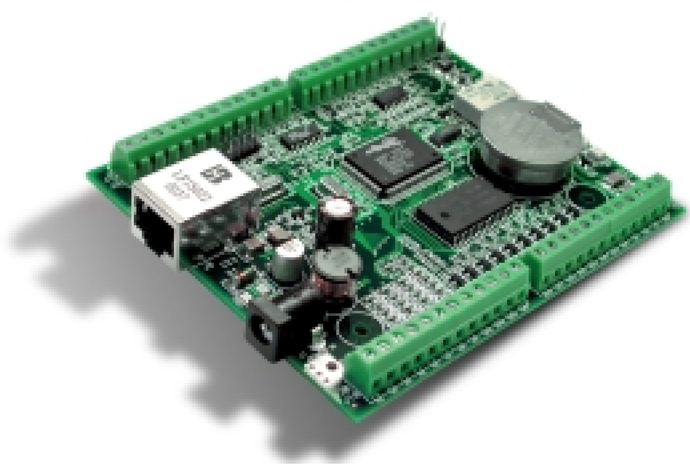


BL2000

C-Programmable Single-Board Computer with Ethernet

User's Manual

019–0094 • 010521–B

BL2000 User's Manual

Part Number 019-0094 • 010521–B • Printed in U.S.A.

Notice to Users

Z-WORLD PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE-SUPPORT DEVICES OR SYSTEMS UNLESS A SPECIFIC WRITTEN AGREEMENT REGARDING SUCH INTENDED USE IS ENTERED INTO BETWEEN THE CUSTOMER AND Z-WORLD PRIOR TO USE. Life-support devices or systems are devices or systems intended for surgical implantation into the body or to sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling and user's manual, can be reasonably expected to result in significant injury.

No complex software or hardware system is perfect. Bugs are always present in a system of any size. In order to prevent danger to life or property, it is the responsibility of the system designer to incorporate redundant protective mechanisms appropriate to the risk involved.

Trademarks

Z-World, Inc.

2900 Spafford Street
Davis, California95616-6800
USA

Telephone:(530)757-3737
Fax:(530)753-5141

www.zworld.com

TABLE OF CONTENTS

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.
- 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.

1.3 Development and Evaluation Tools

1.3.1 Tool Kit

A Tool Kit contains the hardware essentials you will need to create and use your own BL2000 single-board computer.

The items in the Tool Kit and their use are as follows:

- ***BL2000 User's Manual*** with schematics (this document).
- Programming cable, used to connect your PC serial port to the BL2000.
- AC adapter, used to power the BL2000. An AC adapter is supplied with tool kits sold in the North American market. If you are using another power supply, it must provide 9 to 40 V DC.
- Demonstration Board with pushbutton switches and LEDs. The Demonstration Board can be hooked up to the BL2000 to demonstrate the I/O and the TCP/IP capabilities of the BL2000.
- Wire assembly to connect Demonstration Board to BL2000.
- Plastic enclosure with four screws and eight customer-installable light pipes.
- Screwdriver.

