.60 includes the popular µC/OS-II real-time operating system, point-to-point protocol (PPP), FAT file system, RabbitWeb, and other select libraries that were previously sold as individual Dynamic C modules.

Rabbit also offers for purchase the Rabbit Embedded Security Pack featuring the Secure Sockets Layer (SSL) and a specific Advanced Encryption Standard (AES) library. In addition to the Web-based technical support included at no extra charge, a one-year telephone-based technical support subscription is also available for purchase. Visit our Web site at www.rabbit.com for further information and complete documentation, or contact your Rabbit sales representative or authorized distributor.

1.4 CE Compliance

Equipment is generally divided into two classes.

CLASS A	CLASS B
Digital equipment meant for light industrial use	Digital equipment meant for home use
Less restrictive emissions requirement: less than 40 dB µV/m at 10 m (40 dB relative to 1 µV/m) or 300 µV/m	More restrictive emissions requirement: 30 dB µV/m at 10 m or 100 µV/m

These limits apply over the range of 30–230 MHz. The limits are 7 dB higher for frequencies above 230 MHz. Although the test range goes to 1 GHz, the emissions from Rabbit-based systems at frequencies above 300 MHz are generally well below background noise levels.

The BL2000 single-board computer has been tested and was found to be in conformity with the following applicable immunity and emission standards. The BL2010, BL2020, and BL2030 single-board computers are also CE qualified as they are sub-versions of the BL2000 single-board computer. Boards that are CE-compliant have the CE mark.

NOTE: Earlier versions of the BL2000 sold before 2003 that do not have the CE mark are ***not*** CE-compliant.

Immunity

The BL2000 series of single-board computers meets the following EN55024/1998 immunity standards.

- EN61000-4-3 (Radiated Immunity)
- EN61000-4-4 (EFT)
- EN61000-4-6 (Conducted Immunity)

Additional shielding or filtering may be required for a heavy industrial environment.

Emissions

The BL2000 series of single-board computers meets the following emission standards using the enhanced-EMI PCB, Part# 175-0224 Rev. A, and the 668-0003 Rev. A Rabbit 2000 microprocessor.

- EN55022:1998 Class B
- FCC Part 15 Class B

Your results may vary, depending on your application, so additional shielding or filtering may be needed to maintain the Class B emission qualification.

1.4.1 Design Guidelines

Note the following requirements for incorporating the BL2000 series of single-board computers into your application to comply with CE requirements.

General

- The power supply provided with the Tool Kit is for development purposes only. It is the customer's responsibility to provide a CE-compliant power supply for the end-product application.
- When connecting the BL2000 single-board computer to outdoor cables, the customer is responsible for providing CE-approved surge/lighting protection.
- Rabbit recommends placing digital I/O or analog cables that are 3 m or longer in a metal conduit to assist in maintaining CE compliance and to conform to good cable design practices.
- When installing or servicing the BL2000, it is the responsibility of the end-user to use proper ESD precautions to prevent ESD damage to the BL2000.

Safety

- All inputs and outputs to and from the BL2000 series of single-board computers must not be connected to voltages exceeding SELV levels (42.4 V AC peak, or 60 V DC).
- The lithium backup battery circuit on the BL2000 single-board computer has been designed to protect the battery from hazardous conditions such as reverse charging and excessive current flows. Do not disable the safety features of the design.

1.4.2 Interfacing the BL2000 to Other Devices

Since the BL2000 series of single-board computers is designed to be connected to other devices, good EMC practices should be followed to ensure compliance. CE compliance is ultimately the responsibility of the integrator. Additional information, tips, and technical assistance are available from your authorized Rabbit distributor, and are also available on our Web site at www.rabbit.com.

2. GETTING STARTED

Chapter 2 explains how to connect the programming cable and power supply to the BL2000.

2.1 BL2000 Connections

1. Attach the BL2000 to the plastic enclosure base.

Position the BL2000 over the plastic enclosure base as shown below in Figure 2. Attach the BL2000 to the base at the top left and bottom right positions using the two 4-40 × ¼ screws supplied with the enclosure.

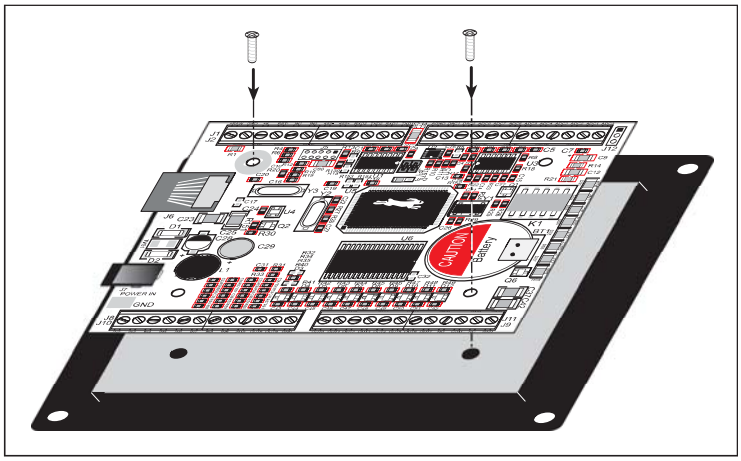


Figure 2. Attach BL2000 to Plastic Enclosure Base

The plastic enclosure base facilitates handling the BL2000 during development, and provides an attractive mounting alternative. Alternatively, you may wish to use standoffs to protect the components on the other side of the board. The plastic enclosure is offered as a separate option when individual BL2000 boards are purchased.

NOTE: Appendix B, "Plastic Enclosure," provides additional information and specifications for the plastic enclosure.

2. Connect the programming cable to download programs from your PC and to debug the BL2000.

Connect the 10-pin **PROG** connector of the programming cable to header J5 on the BL2000. Ensure that the colored edge lines up with pin 1 as shown. (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.

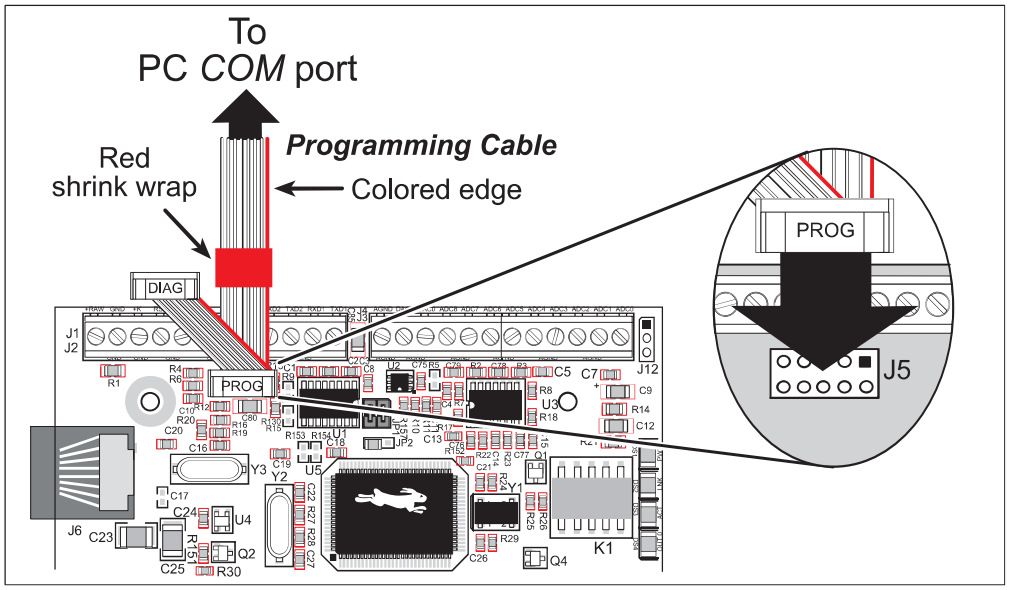


Figure 3. Programming Cable Connections

NOTE: Never disconnect the programming cable by pulling on the ribbon cable. Carefully pull on the connector to remove it from the header.

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 Tool Kit. Note that not all RS-232/USB converters work with Dynamic C.

3. Connect the power supply.

First, prepare the AC adapter for the country where it will be used by selecting the plug. The BL2000 Tool Kit presently includes Canada/Japan/U.S., Australia/N.Z., U.K., and European style plugs. Snap in the top of the plug assembly into the slot at the top of the AC adapter as shown in Figure 4, then press down on the spring-loaded clip below the plug assembly to allow the plug assembly to click into place.

Connect the bare ends of the power supply to the **+RAW** and **GND** positions on screw-terminal header J2 as shown in Figure 4.

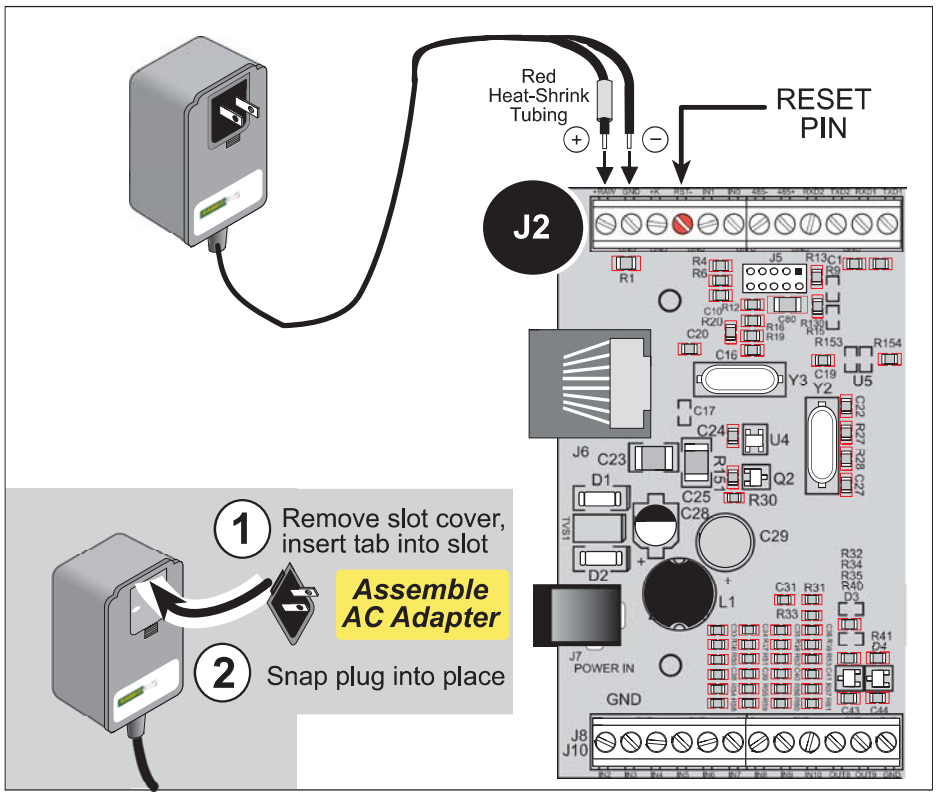


Figure 4. Power Supply Connection

4. Apply power.

Plug in the AC adapter. If you are using your own power supply, it must provide 9 V to 40 V DC—voltages outside this range could damage the BL2000.

CAUTION: Unplug the power supply while you make or otherwise work with the connections to the headers. This will protect your BL2000 from inadvertent shorts or power spikes.

NOTE: The green **PWR** LED and the red **BAD** LED on the opposite end of the board should come on, indicating that the BL2000 is now ready to be used.

NOTE: A hardware RESET is done by unplugging the AC adapter, then plugging it back in, or by momentarily grounding the board reset input at pin 9 on screw-terminal header J2.

2.2 Installing Dynamic C

If you have not yet installed Dynamic C version 7.04 (or a later version), do so now by inserting the Dynamic C CD in your PC's CD-ROM drive. The CD will auto-install unless you have disabled auto-install on your PC.

If the CD does not auto-install, click **Start > Run** from the Windows **Start** button and browse for the Dynamic C `setup.exe` file on your CD drive. Click **OK** to begin the installation once you have selected the `setup.exe` file.

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, create a new desktop icon that points to **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.

The ***Dynamic C User's Manual*** provides detailed instructions for the installation of Dynamic C and any future upgrades.

> **NOTE:** If you have an earlier version of Dynamic C already installed, the default installation of the later version will be in a different folder, and a separate icon will appear on your desktop.

2.3 Starting Dynamic C

Once the BL2000 is connected to your PC and to a power source, start Dynamic C by double-clicking on the Dynamic C icon on your desktop or in your **Start** menu.

Dynamic C defaults to using the serial port on your PC that you specified during installation. If the port setting is correct, Dynamic C should detect the BL2000 and go through a sequence of steps to cold-boot the BL2000 and to compile the BIOS. (Some versions of Dynamic C will not do the initial BIOS compile and load until the first time you compile a program.)

If you are using a USB port to connect your computer to the BL2000, choose **Options > Project Options** and select "Use USB to Serial Converter." Click **OK**.

If you receive the message **`No Rabbit Processor Detected`**, the programming cable may be connected to the wrong COM port, a connection may be faulty, or the target system may not be powered up. First, check both ends of the programming cable to ensure that it is firmly plugged into the PC and the programming port.

If there are no faults with the hardware, select a different COM port within Dynamic C. From the **Options** menu, select **Communications**. Select another COM port from the list, then click OK. Press **<Ctrl-Y>** to force Dynamic C to recompile the BIOS. If Dynamic C still reports it is unable to locate the target system, repeat the above steps until you locate the active COM port. You should receive a **`Bios compiled successfully`** message once this step is completed successfully.

If Dynamic C appears to compile the BIOS successfully, but you then receive a communication error message when you compile and load a sample program, it is possible that your PC cannot handle the higher program-loading baud rate. Try changing the maximum download rate to a slower baud rate as follows.

- Locate the **Serial Options** dialog in the Dynamic C **Options > Communications** menu. Select a slower Max download baud rate.

If a program compiles and loads, but then loses target communication before you can begin debugging, it is possible that your PC cannot handle the default debugging baud rate. Try lowering the debugging baud rate as follows.

- Locate the **Serial Options** dialog in the Dynamic C **Options > Communications** menu. Choose a lower debug baud rate.

2.4 PONG.C

You are now ready to test your set-up by running a sample program.

Find the file **PONG.C**, which is in the Dynamic C **SAMPLES** folder. To run the program, open it with the **File** menu (if it is not still open), compile it using the **Compile** menu, and then run it by selecting **Run** in the **Run** menu. The **STDIO** window will open and will display a small square bouncing around in a box.

This program does not test the serial ports, the I/O, or the TCP/IP part of the board, but does ensure that the board is basically functional. The sample program described in Section 5.3, "Run the PINGME.C Sample Program," tests the TCP/IP portion of the board.

2.5 Where Do I Go From Here?

NOTE: If you purchased your BL2000 through a distributor or Rabbit 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 Technical Bulletin Board and forums at www.rabbit.com/support/bb/ and at www.rabbit.com/forums/.
- Use the Technical Support e-mail form at www.rabbit.com/support/.

If the sample program ran fine, you are now ready to go on to explore other BL2000 features and develop your own applications.

Chapter 3, "Subsystems," provides a description of the BL2000's features, Chapter 4, "Software," describes the Dynamic C software libraries and introduces some sample programs, and Chapter 5, "Using the TCP/IP Features," explains the TCP/IP features.

3. SUBSYSTEMS

Chapter 3 describes the principal subsystems for the BL2000.

- Digital I/O
- Relay Outputs
- Serial Communication
- A/D Converter Inputs
- D/A Converter Outputs
- Memory
- External Interrupts

Figure 5 shows these Rabbit-based subsystems designed into the BL2000.

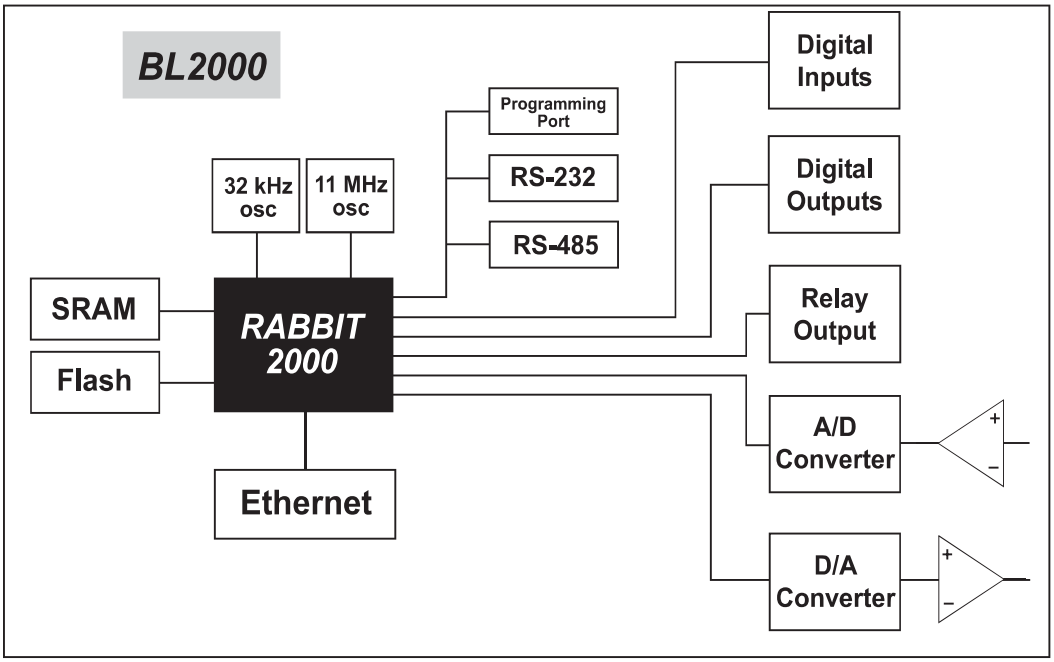


Figure 5. BL2000 Subsystems

3.1 BL2000 Pinouts

The BL2000 pinouts are shown in Figure 6(a) and Figure 6(b).

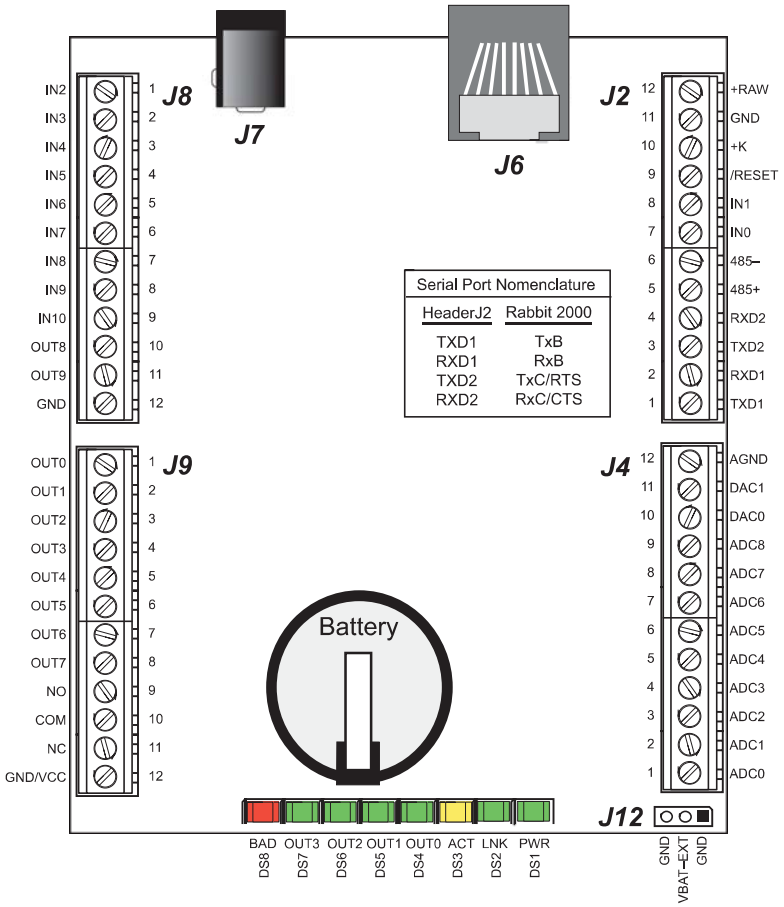


Figure 6(a). BL2000 Pinouts (screw-terminal headers)

3.1.1 Headers and Screw Terminals

All BL2000 models are equipped with 1 × 12 screw terminal strips (J2, J4, J8, and J9) and a 2-pin power jack (J7). The BL2000 and BL2010 also have the RJ-45 Ethernet jack (J6).

There is provision on the circuit board to accommodate one of the following types of connectors instead of the screw-terminal strips.

- 2 × 17 IDC headers with a pitch of 0.1".
- 1 × 17 friction-lock connectors with a pitch of 0.1".
- 1 × 17 bottom-mount sockets with a pitch of 0.1". The holes for the bottom-mount sockets are on the “outside” edges of the connector locations

The pinouts for these connectors are shown in Figure 6(b).

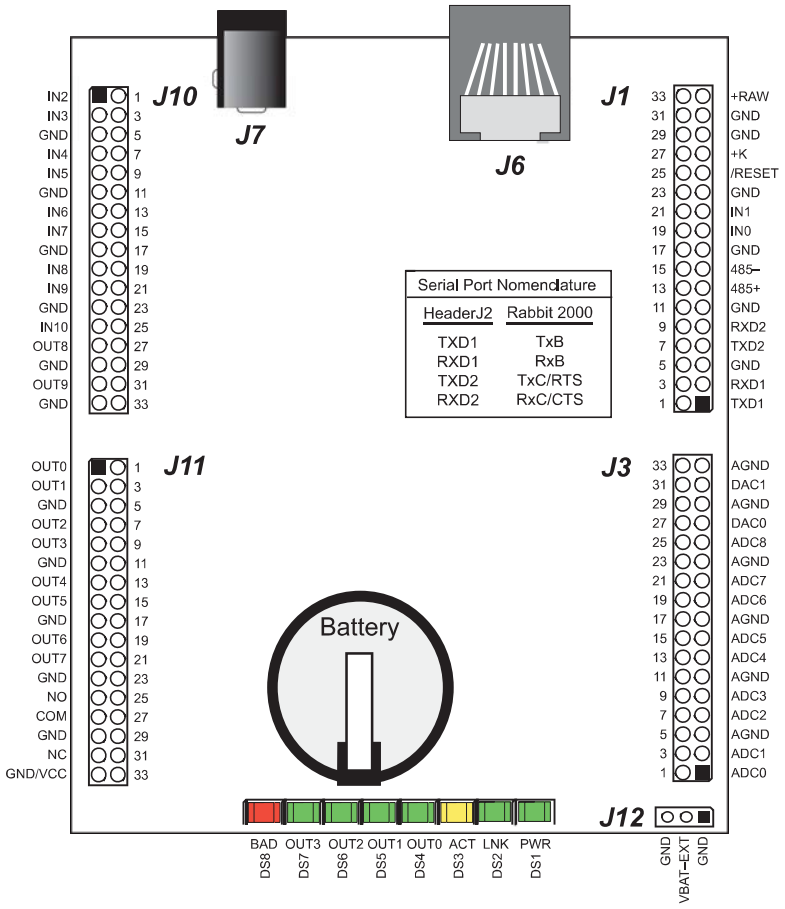


Figure 6(b). BL2000 Pinouts (other 0.1" headers)

3.1.2 Power Supply Pins

Instead of connecting an AC adapter to the power supply jack, J7, the input power supply (9 V to 40 V DC) may be connected to pins 12 and 11 on header J2 (see Figure 6(a) or Figure 6(b)).

Pin 12 on header J9 or J10 is normally GND by factory default, but may be changed to Vcc by removing resistor R161 and installing resistor R160. See Appendix C, “Power Supply,” for more information on this configuration and for information on backup-battery options.

3.2 Digital I/O

3.2.1 Digital Inputs

The BL2000 has 11 digital inputs, IN0–IN10, each of which is protected over a range of –36 V to +36 V. The inputs are factory-configured to be pulled up to +5 V, but they can also be pulled down by moving the surface-mounted jumper at JP6 as shown in Figure 7.

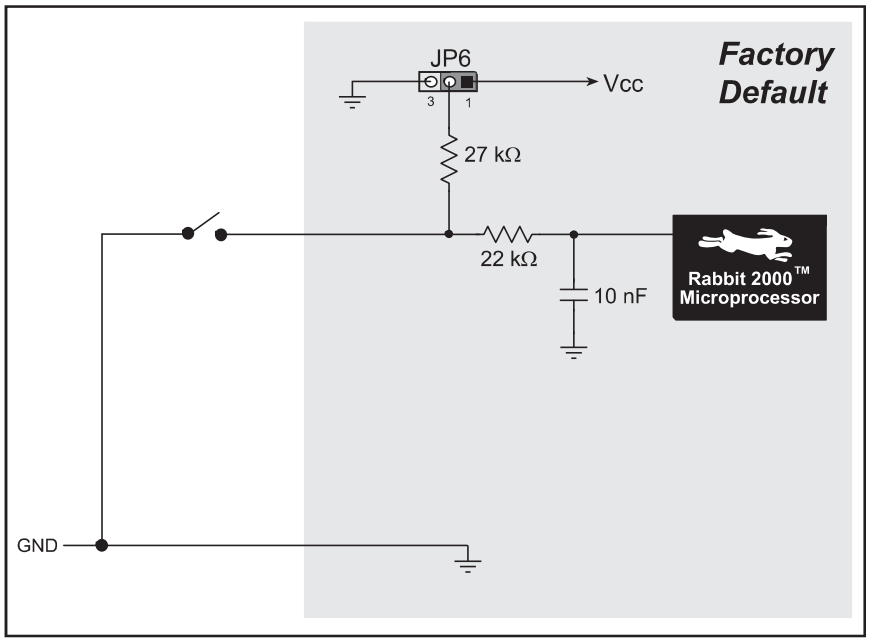


Figure 7(a). BL2000 Digital Inputs [Pulled Up—JP6(1–2) connected]

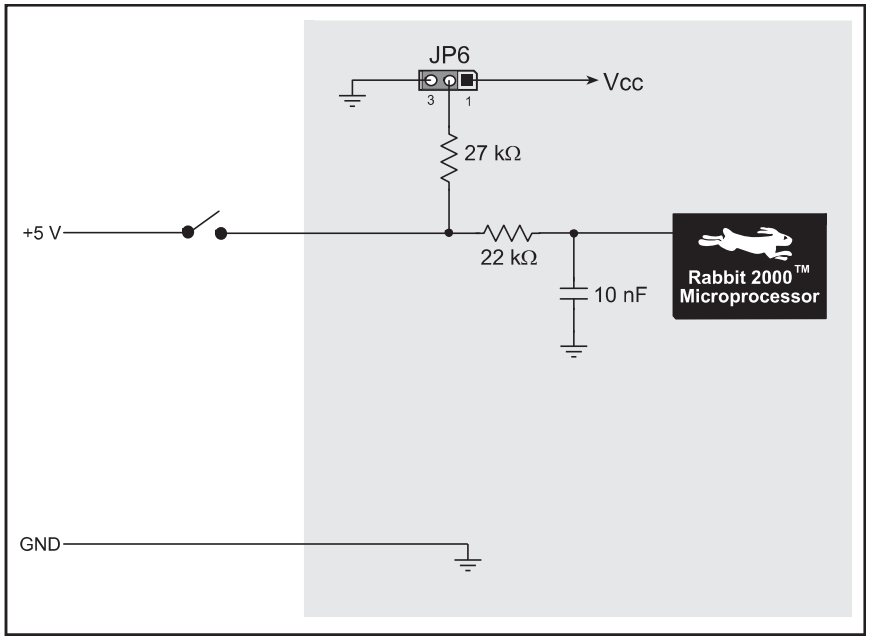


Figure 7(b). BL2000 Digital Inputs [Pulled Down—JP6(2–3) connected]

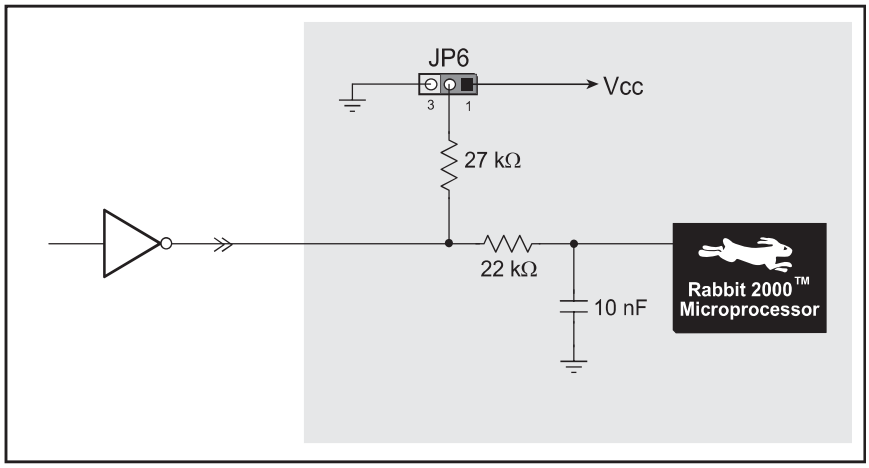


Figure 7(c). Example of Logic Gate Driving BL2000 Digital Input

The actual switching threshold is approximately 2.40 V for channels IN0–IN10. Anything below this value is a logic 0, and anything above is a logic 1.

The A/D converter inputs can be used as additional digital inputs using the parameters specified for the **`digIn`** software function call. The default threshold for channels IN11–IN21 is also set to 2.40 V, but may be changed by adding two lines to your program as discussed for the **`digIn`** software function call.

The digital inputs are each fully protected over a range of -36 V to +36 V, and can handle short spikes of ±40 V.

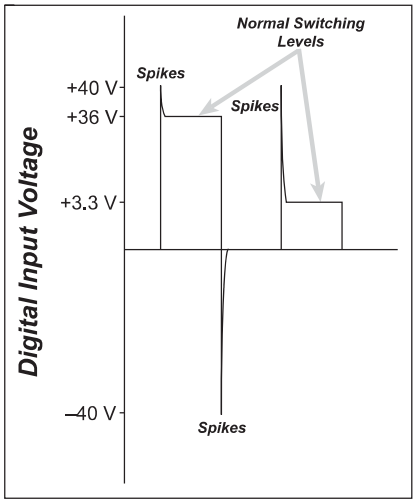


Figure 8. BL2000 Digital Input Protected Range

3.2.2 Digital Outputs

The BL2000 has 10 digital outputs, OUT0–OUT9, each of which can either sink or source up to 200 mA, depending on how the outputs are configured. On boards that carry the CE mark, OUT8 and OUT9 are each capable of sinking up to 750 mA.

Each output can be configured individually as either a sinking or a sourcing output as shown in Figure 9. The outputs can be pulled as a group to Vcc, +K, or GND through 27 kΩ resistors. Tie the outputs high to either Vcc or +K when using the outputs as sinking outputs (via 0 Ω resistors at R32 or R35 respectively), or tie the outputs to GND via R34 when using the outputs as sourcing outputs. +K is an externally supplied voltage of up to 40 V DC, and is used primarily in combination with current sourcing outputs, and ***must*** also be connected to an external supply when an inductive load is connected to a sinking output.

NOTE: Remove the 27 kΩ resistors (R143–R150) from the output circuits if no pull-up/down is required to avoid leakage between the outputs. These resistors are located on the bottom side of the BL2000 board above the solder points for screw terminal header J9.

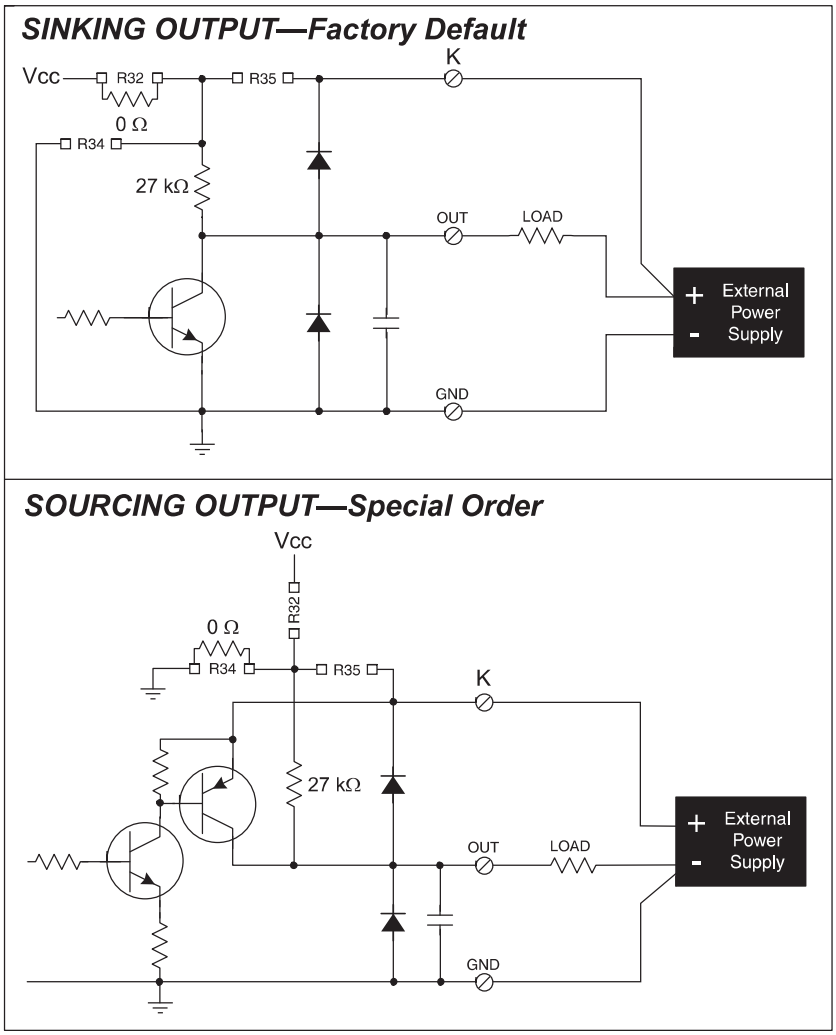


Figure 9. BL2000 Digital Outputs

The locations of the output pull-up/pull-down select resistors R32, R34, and R35 are shown in Figure 10.

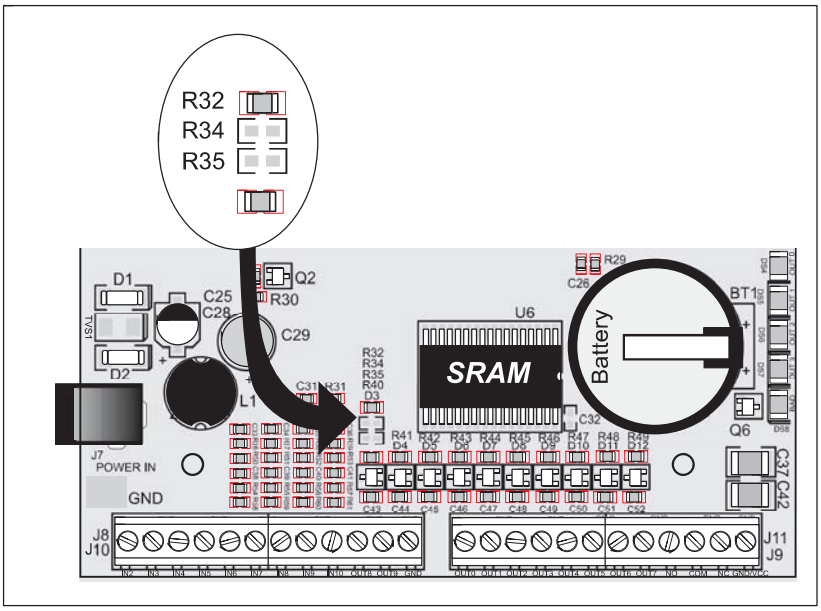


Figure 10. Locations of Resistors R32, R34, and R35

All BL2000 models are factory-configured with sinking outputs and pull-up resistors tied to Vcc via a 0 Ω resistor at R32.

3.3 Relay Outputs

Figure 11 shows the BL2000 relay contact connections. A diode across the coil provides a return path for inductive spikes, and snubbers across the relay contacts protect the relay contacts from inductive spikes.

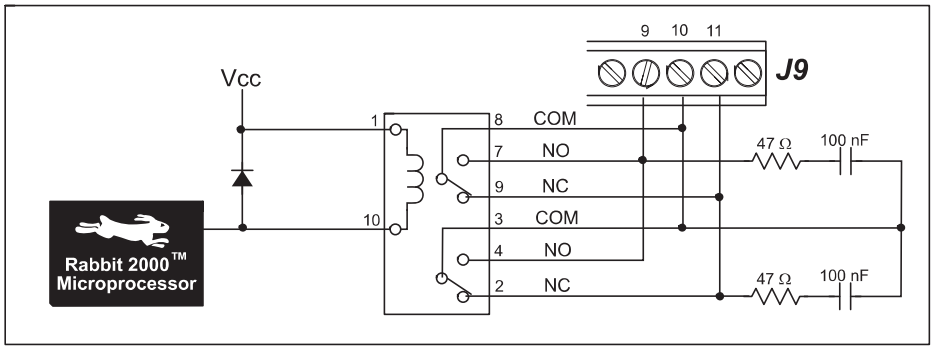


Figure 11. BL2000 Relay Output Contact Connections

The relay is driven by PA0, which is the same Rabbit 2000 parallel port that drives OUT0 and LED DS4. OUT0 therefore works in parallel with the relay output.

The relay included on the BL2000 has contacts rated for 1 A @ 30 V DC or 300 mA @ 120 V AC. When using the BL2000 in a CE-certified application, the voltages handled by the relay must not exceed SELV levels (42.4 V AC peak, or 60 V DC).

3.4 Serial Communication

The BL2000 has one RS-232 serial channel (with RTS/CTS) or two RS-232 (3-wire) channels, one RS-485 serial channel, and one CMOS serial channel. The RS-232 channel(s) are configured with the **`serMode`** software function call. Table 2 summarizes the options.

Table 2. Serial Communication Configurations

Mode	Serial Port		
	B	C	D
0	RS-232, 3-wire	RS-232, 3-wire	RS-485
1	RS-232, 5-wire	CTS/RTS	RS-485

All four serial ports operate in an asynchronous mode. 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 Port A 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. The BL2000 series boards typically use all four ports in the asynchronous serial mode. Serial Ports B and C are used for RS-232 communication, and Serial Port D is used for RS-485 communication. The BL2000 uses an 11.0592 MHz crystal, which is doubled to 22.1184 MHz. At this frequency, the BL2000 supports standard baud rates up to a maximum of 230,400 bps.

3.4.1 RS-232

The BL2000 RS-232 serial communication is supported by an RS-232 transceiver, U1. U1 provides the voltage output, slew rate, and input voltage immunity required to meet the RS-232 serial communication protocol. Basically, the chip translates the Rabbit 2000's CMOS/TTL signals to RS-232 signal levels. Note that the polarity is reversed in an RS-232 circuit so that a +5 V output becomes approximately -10 V and 0 V is output as +10 V. U1 also provides the proper line loading for reliable communication.

RS-232 can be used effectively at this baud rate for distances up to 15 m.

3.4.2 RS-485

The BL2000 has one RS-485 serial channel, which is connected to the Rabbit 2000 Serial Port D through U8, an RS-485 transceiver. U8 supports the RS-485 serial communication protocol. The chip's slew rate limiters provide for a maximum baud rate of 230,400 bps, which allows for a network of up to 300 m (or 1000 ft). The half-duplex communication uses the Rabbit 2000's PB6 pin to control the transmit enable on the communication line.

The BL2000 can be used in an RS-485 multidrop network. Connect the 485+ to 485+ and 485– to 485– using single twisted-pair wires (nonstranded, tinned) as shown in Figure 12. Note that a common ground is recommended.

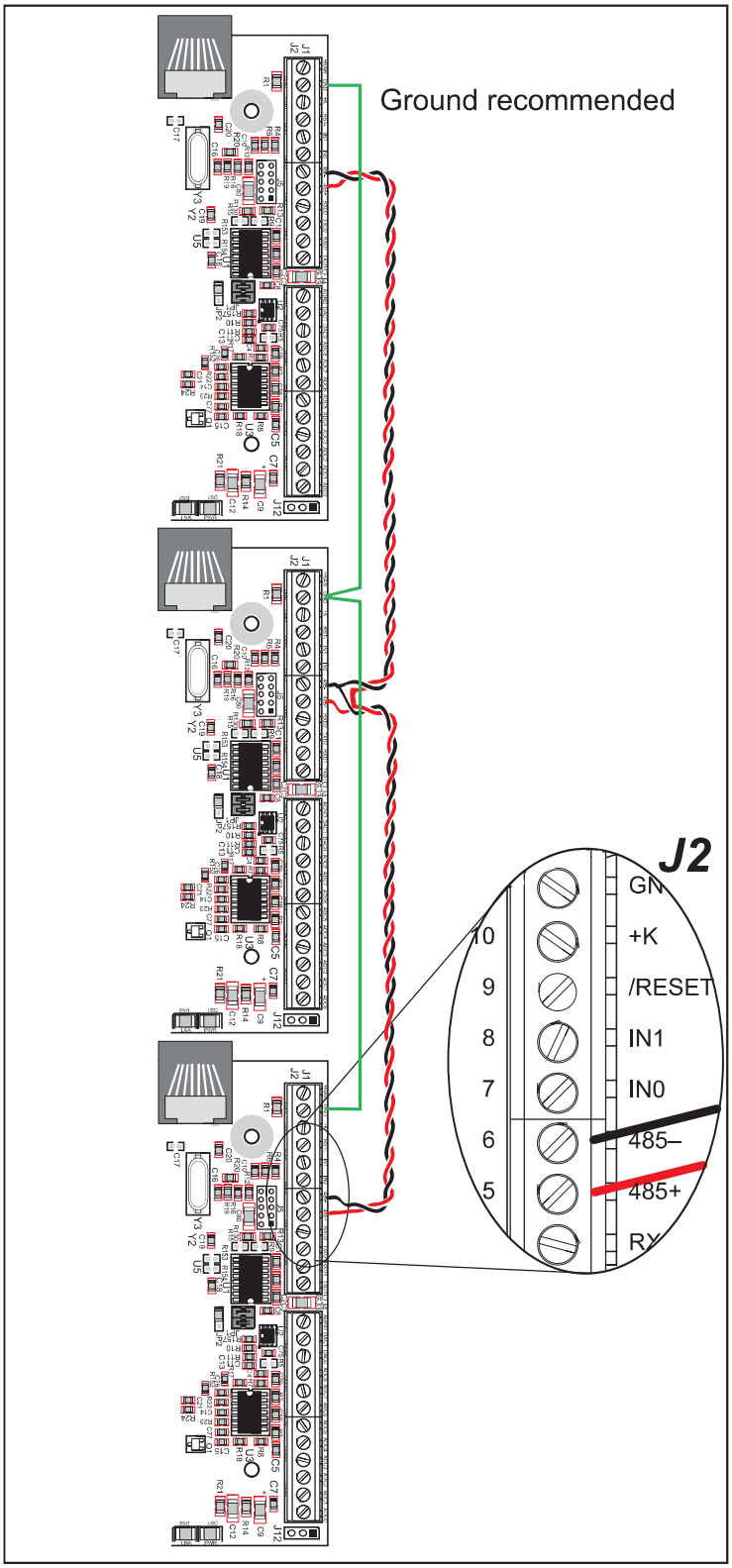


Figure 12. Multidrop BL2000 Network

The BL2000 comes with a 220 Ω termination resistor and two 681 Ω bias resistors installed and enabled with jumpers across pins 1–2 and 3–4 on header JP1, as shown in Figure 13.

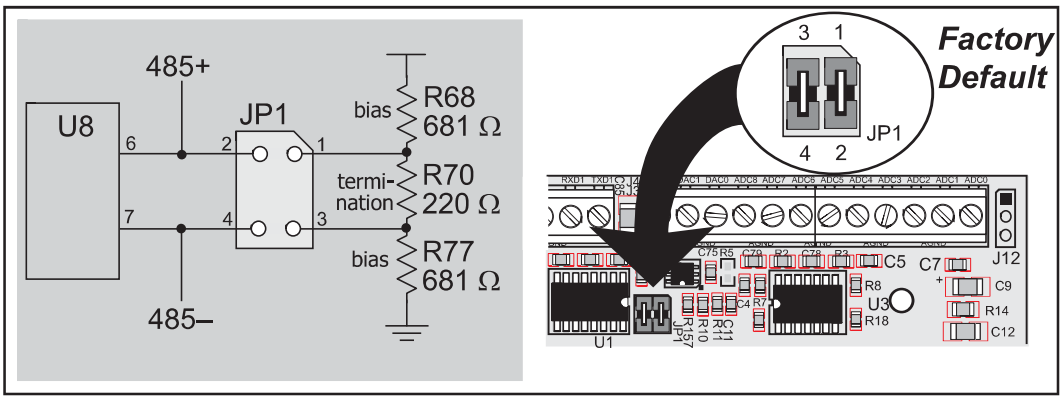


Figure 13. RS-485 Termination and Bias Resistors

The bias and termination resistors in a multidrop network should only be enabled on both end nodes of the network. Disable the termination and bias resistors on the intervening BL2000 units in the network by removing both jumpers from header JP1.

3.4.3 Ethernet Port

Figure 14 shows the pinout for the Ethernet port (J6). Note that there are two standards for numbering the pins on this connector—the convention used here, and numbering in reverse to that shown. Regardless of the numbering convention followed, the pin positions relative to the spring tab position (located at the bottom of the RJ-45 jack in Figure 14) are always absolute, and the RJ-45 connector will work properly with off-the-shelf Ethernet cables.

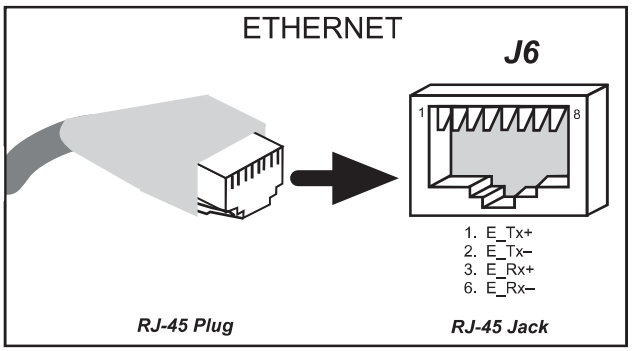


Figure 14. RJ-45 Ethernet Port Pinout

RJ-45 pinouts are sometimes numbered opposite to the way shown in Figure 14.

The transformer/connector assembly ground is connected to the BL2000 printed circuit board digital ground via a 0 Ω resistor "jumper," R1, as shown in Figure 15.

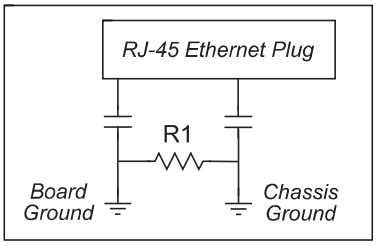


Figure 15. Isolation Resistor R1

The factory default is for the 0 Ω resistor "jumper" at R1 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 ESD and/or EMI problems. Once you have removed the 0 Ω resistor "jumper," R1, use a ring lug to attach the BL2000 to the chassis ground, thereby grounding the transformer/connector assembly.

A convenient position for the ring lug has been provided at the top-left mounting screw hole near the RJ-45 jack as shown in Figure 16.

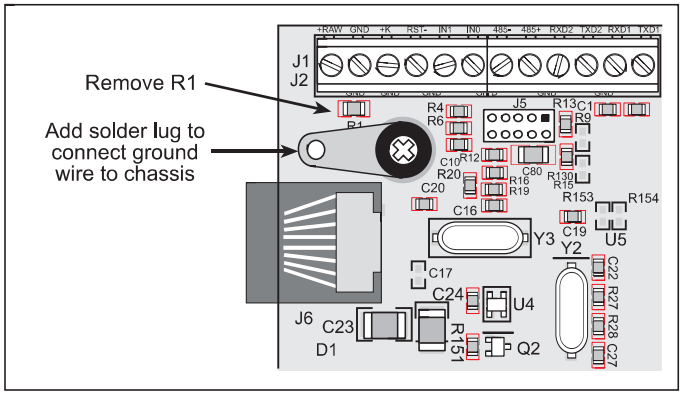


Figure 16. Recommended Location for Ring Lug

3.4.4 Programming Port

The BL2000 has a 10-pin programming header labeled J5. The programming port uses the Rabbit 2000's Serial Port A for communication. Dynamic C uses the programming port to download and debug programs.

The programming port is also used for the following operations.

- Cold-boot the Rabbit 2000 after a reset.
- Remotely download and debug a program over an Ethernet connection using the RabbitLink EG2110.
- Fast copy designated portions of flash memory from one Rabbit-based board (the master) to another (the slave) using the Rabbit Cloning Board.

Alternate Uses of the Serial Programming Port

All three clocked Serial Port A signals are available as

- a synchronous serial port
- an asynchronous serial port, with the clock line usable as a general CMOS input

The serial programming port may also be used as a serial port via the **DIAG** connector on the serial programming cable.

In addition to Serial Port A, the Rabbit 2000 startup-mode (SMODE0, SMODE1), status, and reset pins are available on the serial programming port.

The two startup mode pins determine what happens after a reset—the Rabbit 2000 is either cold-booted or the program begins executing at address 0x0000. These two SMODE pins can be used as general inputs once the cold boot is complete.

The status pin is used by Dynamic C to determine whether a Rabbit microprocessor is present. 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 /RESET_IN pin is an external input that is used to reset the Rabbit 2000 and the onboard peripheral circuits on the RabbitCore module. The serial programming port can be used to force a hard reset on the RabbitCore module by asserting the /RESET_IN signal.

Refer to the ***Rabbit 2000 Microprocessor User's Manual*** for more information.

3.5 A/D Converter Inputs

The single 14-channel A/D converter used in the BL2000 has a resolution of 12 bits (models BL2000 and BL2020) or 10 bits (models BL2010 and BL2030). Eleven of the 14 channels are available externally, and three are used internally for the reference voltages: 4.096 V (V_{ref}), 2.048 V ($V_{ref}/2$), and Analog Ground. These internal voltages can be used to check the functioning of the A/D converter.

The A/D converter only measures voltages between 0 V and the applied reference voltage. Therefore, each external input has circuitry that provides scaling and buffering. The first four external inputs are scaled and buffered to provide the user with an input impedance of 1 MΩ and a range of -10.24 V to +10.24 V. The remaining five or seven inputs are not buffered, but are scaled to provide inputs that can range from 0 V to +49 V.

Figure 17 shows the buffered A/D converter inputs.

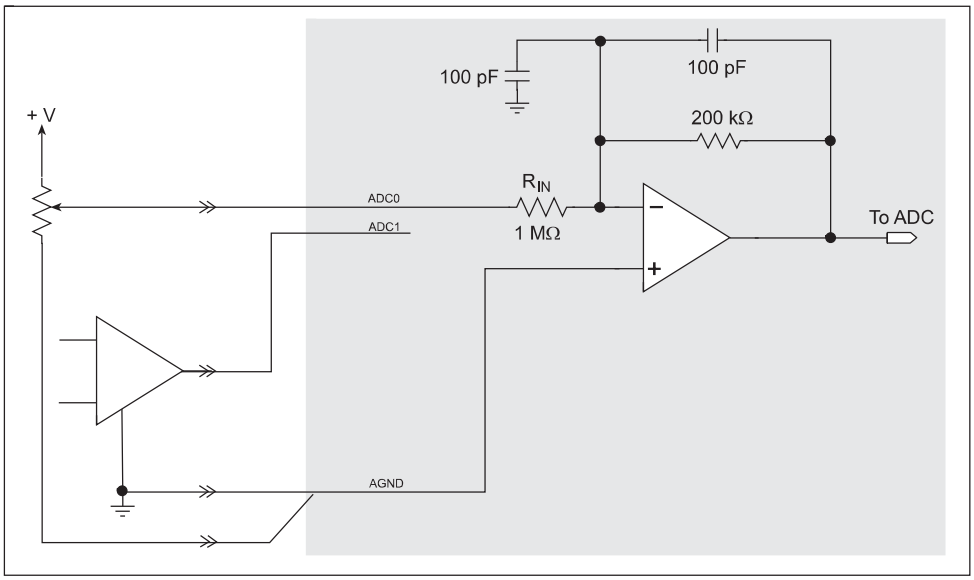


Figure 17. Buffered A/D Converter Inputs

The op-amp is powered from the +V supply. The 1 MΩ and 200 kΩ resistors set the gain (scale factor), which is 0.2 in this case. This results in a dynamic input range of 4.096 V ÷ 0.2 or 20.48 V. The center point of this range is set by the 1.707 V reference voltage. With the reference set to 1.707 V, the center point is at 0 V and the input voltage can range from -10.24 V to +10.24 V. To maintain the best accuracy, the input range should be limited to -10.0 V to +10.0 V.

The five or seven unbuffered inputs have an impedance of 12 kΩ and a scale factor of 0.0833, which provides for an input voltage range of 0 V to 49.15 V. Accuracy is maintained over the specified voltage range from 0 V to 48 V DC.

The analog inputs can also be used as digital inputs when required. In this case a lower quality 10-bit D/A converter can be used, and the software would assign a 1 or 0 to a voltage based on whether it is above or below a particular threshold. See the **`digIn`** function description for more information.

3.6 D/A Converter Outputs

Figure 18 shows the analog voltage reference circuit.

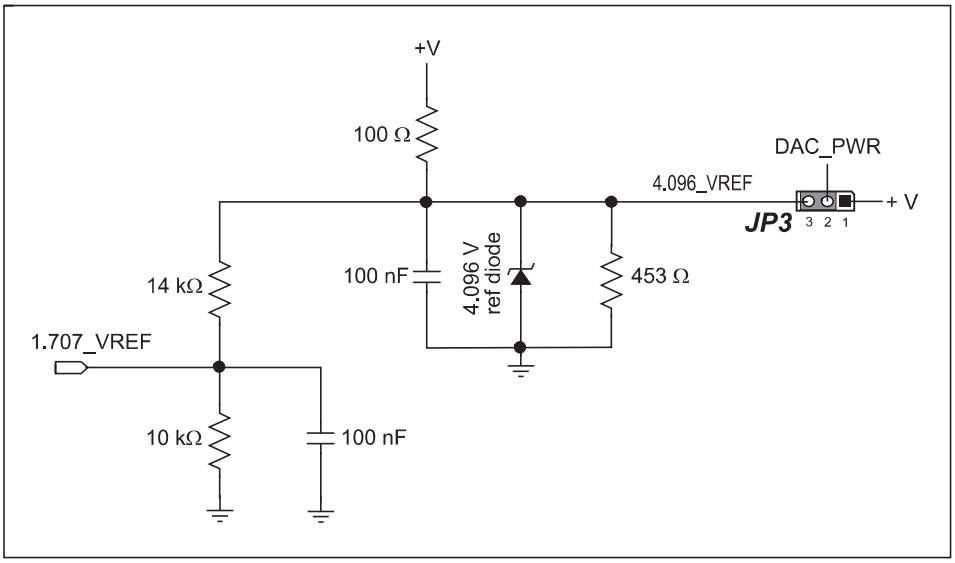


Figure 18. Analog Reference Voltages

This circuit generates the 4.096 V reference voltage, which is used by the A/D converter and optionally by the two D/A converters. This sets the operating range of the A/D converter and the D/A converters (0–4.096 V). To use the full accuracy of the A/D converter and the D/A converters, this voltage must be accurate to the same degree.

Under normal operation, the 453 Ω resistor is not installed. The reference zener diode in combination with the 100 Ω resistor form a shunt regulator. The 4.096 V reference voltage then feeds the A/D converter, the D/A converters, and the voltage divider composed of the 10 kΩ and the 14 kΩ resistors. The voltage divider generates a second reference voltage of 1.707 V to feed the four op-amps for the buffered A/D converter inputs.

The reference voltage can be ratiometric rather than absolute. This is done by removing the zener diode and installing the 453 Ω resistor. With this arrangement, the reference voltages follow changes in the power supply voltages Vcc and V+, which is a filtered version of Vcc. This type of measurement circuit is preferred by some customers whose sensors are powered from the Vcc supply and hence the outputs track Vcc.

A jumper on header JP3 allows the D/A converters to be powered either from the 4.096 V reference (factory default) or from the analog supply +V. The D/A converters use their power source also as the reference input, so normally powering the D/A converters from the more accurate 4.096 V reference is best. However, should a customer desire more dynamic range (0–5 V rather than 0–4.096 V), the jumper across JP3 can be set to power the D/A converters from +V. When powered from the +V supply, the outputs of the D/A converters will always be ratiometric, independent of whether the zener diode is installed.

Only the BL2000 and the BL2020 models are stuffed with D/A converters. The D/A converters provide only a voltage output. This means that in order to maintain the maximum accuracy of the D/A converters, only a small amount of current should be drawn from the D/A converter output (of the order of µA).

With D/A converters installed, the user has the option of using an unbuffered A/D converter input to read the output of a D/A converter or one of the two fixed voltages +V or Vcc. The standard BL2000 configuration is for A/D converter channels 9 and 10 to monitor D/A converter channels 0 and 1 respectively.

Figure 19 shows the D/A converter outputs with buffer amplifiers, which may be used to increase the D/A converter output voltage range to 0 V to +10 V.

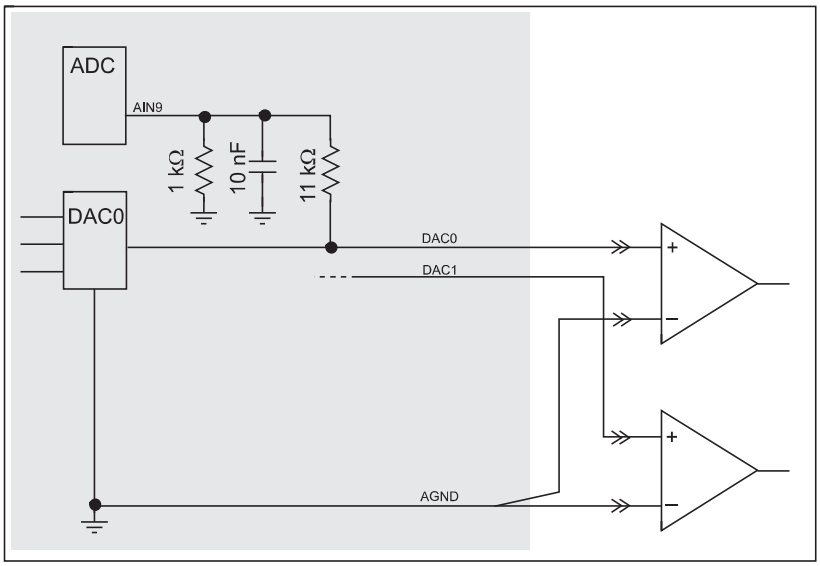


Figure 19. D/A Converter Outputs

3.7 Memory

Section A.3, “Jumper Configurations,” shows where the 0 Ω surface-mounted “jumpers” described in this section are found.

3.7.1 SRAM

The BL2000 is designed to accept 128K to 512K of SRAM packaged in an SOIC case.

The standard models come with 128K of SRAM. Table 3 lists the jumper settings for the jumpers used to set the SRAM size. The “jumpers” are 0 Ω surface-mounted resistors.

Table 3. Memory Jumper Selections

SRAM (JP5)		Flash Memory (JP4)	
1–2	128K	1–2	128K/256K
2–3	512K	2–3	512K

3.7.2 Flash Memory

The BL2000 is also designed to accept 128K to 512K of flash memory packaged in a TSOP case.

The BL2000 comes with one 256K flash memory. Table 3 lists the jumper settings for the jumpers used to set the SRAM size. The “jumpers” are 0 Ω surface-mounted resistors.

> **NOTE:** Rabbit recommends that any customer applications should not be constrained by the sector size of the flash memory since it may be necessary to change the sector size in the future.

A Flash Memory Bank Select jumper configuration option exists at JP2 with 0 Ω surface-mounted resistors. This option, used in conjunction with some configuration macros, allows Dynamic C to compile two different co-resident programs for the upper and lower halves of the 256K flash in such a way that both programs start at logical address 0000. This is useful for applications that require a resident download manager and a separate downloaded program. See Technical Note 218, ***Implementing a Serial Download Manager for a 256K Flash***, for details.

3.8 Programming Cable

The programming cable is used to connect the BL2000's programming port to a PC serial COM port. The programming cable converts the RS-232 voltage levels used by the PC serial port to the TTL voltage levels used by the Rabbit 2000.

When the **PROG** connector on the programming cable is connected to the BL2000's programming header, programs can be downloaded and debugged over the serial interface.

The **DIAG** connector of the programming cable may be used on the BL2000's programming header with the BL2000 operating in the Run Mode. This allows the programming port to be used as a regular serial port.

3.8.1 Changing Between Program Mode and Run Mode

The BL2000 is automatically in Program Mode when the **PROG** connector on the programming cable is attached to the BL2000, and is automatically in Run Mode when no programming cable is attached. When the Rabbit 2000 is reset, the operating mode is determined by the status of the SMODE pins. When the programming cable's **PROG** connector is attached, the SMODE pins are pulled high, placing the Rabbit 2000 in the Program Mode. When the programming cable's **PROG** connector is not attached, the SMODE pins are pulled low, causing the Rabbit 2000 to operate in the Run Mode.

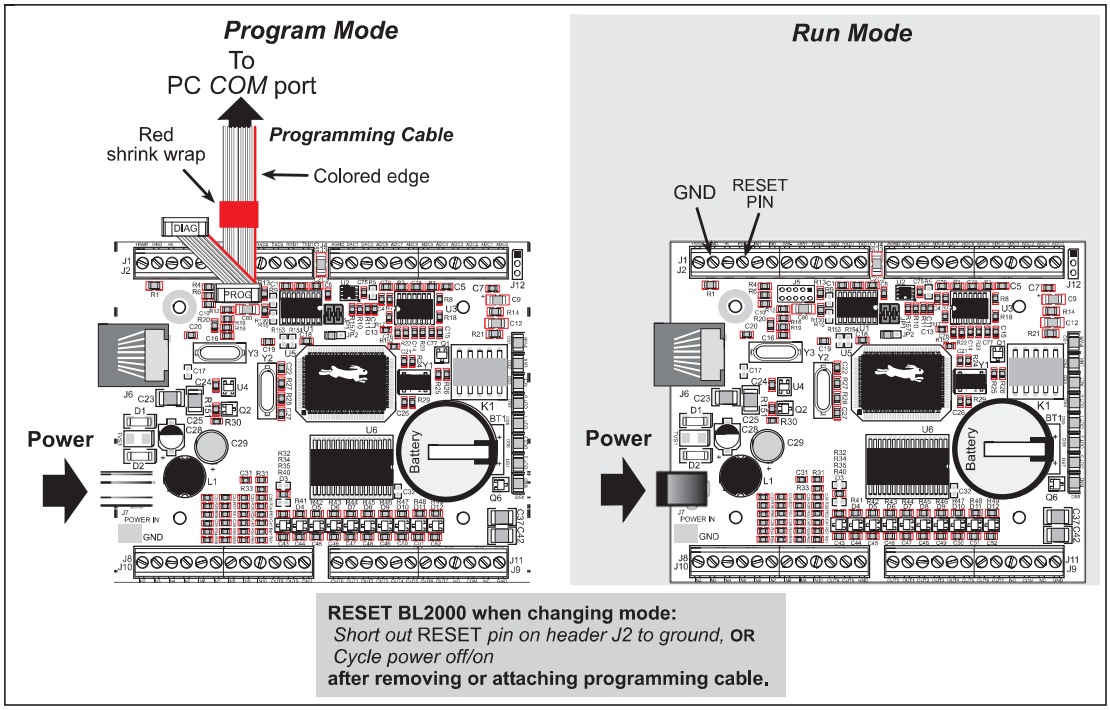


Figure 20. BL2000 Program Mode and Run Mode Setup

A program "runs" in either mode, but can only be downloaded and debugged when the Jackrabbit is in the Program Mode.

Refer to the ***Rabbit 2000 Microprocessor User's Manual*** for more information on the programming port and the programming cable.

3.9 Other Hardware

3.9.1 External Interrupts

BL2000 boards with a Rabbit 2000 microprocessor labeled IQ3T or higher have external interrupts available on digital inputs IN2 and IN3. Older BL2000 boards (Rabbit 2000 microprocessors labeled IQ2T) have one external interrupt available—see Technical Note TN301, ***Rabbit 2000 Microprocessor Interrupt Problem***, for further information on how to use this interrupt on the older boards.

3.9.2 Clock Doubler

The BL2000 takes advantage of the Rabbit 2000 microprocessor's internal clock doubler. A built-in clock doubler allows half-frequency crystals to be used to reduce radiated emissions. The 22.1 MHz frequency is generated using an 11.0592 MHz crystal. The clock doubler is disabled automatically in the BIOS for crystals with a frequency above 12.9 MHz.

The clock doubler may be disabled if 22.1 MHz clock speeds are not required. Disabling the Rabbit 2000 microprocessor's internal clock doubler will reduce power consumption and further reduce radiated emissions. The clock doubler is disabled with a simple configuration macro as shown below.

1. Select the "Defines" tab from the Dynamic C **Options > Project Options** menu.
2. Add the line `CLOCK_DOUBLED=0` to always disable the clock doubler.

   The clock doubler is enabled by default, and usually no entry is needed. If you need to specify that the clock doubler is always enabled, add the line `CLOCK_DOUBLED=1` to always enable the clock doubler. The clock speed will be doubled as long as the crystal frequency is less than or equal to 26.7264 MHz.
3. Click **OK** to save the macro. The clock doubler will now remain off whenever you are in the project file where you defined the macro.

3.9.3 Spectrum Spreader

BL2000 boards that carry the CE mark have a Rabbit 2000 microprocessor that features a spectrum spreader, which helps to mitigate EMI problems. By default, the spectrum spreader is on automatically for BL2000 boards that carry the CE mark when used with Dynamic C 7.32 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 unnecessary for the BL2000.

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 BL2000 boards that do not carry the CE mark or when using any BL2000 with a version of Dynamic C prior to 7.30.

4. SOFTWARE

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-based single-board computers and other devices based on the Rabbit microprocessor.

Chapter 4 provides the libraries, function calls, and sample programs related to the BL2000.

4.1 An Overview of Dynamic C

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

You have a choice of doing your software development in the flash memory or in the data SRAM included on the Jackrabbit. 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 Jackrabbit 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 and later. 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-based 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.

4.1.1 Upgrading Dynamic C

4.1.1.1 Patches and Bug Fixes

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 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.

4.1.1.2 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 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.

4.2 Sample Programs

Sample programs are provided in the Dynamic C **SAMPLES** folder. The sample program **PONG.C** demonstrates the output to the **STDIO** window. The various directories in the **SAMPLES** folder contain specific sample programs that illustrate the use of the corresponding Dynamic C libraries.

The **SAMPLES\BL2000** folder provides sample programs specific to the BL2000. Each sample program has comments that describe the purpose and function of the program. Follow the instructions at the beginning of the sample program.

To run a sample program, open it with the **File** menu (if it is not still open), compile it using the **Compile** menu, and then run it by selecting **Run** in the **Run** menu. The BL2000 must be in **Program** mode (see Section 3.8, "Programming Cable,") and must be connected to a PC using the programming cable as described in Section 2.1, "BL2000 Connections."

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

4.2.1 General BL2000 Sample Programs

- **BOARD_ID.C**—This program is used to identify the model of BL2000 being used, and displays that information in the **STDIO** window.
- **COUNTLEDS.C**—This program will count from 0 to 31 in binary, using the four general-purpose LEDs, DS4–DS7, and the Processor Bad LED, DS8. The LEDs are used in reverse logical order to minimize the cycling of the relay, which is slaved to the same output as DS4.
- **LEDS_4.C**—This program creates four "devices" (lights), and four buttons to toggle them. Users can view the devices with their Web browser, and change the status of the lights. If the Demonstration Board is connected to the BL2000, the lights on the Demonstration Board will match the ones on the Web page. See Appendix D for hookup instructions for the Demonstration Board.

4.2.2 Digital I/O

The following sample programs are found in the **IO** subdirectory in **SAMPLES/BL2000**.

- **ANADIGIN.C**—Demonstrates using the A/D converter channels as digital inputs. You will be able to see an input channel toggle HIGH and LOW when pressing the pushbuttons on the Demonstration Board. See Appendix D for hookup instructions for the Demonstration Board.
- **DIGIN.C**—Demonstrates the use of the digital inputs. Using the Demonstration Board, you can see an input channel toggle from HIGH to LOW when pressing a pushbutton on the Demonstration Board. See Appendix D for hookup instructions for the Demonstration Board.
- **DIGOUT.C**—Demonstrates the use of the high-current outputs. Using the Demonstration Board, you can see an LED toggle on/off via a high-current output. See Appendix D for hookup instructions for the Demonstration Board.
- **LED.C**—Demonstrates how to toggle the output LEDs on the BL2000 on/off.

- **PWM.C**—Demonstrates the use of Timer B to generate a PWM signal on digital output OUT8. The program generates a 42 Hz PWM signal with the duty cycle adjustable from 1 to 99%.
- **RELAY.C**—Demonstrates how to control the relay on the BL2000.

4.2.3 Serial Communication

The following sample programs are found in the **RS232** subdirectory in **SAMPLES/BL2000**.

- **PUTS.C**—Transmits and then receives an ASCII string on Serial Ports B and C. It also displays the serial data received from both ports in the **STDIO** window.
- **RELAYCHR.C**—This program echoes characters over Serial Port B to Serial Port C. It must be run with a serial utility such as Hyperterminal.

The following sample programs are found in the **RS485** subdirectory in **SAMPLES/BL2000**.

- **MASTER.C**—This program demonstrates a simple RS-485 transmission of lower case letters to a slave BL2000. The slave will send back converted upper case letters back to the master BL2000 and display them in the **STDIO** window. Use **SLAVE.C** to program the slave BL2000.
- **SLAVE.C**—This program demonstrates a simple RS-485 transmission of lower case letters to a slave BL2000. The slave will send back converted upper case letters back to the master BL2000 and display them in the **STDIO** window. Use **MASTER.C** to program the master BL2000.

4.2.4 A/D Converter Inputs

The following sample programs are found in the **ADC** subdirectory in **SAMPLES/BL2000**.

- **AD_CALIB.C**—Demonstrates how to recalibrate an A/D converter channel using two known voltages to generate two coefficients, gain and offset, which are rewritten into the user block data area. The voltage that is being monitored is displayed continuously. Note that this sample program will overwrite the calibration constants set at the factory.
- **AD1.C**—Demonstrates how to access the A/D internal test voltages in both the TLC2543 and TLC1543 A/D converter chips. The program reads the A/D internal voltages and then uses the **STDIO** window to display the RAW data.
- **AD2.C**—Demonstrates how to access the A/D channels using the **anaInVolt** function. The program uses the **STDIO** window to display the voltage that is being monitored.
- **AD3.C**—Demonstrates how to access the A/D converter channels with the low-level A/D driver. The program uses the **STDIO** window to display the voltage that is being monitored on all the A/D channels using the low-level A/D driver.
- **AD4.C**—Demonstrates how to use the A/D converter channels with the low-level A/D driver. The program uses the **STDIO** window to display the voltage (average of 10 samples) that is being monitored on all the A/D converter channels using the low-level A/D driver.

4.2.5 D/A Converter Outputs

The following sample programs are found in the **DAC** subdirectory in **SAMPLES/BL2000**.

- **DACAL.C**—This program demonstrates how to recalibrate an D/A converter channel using two known voltages, and defines the two coefficients, gain and offset, that will be rewritten into the D/A converter's EEPROM simulated in flash memory. Note that this sample program will overwrite the calibration constants set at the factory.
- **DAOUT1.C**—This program outputs a voltage that can be read with a voltmeter. The output voltage is computed using the calibration constants that are read from the EEPROM simulated in flash memory.
- **DAOUT2.C**—This program demonstrates the use of both the D/A and the A/D converters. The user selects both the D/A converter and A/D channel to be used, then sets the D/A converter output voltage to be read by the A/D channel. All activity will be displayed in the **STDIO** window.

4.2.6 Real-Time Clock

If you plan to use the real-time clock functionality in your application, you will need to set the real-time clock. You may set the real-time clock using the **SETRTCKB.C** sample program from the Dynamic C **SAMPLES\RTCLOCK** folder. The **RTC_TEST.C** sample program in the Dynamic C **SAMPLES\RTCLOCK** folder provides additional examples of how to read and set the real-time clock

4.2.7 TCP/IP Sample Programs

TCP/IP sample programs are described in Chapter 5.

4.3 BL2000 Libraries

Two library directories are used to develop applications for the BL2000.

- **`BL2000-`**—libraries associated with features specific to the BL2000.
- **`TCPIP`**—libraries specific to using TCP/IP functions on the BL2000.

Other generic functions applicable to all devices based on the Rabbit 2000 microprocessor are described in the ***Dynamic C Function Reference Manual***.

4.4 BL2000 Function Calls

4.4.1 Board Initialization

```
void brdInit (void);
```

Call this function at the beginning of your program. This function initializes the system I/O ports and loads all the A/D and DAC calibration constants from flash memory into SRAM for use by your program.

The ports are initialized as follows:

Port	I/O	Function	Output Function State
PA0	Output	OUT0/RELAY/LED_DS4	High-Current Driver Off
PA1	Output	OUT1/LED_DS5	High-Current Driver Off
PA2	Output	OUT2/LED_DS6	High-Current Driver Off
PA3	Output	OUT3/LED_DS7	High-Current Driver Off
PA4	Output	OUT4	High-Current Driver Off
PA5	Output	OUT5	High-Current Driver Off
PA6	Output	OUT6	High-Current Driver Off
PA7	Output	OUT7	High-Current Driver Off
PB0	Input	IN6	N/A
PB1	Input	CLKA	N/A
PB2	Input	IN7	N/A
PB3	Input	IN8	N/A
PB4	Input	IN9	N/A
PB5	Input	IN10	N/A
PB6	Output	RS485_EN	Off
PB7	Output	UPGOOD	Bad Indicator Off
PC0	Output	TXD RS-485	Inactive high
PC1	Input	RXD RS-485	N/A

Port	I/O	Function	Output Function State
PC2	Output	RTS/TXC RS-232	Inactive high
PC3	Input	CTS/RXC RS-232	N/A
PC4	Output	TXB RS-232	Inactive high
PC5	Input	RXB RS-232	N/A
PC6	Output	TXA Programming Port	Inactive high
PC7	Input	RXA Programming Port	N/A
PD0	Output	DAC-ADC_SK	On
PD1	Output	DAC-ADC_SDI	On
PD2	Input	RTL-ADC_SDO	N/A
PD3	Input	RTL_SK	N/A*
PD4	Output	RTL_SDI	On
PD5	Output	DAC0_CS	Inactive high
PD6	Output	DAC1_CS	Inactive high
PD7	Output	ADC_CS	Inactive high
PE0	Output	OUT8	High-Current Driver Off
PE1	Output	OUT9	High-Current Driver Off
PE2	Input	IN0	N/A
PE3	Input	IN1	N/A
PE4	Input	IN2	N/A
PE5	Input	IN3	N/A
PE6	Input	IN4	N/A
PE7	Input	IN5	N/A

* PD3 is an output (and is on) for the BL2020 and the BL2030.

SEE ALSO

`digOut, digIn, serMode`

4.4.2 Digital I/O

int digIn(int channel);

Reads the state of an input channel:

IN0–IN10—standard digital inputs, ± 36 V DC

IN11–IN14—pseudo digital inputs using A/D converter inputs ADC0–ADC3, ± 10 V DC

IN15–IN19—pseudo digital inputs using A/D converter inputs ADC4–ADC8, 0 V to 48 V DC

IN20–IN21—pseudo digital inputs using A/D converter inputs DAC0–DAC1, 0 V to 48 V DC (BL2010 and BL2030)

The threshold is fixed at 2.40 V for channels IN0–IN10. Anything below 2.40 V is a logic 0, and anything higher than or equal to 2.40 V is a logic 1.

The default threshold for channels IN11–IN21 is also set to 2.40 V. The threshold for these channels may be changed by adding the following two lines to your program.

```
#undef   THRESHOLD
#define  THRESHOLD xx.xx
```

where **xx.xx** is the desired threshold voltage. Anything below the threshold value is a logic 0, and anything higher than or equal to the threshold value is a logic 1.

PARAMETER

channel is the input channel number (0–21)

RETURN VALUE

The state of the input (0 or 1).

SEE ALSO

brdInit, digOut

void digOut(int channel, int value);

Sets the state of a digital output (OUT0–OUT9).

The default setting for the function is for current-sinking outputs. To change from sinking to sourcing outputs, add the following two lines at the beginning of your program.

```
#undef   OUTPUT_DRIVE
#define  OUTPUT_DRIVE  SOURCING
```

The relay is driven by PA0, which is the same Rabbit 2000 parallel port that drives OUT0 and LED DS4. OUT0 therefore works in parallel with the relay output. Rabbit therefore recommends that you do not use OUT0 for a digital output when you are using the relay.

PARAMETERS

channel is the output channel number (0–9).

value is the output value (0 or 1).

SEE ALSO

brdInit, digIn

4.4.3 Serial Communication

Library files included with Dynamic C provide a full range of serial communications support. The **RS232.LIB** library provides a set of circular-buffer-based serial functions. The **PACKET.LIB** library provides packet-based serial functions where packets can be delimited by the 9th bit, by transmission gaps, or with user-defined special characters. Both libraries provide blocking functions, which do not return until they are finished transmitting or receiving, and nonblocking functions, which must be called repeatedly until they are finished. For more information, see the ***Dynamic C Function Reference Manual*** and Technical Note 213, ***Rabbit Serial Port Software***.

The following function calls are specific to the BL2000.

```
int serMode(int mode);
```

User interface to set up BL2000 serial communication lines. Call this function after **serXOpen()**.

If Mode 1 is selected, CTS/RTS flow control is exercised using the **serCflowcontrolOn()** and **serCflowcontrolOff()** functions from the **RS232.LIB** library.

PARAMETER

mode is the defined serial port configuration.

Mode	Serial Port		
	B	C	D
0	RS-232, 3-wire	RS-232, 3-wire	RS-485
1	RS-232, 5-wire	CTS/RTS	RS-485

RETURN VALUE

0 if valid mode, 1 if not.

SEE ALSO

ser485Tx, ser485Rx

```
void ser485Tx(void);
```

Sets (high) pin 3 (DE) to enable Tx.

SEE ALSO

serMode, ser485Rx

```
void ser485Rx(void);
```

Resets (low) pin 3 (DE) to disable Tx.

SEE ALSO

serMode, ser485Tx

4.4.4 Relay and LED Outputs

```
void relayOut(int relay, int value);
```

Sets the state of a relay.

The relay is driven by PA0, which is the same Rabbit 2000 parallel port that drives OUT0 and LED DS4. OUT0 therefore works in parallel with the relay output. Rabbit therefore recommends that you do not use OUT0 for a digital output when you are using the relay.

PARAMETERS

`relay` is the relay to control, 0 = Relay 0.

`value` is the value used to connect the relay common contact to one of the following contacts:

0 = relay common connected to relay normally closed contact

1 = relay common connected to relay normally open contact

SEE ALSO

`brdInit`

```
void ledOut(int led, int value);
```

LED ON/OFF control.

The relay is driven by PA0, which is the same Rabbit 2000 parallel port that drives OUT0 and LED0. OUT0 therefore works in parallel with the relay output. Rabbit therefore recommends that you do not use OUT0 for a digital output when you are using the relay. The relay and OUT0 are also turned on when LED0 is turned on.

PARAMETERS

`led` is the LED to control:

0 = OUT0 LED
1 = OUT1 LED
2 = OUT2 LED
3 = OUT3 LED
4 = BAD indicator

value is the value used to control the LED:

0 = OFF
1 = ON

SEE ALSO

`brdInit`

4.4.5 A/D Converter Inputs

```
void anaInCalib(int channel, int value1,
   float volts1, int value2, float volts2);
```

Calibrates the response of the A/D converter channel as a linear function using the two conversion points provided. Gain and offset constants are calculated and placed into global table **`_adcInCalib`**.

PARAMETERS

`channel` is the A/D converter input channel (0–10).

`value1` is the first A/D converter channel value.

`volts1` is the voltage corresponding to the first A/D converter channel value.

`value2` is the second A/D converter channel value.

`volts2` is the voltage corresponding to the second A/D converter channel value.

RETURN VALUE

0 if successful.

-1 if not able to make calibration constants.

SEE ALSO

`anaIn, anaInVolts, brdInit`

```
int _anaIn(unsigned char cmd, char len);
```

Reads the voltage of an analog input channel by serially clocking out an 8-bit command to the A/D converter device of the following formats:

TLC2543 commands

D7–D4

Channel 0 - 10
Channel 11 = $(V_{ref+} - V_{ref-})/2$
Channel 12 = V_{ref-}
Channel 13 = V_{ref+}
Channel 14 = software powerdown

D3-D2

Output data length:
01—8 bits
00—12 bits (normally used as default)
11—16 bits (not supported by driver)

D1

Output data format
0—MSB first
1—LSB first (not supported by driver)

D0

Mode of operation
0—Unipolar (normally used as default)
1—Bipolar

TLC1543 commands (the TLC1543 is a 10-bit A/D converter)

D7–D4

Channel 0 - 10
Channel 11 = (Vref+ - Vref-)/2
Channel 12 = Vref-
Channel 13 = Vref+
(No software power-down mode available)

D3–D0

No specific values assigned.

PARAMETERS

`cmd` is the A/D converter input channel (0–10) to read.

`len` is the output data length:

0 = 12-bit mode (BL2000/BL2020 only)
1 = 8-bit mode (BL2000/BL2020 only)
2 = 10-bit mode (BL2010/BL2030 only)

RETURN VALUE

A value corresponding to the voltage on the A/D converter input channel, which will be:

0–4095 for 12-bit A/D conversions
0–1023 for 10-bit A/D conversions
0–255 for 8-bit A/D conversions

SEE ALSO

`anaIn, anaInVolts, brdInit, samples/bl2000/adc/AD3.C`

`int anaIn(unsigned int channel);`

Reads the state of an A/D converter input channel.

PARAMETER

`channel` is the A/D converter input channel (0–10) to read.

RETURN VALUE

A value corresponding to the voltage on the analog input channel, which will be:

0–4095 for 12-bit A/D conversions (BL2000,BL2020)
0–1023 for 10-bit A/D conversions (BL2010,BL2030).

SEE ALSO

`anaInVolts, anaInCalib, _anaIn, brdInit`

float anaInVolts(unsigned int channel);

Reads the state of an A/D converter input channel and uses the previously set calibration constants to convert it to volts.

PARAMETER

channel is the A/D converter input channel (0–10).

RETURN VALUE

A voltage value corresponding to the voltage on the analog input channel.

SEE ALSO

anaIn, anaInCalib, brdInit

int anaInEERd(unsigned int channel);

Reads the calibration constants, gain, and offset from the simulated EEPROM in flash memory (located in reserved user block memory area 0x1C00–0x1FFF).

PARAMETER

channel is the A/D converter input channel (0–10).

RETURN VALUE

0 if successful.

-1 if address or range is invalid.

SEE ALSO

anaInEEWr, brdInit

int anaInEEWr(unsigned int channel);

Writes the calibration constants, gain, and offset to the simulated EEPROM in flash memory (located in reserved user block memory area 0x1C00–0x1FFF).

PARAMETER

channel is the A/D converter input channel (0–10).

RETURN VALUE

0 if successful.

-1 if address or range is invalid.

SEE ALSO

anaInEERd, brdInit

The address range of the user block memory area where the calibration constants are stored may be determined using

```
readUserBlock( _adcCalib, ADC_CALIB_ADDR, sizeof( _adcCalib ) );
```

where **_adcCalib** and **ADC_CALIB_ADDR** are macros defined in the **BL20XX.LIB** library.

4.4.6 D/A Converter Outputs

The functions in this section apply only to the BL2000 and the BL2020 models.

```
int anaOutCalib(int channel, int value1,
   float volts1, int value2, float volts2);
```

Calibrates the response of the D/A converter channel desired as a linear function using the two conversion points provided. Gain and offset constants are calculated and placed into global table **`_dacCalib`**.

PARAMETERS

`channel` is the D/A converter output channel (0 or 1).

`value1` is the first D/A converter value.

`volts1` is the voltage corresponding to the first D/A converter value.

`value2` is the second D/A converter value.

`volts2` is the voltage corresponding to the second D/A converter value.

RETURN VALUE

0 if sucessful.

-1 if not able to make calibration constants.

SEE ALSO

`anaOut, anaOutVolts, brdInit`

void anaOut(unsigned int channel, unsigned int modecount);

Sets the voltage of a D/A converter output channel by serially clocking in 16 bits to a D/A converter using the following format:

D15–D14

Doesn't matter.

D13–D12

Mode of operation
00—Normal Operation
01—Software Powerdown, 1 kΩ to GND
10—Software Powerdown, 100 kΩ to GND
11—Software Powerdown, three-state

D11–D0

Data bits, MSB–LSB (0–4095)

PARAMETERS

`channel` is the D/A converter output channel to write (0 or 1).

`modecount` is a value corresponding to the voltage on the D/A converter output and/or setting the mode of operation

:

Operation Mode	Description	`modecount` Value
0	Normal Mode	0–4095
1	Software Powerdown, 1 kΩ to GND	0x1000
2	Software Powerdown, 100 kΩ to GND	0x2000
3	Software Powerdown, three-state	0x3000

RETURN VALUE

None

SEE ALSO

`anaOutVolts, anaOutCalib, brdInit`

void anaOutVolts(unsigned int channel, float voltage);

Sets the voltage of a D/A converter output channel by using the previously set calibration constants to calculate the correct data values.

PARAMETERS

channel is the D/A converter output channel (0 or 1).

voltage is the voltage desired on the output channel.

SEE ALSO

anaOut, anaOutCalib, brdInit

int anaOutEERd(unsigned int channel);

Reads the calibration constants, gain, and offset from the simulated EEPROM in flash memory (located in reserved user block memory area 0x1C00–0x1FFF).

PARAMETER

channel is the D/A converter output channel (0 or 1).

RETURN VALUE

0 if successful.

-1 if address or range is invalid.

SEE ALSO

anaOutEEWr, brdInit

int anaOutEEWr(unsigned int channel);

Writes the calibration constants, gain, and offset to the simulated EEPROM in flash memory (located in reserved user block memory area 0x1C00–0x1FFF).

PARAMETER

channel is the D/A converter output channel (0 or 1).

RETURN VALUE

0 if successful.

-1 if address or range is invalid.

SEE ALSO

anaOutEERd, brdInit

The address range of the user block memory area where the calibration constants are stored may be determined using

```
readUserBlock( _dacCalib, DAC_CALIB_ADDR, sizeof( _dacCalib ) );
```

where **_dacCalib** and **DAC_CALIB_ADDR** are macros defined in the **BL20XX.LIB** library.

5. USING THE TCP/IP FEATURES

Chapter 5 provides an introduction to using the TCP/IP features on your BL2000 board.

5.1 TCP/IP Connections

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

If you do not have access to an Ethernet network, use a crossover Ethernet cable to connect the BL2000 to a PC that at least has a 10Base-T Ethernet card.

If you have Ethernet access, use a straight through Ethernet cable to establish an Ethernet connection to the BL2000 from an Ethernet hub. These connections are shown in Figure 21.

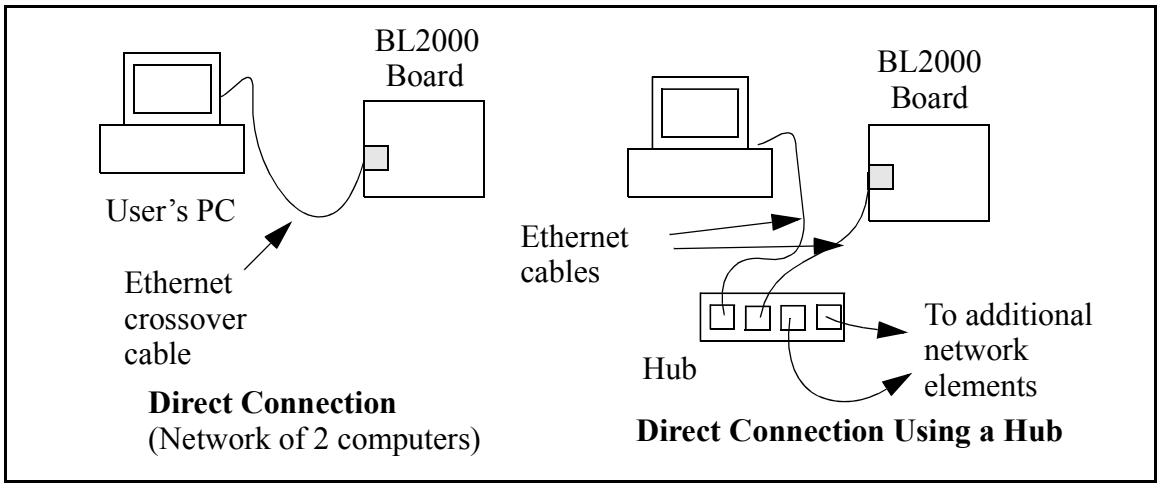


Figure 21. Ethernet Connections

The PC running Dynamic C through the serial programming port on the BL2000 does not need to be the PC with the Ethernet card.

3. Apply Power

Plug in the AC adapter. The BL2000 is now ready to be used.

> **NOTE:** A hardware RESET is accomplished by unplugging the AC adapter, then plugging it back in, or by momentarily grounding the board reset input at pin 9 on screw terminal header J2.

When working with the BL2000, the green **LNK** light is on when a program is running and the board is properly connected either to an Ethernet hub or to an active Ethernet card. The orange **ACT** light flashes each time a packet is received.

5.2 TCP/IP 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 BL2000 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.

5.2.1 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 BL2000 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 BL2000:

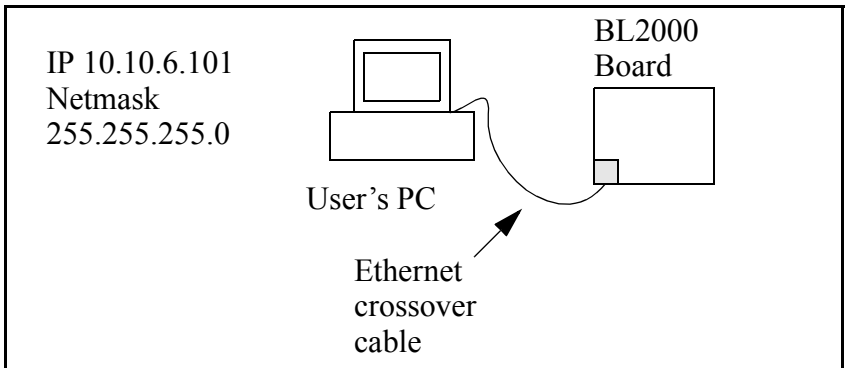
```
#define MY_IP_ADDRESS "10.1.1.2"
#define MY_NETMASK "255.255.255.0"
// #define MY_GATEWAY "10.10.6.1"
// #define MY_NAMESERVER "10.10.6.1"
```

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 you PC can be defined as **10.1.1.1**.

5.2.2 How to Set Up your Computer's IP Address for a Direct Connection

When your computer is connected directly to the BL2000 via an Ethernet connection, you need to assign an IP address to your computer. To assign the PC the address `10.10.6.101` with the netmask `255.255.255.0`, do the following.

Click on **Start > Settings > Control Panel** to bring up the Control Panel, and then double-click the Network icon. Depending on which version of Windows you are using, look for the **TCP/IP Protocol**/**Network > Dial-Up Connections**/**Network** line or tab. Double-click on this line or select **Properties** or **Local Area Connection > Properties** to bring up the TCP/IP properties dialog box. You can edit the IP address and the subnet mask directly. (Disable "obtain an IP address automatically.") You may want to write down the existing values in case you have to restore them later. It is not necessary to edit the gateway address since the gateway is not used with direct connect.



Direct Connection PC to BL2000 Board

5.3 Run the `PINGME.C` Sample Program

Connect the crossover cable from your computer's Ethernet port to the BL2000'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 BL2000 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 orange **ACT** light flashes on the BL2000 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.

5.4 Running More Sample Programs With a Direct Connection

The program **SSI.C** (**SAMPLES\BL2000\TCPIP**) demonstrates how to make the BL2000 a Web server. This program allows you to turn the LEDs on an attached Demonstration Board from the Tool Kit on and off from a remote Web browser. LEDs DS4–DS8 on the BL2000 will match those on the Web page. As long as you have not modified the **TCPCONFIG 1** macro in the sample program, enter the following server address in your Web browser to bring up the Web page served by the sample program.

http://10.10.6.100

Otherwise use the TCP/IP settings you entered in the **TCP_CONFIG.LIB** library.

The sample program **SMTP.C** (**SAMPLES\BL2000\TCPIP**) allows you to send an E-mail when a switch on the Demonstration Board is pressed. Follow the instructions included with the sample program.

The sample program **TELNET.C** (**SAMPLES\BL2000\TCPIP**) allows you to communicate with the BL2000 using the Telnet protocol. This program takes anything that comes in on a port and sends it out Serial Port B. It uses digital input IN0 to indicate that the TCP/IP connection should be closed, and it uses high-current output OUT0 to indicate that there is an open connection. You may change the digital input and output to suit your application needs.

Run the Telnet program on your PC (**Start > Run** `telnet 10.10.6.100`). As long as you have not modified the **TCPCONFIG 1** macro in the sample program, the IP address is 10.10.6.100 as shown; otherwise use the TCP/IP settings you entered in the **TCP_CONFIG.LIB** library. Each character you type will be printed in Dynamic C's **STDIO** window, indicating that the board is receiving the characters typed via TCP/IP.

5.5 Where Do I Go From Here?

NOTE: If you purchased your BL2000 through a distributor or Rabbit 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 Technical Bulletin Board and forums at www.rabbit.com/support/bb/ and at www.rabbit.com/forums/.
- 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.

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***.

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 Web site.

APPENDIX A. SPECIFICATIONS

Appendix A provides the specifications for the BL2000 and describes the conformal coating.

A.1 Electrical and Mechanical Specifications

Figure A-1 shows the mechanical dimensions for the BL2000.

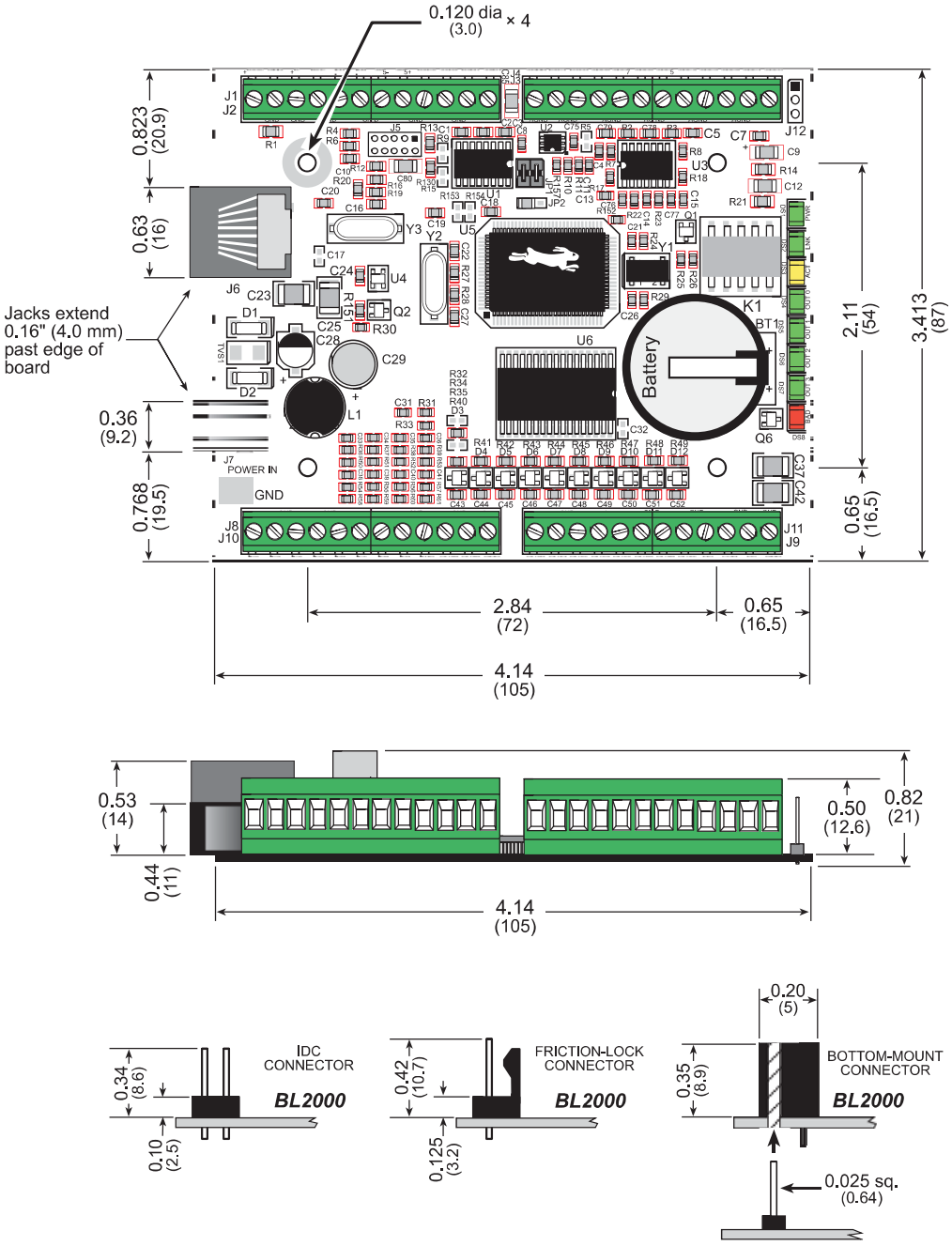


Figure A-1. BL2000 Dimensions

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 BL2000.

Table A-1. BL2000 Specifications

Feature	BL2000	BL2010	BL2020	BL2030
Microprocessor	Rabbit 2000® at 22.1 MHz			
Ethernet Port	10Base-T, **LNK** and **ACT** LEDs		None	
Flash Memory	256K (standard)			
SRAM	128K (standard)			
Backup Battery	Panasonic CR2330 or equivalent 3 V lithium coin type, 265 mA·h standard using onboard battery holder; optional 3 V, 950 mA·h solder-in battery available; external battery connector			
Digital Inputs	11 inputs hardware-configurable pull-up or pull-down, ± 36 V DC, switching threshold 2.4 V typical			
Digital Outputs	10 outputs sinking or sourcing, +40 V DC, 200 mA maximum per channel (2 channels can sink up to 750 mA each)			
Analog Inputs	Four 12-bit res., ± 10 V DC, 1 MΩ, up to 4,000 samples/s	Four 10-bit res., ± 10 V DC, 1 MΩ, up to 4,000 samples/s	Four 12-bit res., ± 10 V DC, 1 MΩ, up to 4,000 samples/s	Four 10-bit res., ± 10 V DC, 1 MΩ, up to 4,000 samples/s
Analog Outputs	Two 12-bit res., 0–4 V DC, update rate 12 kHz	None	Two 12-bit res., 0–4 V DC, update rate 12 kHz	None
Dual-Purpose Analog or Digital Inputs	Five at 12 kΩ, 12-bit res., 0–48 V DC	Seven at 12 kΩ, 10-bit res., 0–48 V DC	Five at 12 kΩ, 12-bit res., 0–48 V DC	Seven at 12 kΩ, 10-bit res., 0–48 V DC
Relay Output	SPDT with snubbers*: 1 A @ 30 V DC, 300 mA @ 120 V AC (uses one digital output) max. contact settling time 4 ms			
Serial Ports	4 serial ports: • two RS-232 or one RS-232 (with CTS/RTS) • one RS-485, onboard network termination and bias resistors • one 5 V CMOS-compatible programming port			
Serial Rate	Max. burst rate = CLK/32 Max. sustained rate = CLK/64			
Connectors	one RJ-45 (Ethernet) one 2 × 5, 2 mm pitch (serial programming port) one power jack for AC adapter four 12-terminal screw connectors (14 AWG/1.5 mm^2 wire) for analog and digital I/O, relay			
Real-Time Clock	Yes			

Table A-1. BL2000 Specifications (continued)

Feature	BL2000	BL2010	BL2020	BL2030
Timers	Five 8-bit timers (four are cascadable from the first) and one 10-bit timer with two match registers			
Watchdog/Supervisor	Yes			
Power	9–40 V DC or 24 V AC (±10%), 1.5 W max.			
Operating Temperature	–40°C to +70°C			
Humidity	5–95%, noncondensing			
Board Size	3.43" × 4.15" × 0.82" (87 mm × 105 mm × 21 mm)			

* When using the BL2000 in a CE-certified application, the voltages handled by the relay must not exceed SELV levels (42.4 V AC peak, or 60 V DC).

A.1.1 Headers

The BL2000 has an option for 0.1" IDC headers or friction-lock connectors at J1, J3, J10, and J11 for physical connection to other boards or ribbon cables.

Figure A-2 shows the BL2000 footprint. These values are relative to one of the mounting holes.

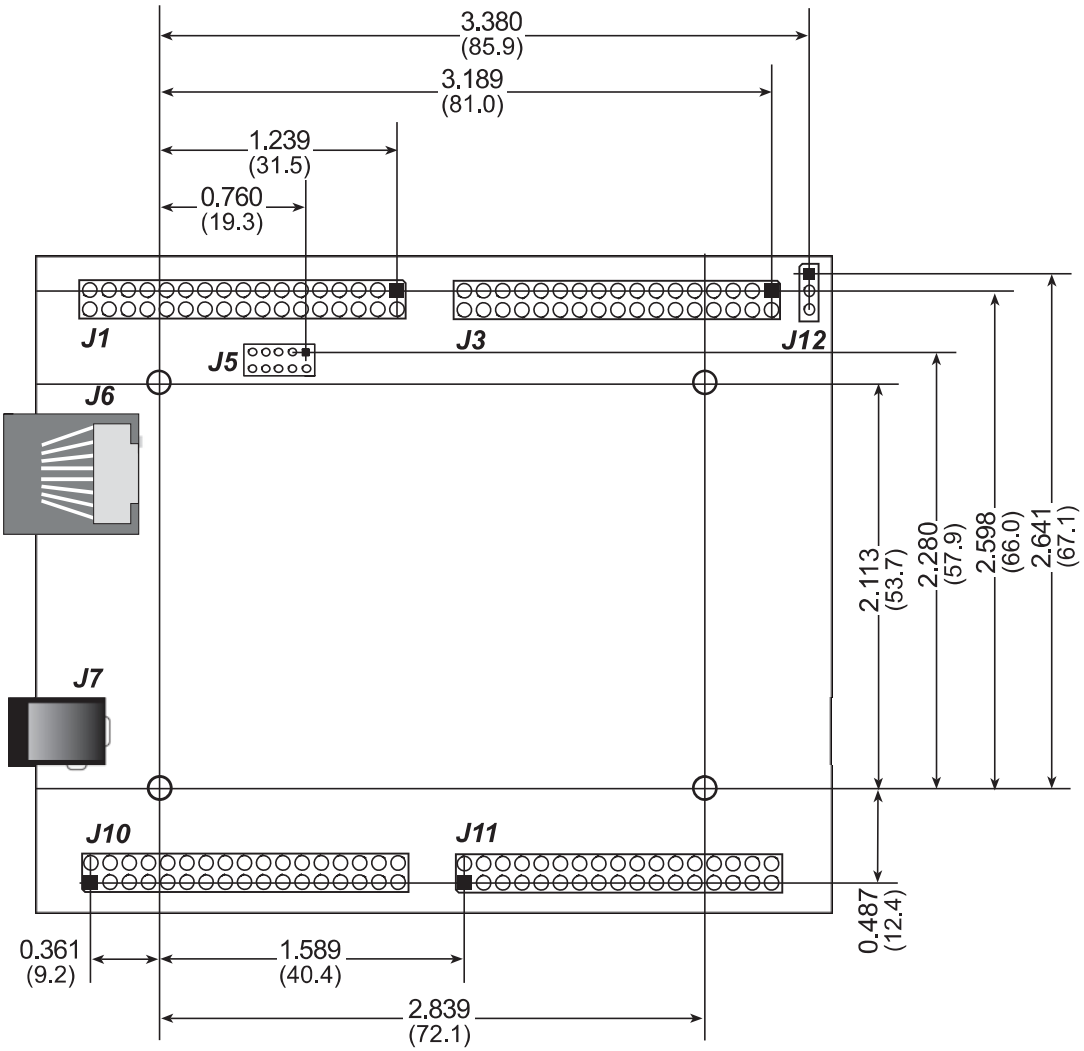


Figure A-2. User Board Footprint for BL2000

NOTE: The same footprint applies for the IDC header and bottom-mount socket options.

A.2 Conformal Coating

The areas around the crystal oscillator and the battery backup circuit on the BL2000 have had the Dow Corning silicone-based 1-2620 conformal coating applied. The conformally coated areas are shown in Figure A-3. The conformal coating protects these high-impedance circuits from the effects of moisture and contaminants over time.

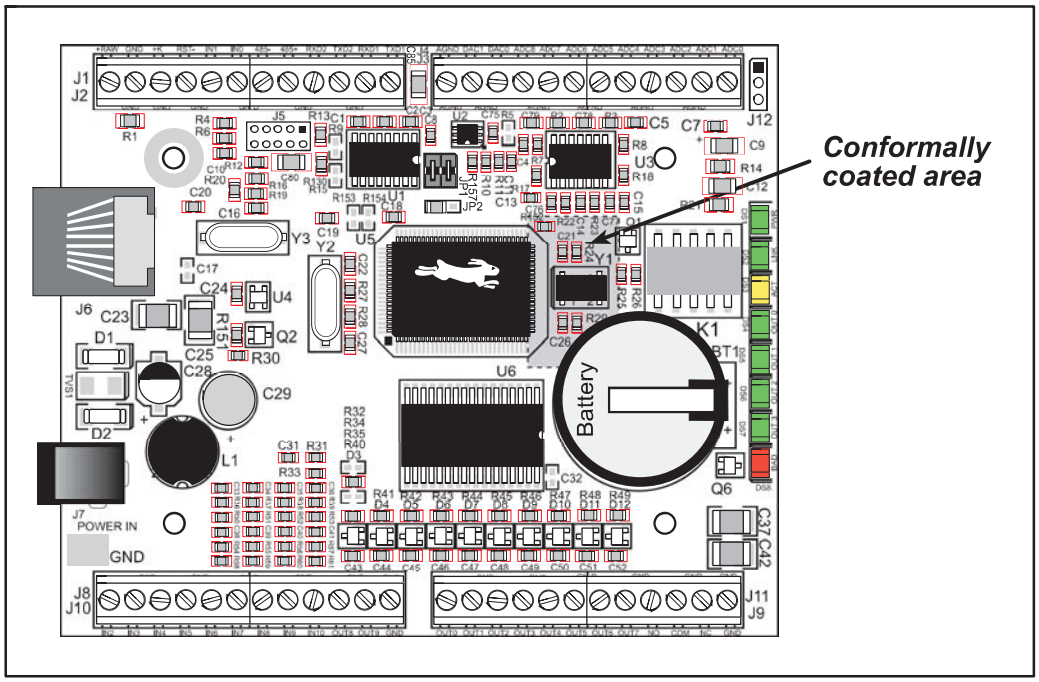


Figure A-3. BL2000 Areas Receiving Conformal Coating

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

> **NOTE:** For more information on conformal coatings, refer to Rabbit Technical Note 303, ***Conformal Coatings***.

A.3 Jumper Configurations

Figure A-4 shows the header locations used to configure the various BL2000 options via jumpers.

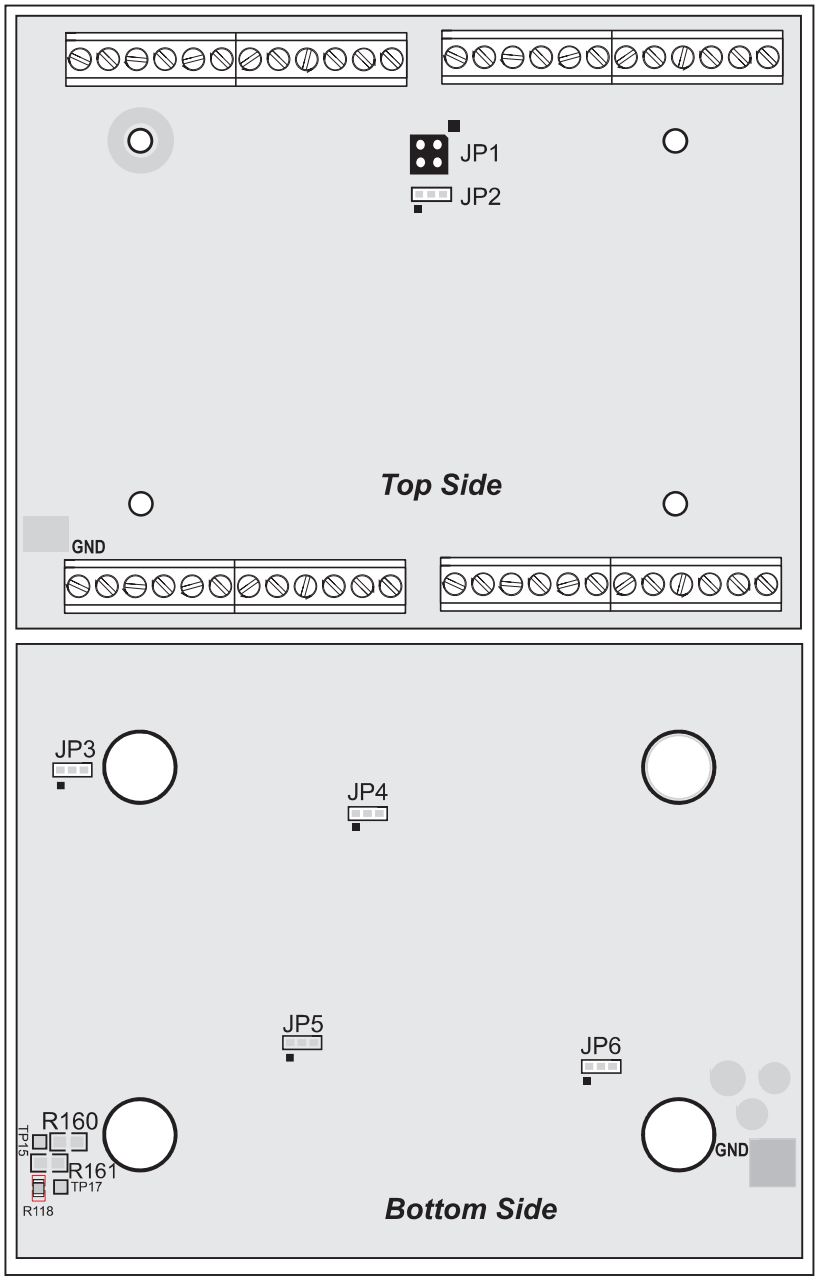


Figure A-4. Location of BL2000 Configurable Positions

Table A-2 lists the configuration options.

Table A-2. BL2000 Jumper Configurations

Header	Description	Pins Connected		Factory Default
J9	Pin 12 is Vcc	R160 installed		
	Pin 12 is GND	R161 installed		×
JP1	RS-485 Bias and Termination Resistors	1–2 3–4	Bias and termination resistors connected	×
		None	Bias and termination resistors ***not*** connected	
JP2	Flash Memory Bank Select	1–2	Normal Mode	×
		2–3	Bank Mode	
JP3	D/A Converter Power Supply	1–2	+V	×
		2–3	+REF	
JP4	Flash Memory Size	1–2	128K/256K	×
		2–3	512K	
JP5	SRAM Size	1–2	128K	×
		2–3	512K	
JP6	Digital Input Pull-Up/Pull-Down Resistors	1–2	Pulled up	×
		2–3	Pulled down	

NOTE: Only header JP1 uses actual jumpers. The other connections are made using 0 Ω surface-mounted resistors.

A.4 Use of Rabbit 2000 Parallel Ports

Figure A-5 shows the Rabbit-based subsystems designed into the BL2000.

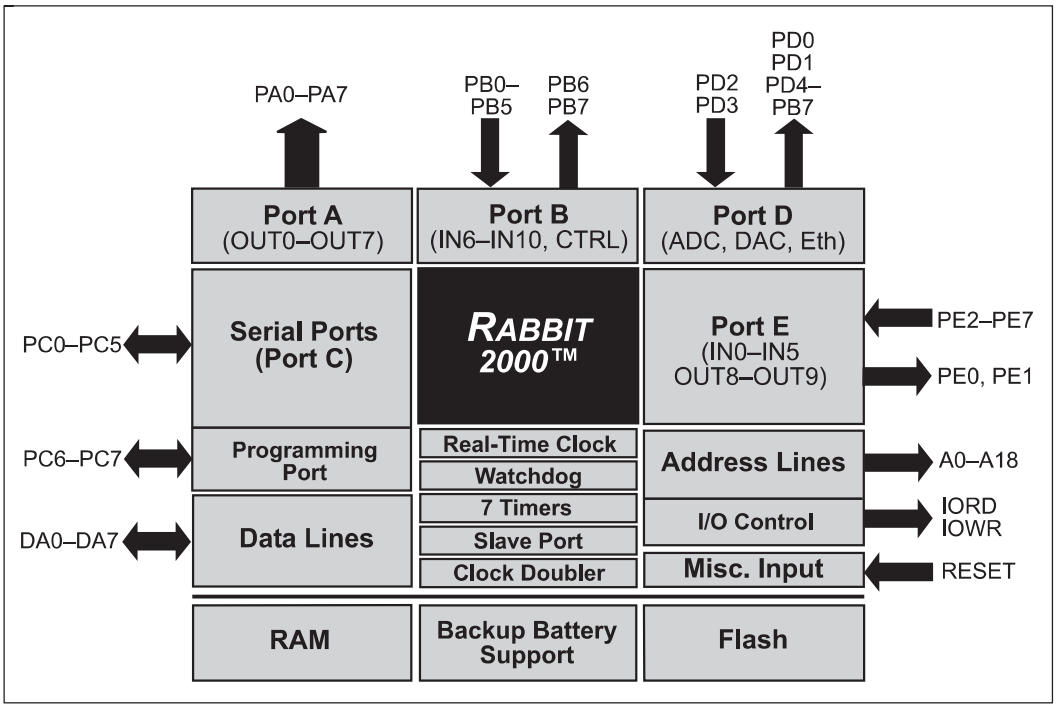


Figure A-5. BL2000 Rabbit-Based Subsystems

Table A-3 lists the Rabbit 2000 parallel ports and their use in the BL2000.

Table A-3. Use of Rabbit 2000 Parallel Ports

Port	I/O	Signal	Output Function State
PA0	Output	OUT0/RELAY/LED_DS4	Off
PA1	Output	OUT1/LED_DS5	Off
PA2	Output	OUT2/LED_DS6	Off
PA3	Output	OUT3/LED_DS7	Off
PA4	Output	OUT4	Off
PA5	Output	OUT5	Off
PA6	Output	OUT6	Off
PA7	Output	OUT7	Off
PB0	Input	IN6	N/A
PB1	Input	CLKA	N/A
PB2	Input	IN7	N/A

Table A-3. Use of Rabbit 2000 Parallel Ports (continued)

Port	I/O	Signal		Output Function State
PB3	Input	IN8		N/A
PB4	Input	IN9		N/A
PB5	Input	IN10		N/A
PB6	Output	RS485_EN		Off
PB7	Output	UPGOOD		Off
PC0	Output	TXD RS-485	Serial Port D	Inactive high
PC1	Input	RXD RS-485		N/A
PC2	Output	RTS/TXC RS-232	Serial Port C	Inactive high
PC3	Input	CTS/RXC RS-232		N/A
PC4	Output	TXB RS-232	Serial Port B	Inactive high
PC5	Input	RXB RS-232		N/A
PC6	Output	TXA Programming Port	Serial Port A	Inactive high
PC7	Input	RXA Programming Port		N/A
PD0	Output	DAC-ADC_SK		On
PD1	Output	DAC-ADC_SDI		On
PD2	Input	RTL-ADC_SDO		N/A
PD3	Input	RTL_SK		N/A*
PD4	Output	RTL_SDI		On
PD5	Output	/DAC0_CS		Inactive high
PD6	Output	/DAC1_CS		Inactive high
PD7	Output	/ADC_CS		Inactive high
PE0	Output	OUT8		Off
PE1	Output	OUT9		Off
PE2	Input	IN0		N/A
PE3	Input	IN1		N/A
PE4	Input	IN2		N/A
PE5	Input	IN3		N/A
PE6	Input	IN4		N/A
PE7	Input	IN5		N/A

* PD3 is an output (and is on) for the BL2020 and the BL2030.

APPENDIX B. PLASTIC ENCLOSURE

The plastic enclosure provides a secure way to protect your BL2000. The enclosure itself may be mounted on any flat surface.

Appendix B describes how to mount the BL2000 inside the plastic enclosure, how to install the optional light pipes, and provides details on mounting the assembly.

B.1 Assembly

1. Attach the BL2000 to the plastic enclosure base.

Position the BL2000 over the plastic enclosure base as shown below in Figure B-1. Attach the BL2000 to the base using the two 4-40 × ¼ screws supplied.

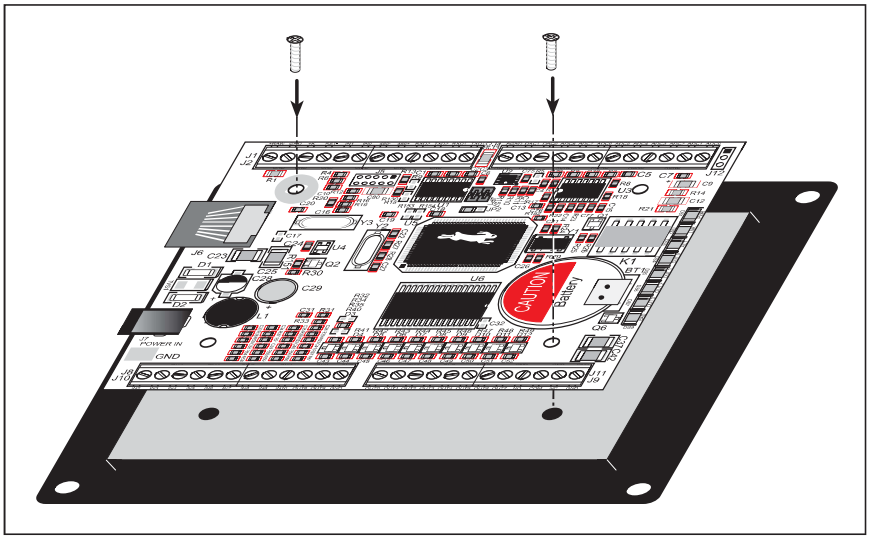


Figure B-1. Attach BL2000 to Plastic Enclosure Base

2. Install light pipes (optional).

Light pipes are included in the Tool Kit to facilitate seeing the LEDs on the BL2000 board once the enclosure is assembled.

With the enclosure top positioned as shown in Figure B-2, insert the eight light pipes into the slots identified in Figure B-2. Position the light pipes snugly against the enclosure top since there is little clearance between the light pipes and the LEDs on the BL2000. The light pipes "snap" in place. Verify that the light pipes are aligned over the LEDs, then apply a drop of cyanoacrylate or contact cement to the inside of the enclosure around each light pipe to hold it in place.

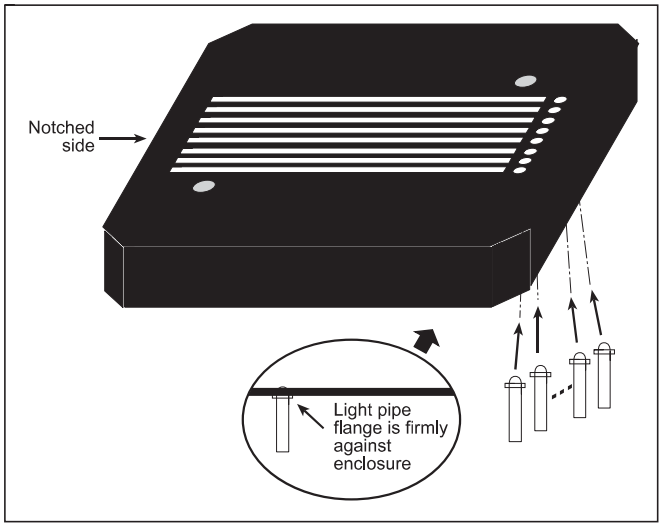


Figure B-2. Install Light Pipes in Enclosure Top

NOTE: Once the glue is applied, it will not be possible to change the alignment of the light pipes without damaging the plastic enclosure.

3. Attach the enclosure top to the base.

Position the enclosure top over the plastic enclosure base as shown below in Figure B-3. Attach the enclosure top to the base using the two 4-40 × ½ screws supplied. If you installed the light pipes, be sure they are aligned over the LEDs as shown.

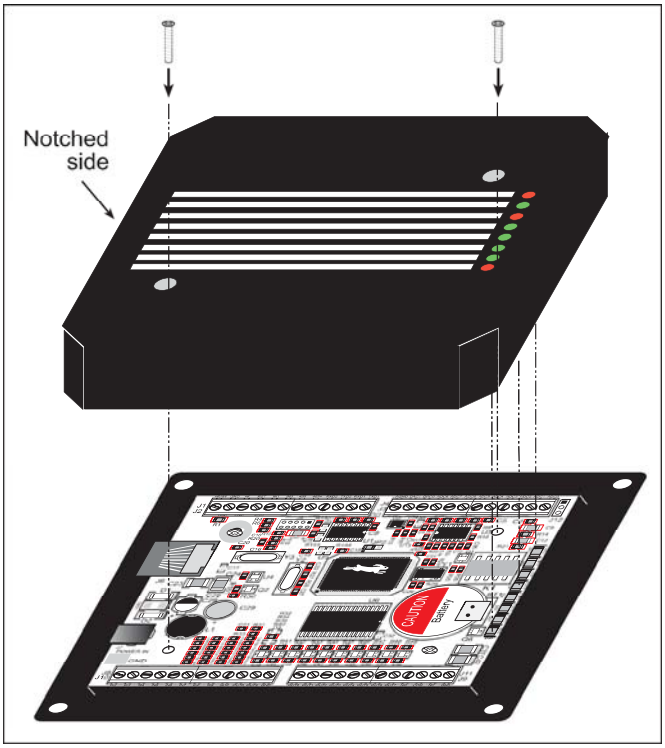


Figure B-3. Attach Enclosure Top

4. Mount plastic enclosure (optional).

Use four #10 screws to attach the assembled plastic enclosure to the surface on which it will be mounted. This step applies to production versions of BL2000 boards once development has been completed.

B.2 Dimensions

Figure B-4 shows the dimensions for the plastic enclosure.

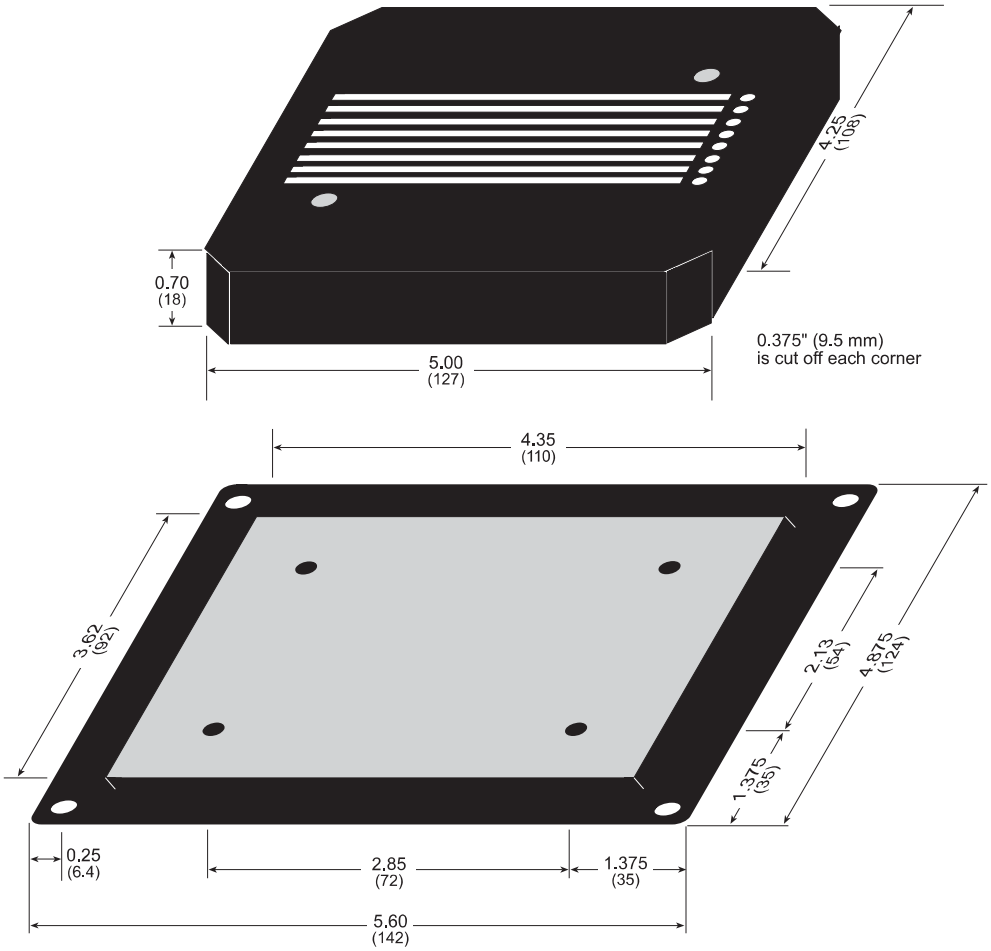


Figure B-4. Plastic Enclosure Dimensions

When fully assembled with the BL2000 installed, the total height of the plastic enclosure will be 1.1" (28 mm).

APPENDIX C. POWER SUPPLY

Appendix C describes the power circuitry distributed on the BL2000.

C.1 Power Supplies

Power is supplied to the BL2000 via a mini phone jack located at J7 or through the screw terminal strip, header J2. The BL2000 itself is protected against reverse polarity by a diode at D1 as shown in Figure C-1.

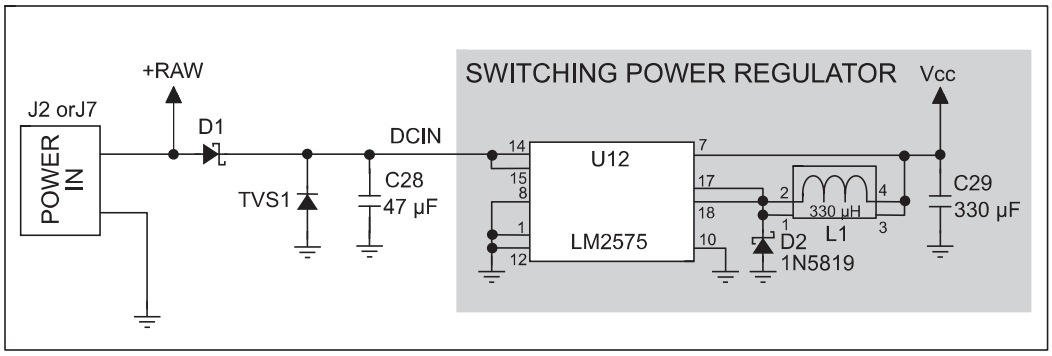


Figure C-1. BL2000 Power Supply

Capacitor C28 provides surge current protection for the voltage regulator, and allows the external power supply to be located some distance away from the BL2000. A switching power regulator is used. The input voltage range is from 9 V to 40 V.

The BL2000 can alternatively be powered by 24 V AC. In this case D1 and C28 act as a half-wave rectifier to produce approximately 40 V DC at the input of the switching regulator, U12. Although a significant drop will be measured at DCIN, the voltage will never drop below +9 V DC. As long as the minimum input level is maintained at the input to the regulator, Vcc will be held at +5 V DC.

Pin 12 on header J9 can be configured to either supply Vcc (0 Ω surface-mounted resistor installed at R160) or GND (0 Ω surface-mounted resistor installed at R161). When using pin 12 on header J9 to supply Vcc, take care not to draw more than 25 mA current from this pin, especially if you are using 24 V AC as your +RAW input power supply. The R160 and R161 locations are shown in Figure A-4.

C.1.1 Power for Analog Circuits

Power to the analog circuits is provided by way of a two-stage low-pass filter, which isolates the analog section from digital noise generated by the other components. The analog power voltage +V powers the op-amp for the buffered A/D converter inputs, the A/D converter, and the 4.096 V reference circuit. The two D/A converters can be powered either from the reference, which is the standard, or from +V when ratiometric measurements are desired. The maximum current draw on +V is less than 10 mA.

There are three digital grounds, one on each of the screw-terminal headers associated with the digital functions (J2, J8, and J9). The digital ground and the analog ground share a single split ground plane on the board, with the analog ground connected at a single point to the digital ground by a 0 Ω resistor (R87). This is done to minimize digital noise in the analog circuits and to eliminate the possibility of ground loops. External connections to analog ground are made on screw-terminal header J4.

C.2 Batteries and External Battery Connections

The SRAM and the real-time clock have battery backup. Power to the SRAM and the real-time clock (VRAM) is provided by two different sources, depending on whether the main part of the BL2000 is powered or not. When the BL2000 is powered normally, and Vcc is within operating limits, the SRAM and the real-time clock are powered from Vcc. If power to the board is lost or falls below 4.63 V, the VRAM power will come from the battery. The reset generator circuit controls the source of power by way of its **/RESET** output signal.

A replaceable 265 mA·h lithium battery provides power to the real-time clock and SRAM when external power is removed from the circuit board. The drain on the battery is typically less than 10 µA when there is no external power applied, and so the expected in-service life of the battery is

$$\frac{265\ \text{mA}\cdot\text{h}}{10\ \mu\text{A}} = 3.0\ \text{years.}$$

The drain on the battery is typically less than 4 µA when external power ***is*** applied, and so the expected BL2000 shelf life is

$$\frac{265\ \text{mA}\cdot\text{h}}{4\ \mu\text{A}} = 7.5\ \text{years.}$$

A long-life 950 mA·h solder-in battery is also provided for in the board layout. Alternatively, an external battery may be connected to the BL2000 via header J12. The existing battery does not have to be removed when an external battery is used.

C.2.1 Replacing the Backup Battery

The battery is user-replaceable, and is fitted in a battery holder. To replace the battery, lift up on the spring clip and slide out the old battery. Use only a Panasonic CR2330 or equivalent replacement battery, and insert it into the battery holder with the + side facing up.

NOTE: The SRAM contents and the real-time clock settings will be lost if the battery is replaced with no power applied to the BL2000. Exercise care if you replace the battery while external power is applied to the BL2000.

CAUTION: There is an explosion danger if the battery is short-circuited, recharged, or replaced incorrectly. Replace the battery only with the same type or an equivalent type recommended by the battery manufacturer. Dispose of used batteries according to the battery manufacturer's instructions.

C.2.2 Battery-Backup Circuit

Figure C-2 shows the battery-backup circuit.

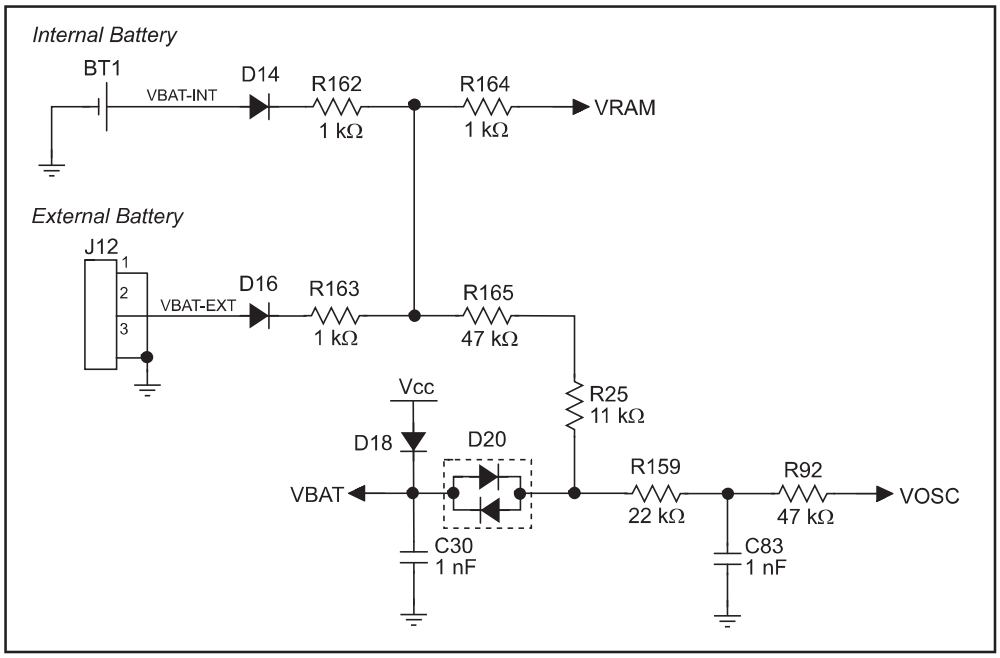


Figure C-2. BL2000 Backup Battery Circuit

The battery-backup circuit serves three purposes:

- It reduces the battery voltage to the SRAM and to the real-time clock, thereby limiting the current consumed by the real-time clock and lengthening the battery life.
- It ensures that current can flow only ***out*** of the battery to prevent charging the battery.
- A voltage, VOSC, is supplied to U14, which keeps the 32.768 kHz oscillator working when the voltage begins to drop.

VRAM and Vcc are nearly equal (<100 mV, typically 10 mV) when power is supplied to the BL2000.

C.2.3 Power to VRAM Switch

The VRAM switch, shown in Figure C-3, allows the battery backup to provide power when the external power goes off. The switch provides an isolation between Vcc and the battery when Vcc goes low. This prevents the Vcc line from draining the battery.

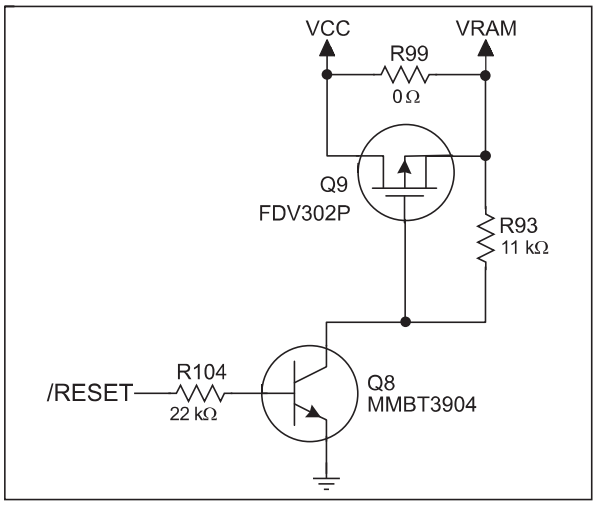


Figure C-3. VRAM Switch

Field-effect transistor Q9 is needed to provide a very small voltage drop between Vcc and VRAM (<100 mV, typically 10 mV) so that the board components powered by Vcc will not have a significantly different voltage than VRAM.

When the BL2000 is ***not*** in reset, the **/RESET** line will be high. This turns on Q8, causing its collector to go low. This turns on Q9, allowing VRAM to nearly equal Vcc.

When the BL2000 ***is*** in reset, the **/RESET** line will go low. This turns off Q8 and Q9, providing an isolation between Vcc and VRAM.

C.2.4 Reset Generator

The BL2000 uses a reset generator, U4, to reset the Rabbit 2000 microprocessor when the voltage drops below the voltage necessary for reliable operation. The reset occurs between 4.50 V and 4.75 V, typically 4.63 V. The reset can be initiated either externally or by a watchdog timeout (**WDTOUT**) on the Rabbit 2000 microprocessor.

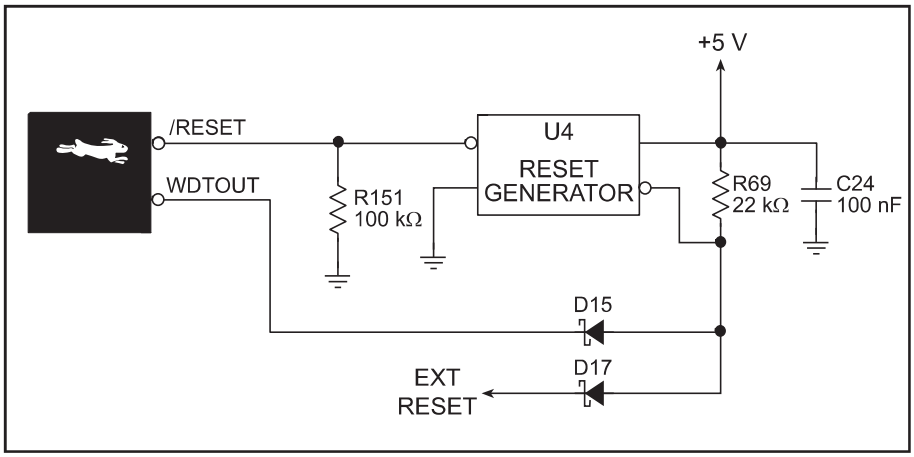


Figure C-4. Reset Generator

NOTE: The Dynamic C function **`chkWDTO`** is not able to detect whether a watchdog timeout has occurred on the BL2000. The GCSR status bits are read and stored by the BIOS, and the reset status bit would normally change once a reset has occurred. However, since **WDTOUT** is tied to the reset generator, a watchdog timeout forces a hardware reset, followed by the BIOS reading and storing the status bits corresponding to power-up or reset.

To use **`chkWDTO`** to detect a watchdog timeout, remove D15 (located on the bottom side of the BL2000).

C.3 Chip Select Circuit

Figure C-5 shows a schematic of the chip select circuit.

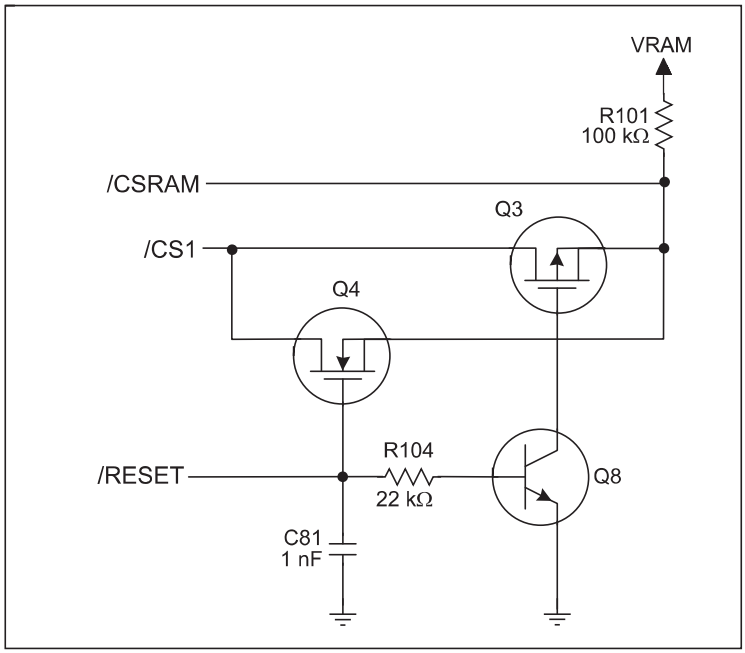


Figure C-5. Chip Select Circuit

The current drain on the battery in a battery-backed circuit must be kept at a minimum. When the BL2000 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 circuit accomplishes this task for the SRAM's chip select signal line.

In a powered-up condition, the CS control circuit 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 battery voltage high). Q3 and Q4 are MOSFET transistors with complementary 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 (R101). 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 voltage can be passed. When the /CS1 voltage is low, Q4 will conduct. When the /CS1 voltage is high, Q3 conducts. It takes time for the transistors to turn on, creating a propagation delay. This propagation delay is typically very small, about 10 ns to 15 ns.

The signal that turns the transistors on is a high on the processor's reset line, **/RESET**. When the BL2000 is not in reset, the reset line will be high, turning on n-channel Q4 directly and p-channel FET Q3 by way of Q8. When the board is in reset both Q3 and Q4 are off, isolating /CSRAM and allowing it to be pulled to VRAM.

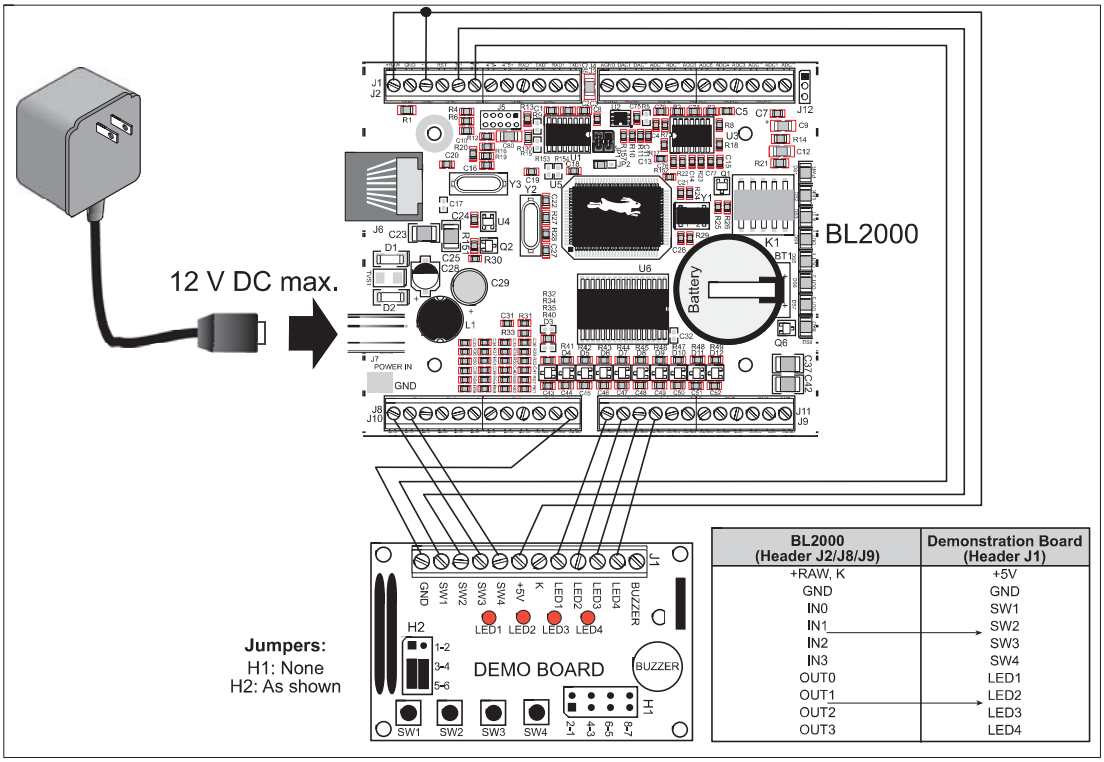
APPENDIX D. DEMONSTRATION BOARD

Appendix D shows how to connect the Demonstration Board to the BL2000.

D.1 Connecting Demonstration Board

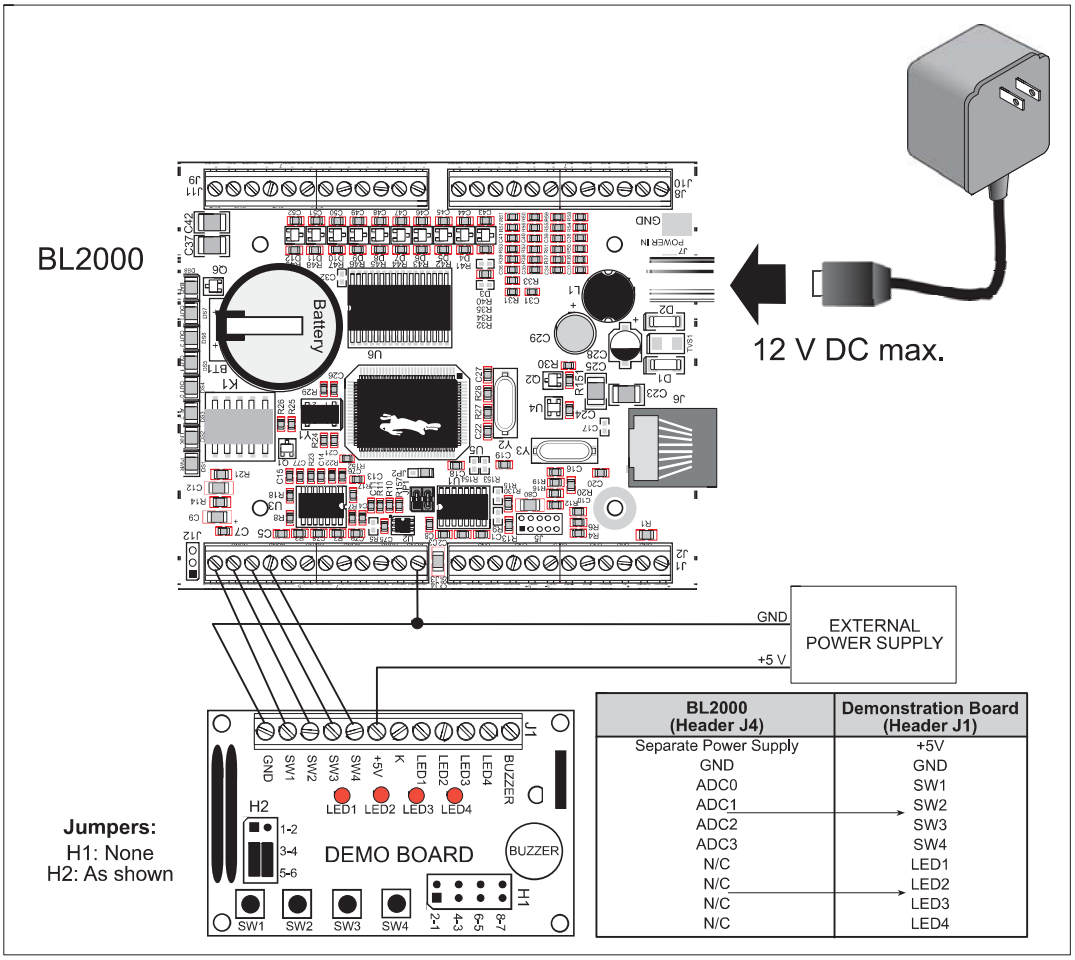
Before running sample programs based on the Demonstration Board, you will have to connect the Demonstration Board from the BL2000 Tool Kit to the BL2000 board. Proceed as follows.

1. Use the wires included in the BL2000 Tool Kit to connect header J1 on the Demonstration Board to header J8 and J9 on the BL2000. The connections are shown in Figure D-1 for sample programs **`DIGIN.C`**, **`DIGOUT.C`**, and **`SMTP.C`**, and in Figure D-2 for sample program **`ANADIGIN.C`**.
2. Make sure that your BL2000 is connected to your PC and that the power supply is connected to the BL2000 and plugged in as described in Chapter 2, “Getting Started.”



BL2000 (Header J2/J8/J9)	Demonstration Board (Header J1)
+RAW, K	+5V
GND	GND
IN0	SW1
IN1	SW2
IN2	SW3
IN3	SW4
OUT0	LED1
OUT1	LED2
OUT2	LED3
OUT3	LED4

Figure D-1. General Digital Connections Between BL2000 and Demonstration Board



BL2000 (Header J4)	Demonstration Board (Header J1)
Separate Power Supply	+5V
GND	GND
ADC0	SW1
ADC1	SW2
ADC2	SW3
ADC3	SW4
N/C	LED1
N/C	LED2
N/C	LED3
N/C	LED4

Figure D-2. Analog Connections Between BL2000 and Demonstration Board

INDEX

A

A/D converter
  function calls
    _anaIn 47
    anaIn 48
    anaInCalib 47
    anaInEERd 49
    anaInEEWr 49
    anaInVolts 49
analog I/O
  reference voltages 28
analog inputs 27
  use as digital inputs 27
analog outputs 28

B

battery
  connections 74
  life 74
  replacing the backup battery 75
board initialization
  function calls 42
    brdInit 42

C

C language 35
CE compliance 5
  design guidelines 6
chip select circuit 78
clock doubler 32
conformal coating 66
connections
  battery 74
  Ethernet cable 53

D

D/A converter
  function calls
    anaOut 51
    anaOutCalib 50
    anaOutEERd 52
    anaOutEEWr 52
    anaOutVolts 52
D/A converter outputs 28
Demonstration Board 3
  hookup instructions 81
    analog sample programs 83
    digital sample programs 82
  jumper configurations . 82, 83
  wire assembly 3
digital I/O
  function calls
    digIn 44
    digOut 44
  SMODE0 26
  SMODE1 26
digital inputs 17
  configured pull-up or pull-down 17
  dual-purpose A/D converter inputs as digital inputs .. 18
  switching threshold 18
digital outputs 19
  +K 19
  pulled up or pulled down .. 19
  sinking 19
  sourcing 19
dimensions
  BL2000 board 60
  plastic enclosure 72
Dynamic C 4, 35
  add-on modules 37
  COM port 11
  installation 10
  Rabbit Embedded Security Pack 4
  standard features
    debugging 36
  starting 11
  telephone-based technical support 4, 37
  upgrades and patches 37

E

EMI
  spectrum spreader feature . 33
Ethernet cables 53
Ethernet connections 53
  steps 53
Ethernet port 24
  ground lug 25
  handling EMI and noise 25
  pinout 24
external interrupts 32

F

features 1

H

headers
  Demonstration Board
    H1 ..... 82, 83
    H2 ..... 82, 83

I

installation
  plastic enclosure
    BL2000 70
    light pipes 70
    top 71

IP addresses 56
how to set 55
how to set PC IP address ...56

J

jumper configurations 65, 66
Demonstration Board ..82, 83
J9(12) 73
J9(12) is GND 66
J9(12) is Vcc 66
JP1 (RS-485 bias and termination resistors) 24, 66
JP2 (flash memory bank select) 30, 66
JP3 (D/A converter power supply) 28, 66
JP4 (flash memory size) 30, 66
JP5 (SRAM size) 30, 66
JP6 (digital input pull-up/pull-down resistors) 17, 66
jumper locations 65

K

K 19

L

LED outputs
function calls
ledOut 46

M

memory 30
flash memory configurations 30
SRAM configuration for different sizes 30
models 2
BL2000 2
BL2010 2
BL2020 2
BL2030 2

P

pinout
BL2000 headers 14, 15
Ethernet port 24
plastic enclosure 3, 70
attach BL2000 to base 7
dimensions 72
power management 73
power supplies 73
backup-battery circuit 75
battery backup 74
chip select circuit 78
switching voltage regulator 73
VRAM switch 76
power supply
connections 9, 16
programming
programming cable 3
programming port 25
programming cable 3
connections 8
PROG connector 31

R

Rabbit 2000
parallel ports 67
real-time clock
how to set 40
relay output
function calls
relayOut 46
relay outputs 21
operation in parallel with digital output OUT0 21
reset 9
hardware 9
reset generator 77
RS-232 22
RS-485 22
RS-485 network 23
termination and bias resistors 24

S

sample programs 38
A/D converter inputs
AD_CALIB.C 39
AD1.C 39
AD2.C 39
AD3.C 39
AD4.C 39
BOARD_ID.C 38
COUNTLEDS.C 38
D/A converter outputs
DACAL.C 40
DAOUT1.C 40
DAOUT2.C 40
digital I/O
ANADIGIN.C 38
DIGIN.C 38
DIGOUT.C 38
LED.C 38
PWM.C 39
RELAY.C 39
how to set IP address 55
LEDS_4.C 38
PONG.C 12
real-time clock
RTC_TEST.C 40
SETRTCKB.C 40
serial communication
MASTER.C 39
PUTS.C 39
RELAYCHR.C 39
SLAVE.C 39
TCP/IP 55
PINGME.C 57
SMTP.C 58
SSI.C 58
TELNET.C 58
serial communication 22
flow control 45
function calls
ser485Rx 45
ser485Tx 45
serCflowcontrolOff 45
serCflowcontrolOn 45
programming port 25
RS-232 description 22
RS-485 description 22
RS-485 network 23
RS-485 termination and bias resistors 24
serial ports
Ethernet port 24
setup 7
attach BL2000 to enclosure base 7
power supply connections ...8
programming cable connections 8
software 4
libraries
BL2000 41
PACKET.LIB 45
RS232.LIB 45
TCP/IP 41
sample programs 38

specifications
dimensions
BL2000 60
plastic enclosure 72
electrical 61
header footprint 63
headers 63
relative pin 1 locations 63
temperature 61
spectrum spreader 33
subsystems 13, 14

T

TCP/IP connections 53
10Base-T Ethernet card 53
additional resources 58
Ethernet hub 53
steps 53
technical support 12
Tool Kit 3
AC adapter 3
DC power supply 3
Demonstration Board 3
Dynamic C software 3
plastic enclosure 3
programming cable 3
software 3
User's Manual 3
wire assembly 3

U

USB/serial port converter 8
Dynamic C settings 11

V

Vcc
J9(12) 73

W

watchdog timeout
function calls
chkWDTO 77

SCHEMATICS

090-0117 BL2000 Schematic

www.rabbit.com/documentation/schemat/090-0117.pdf

090-0042 Demonstration Board Schematic

www.rabbit.com/documentation/schemat/090-0042.pdf

090-0128 Programming Cable Schematic

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

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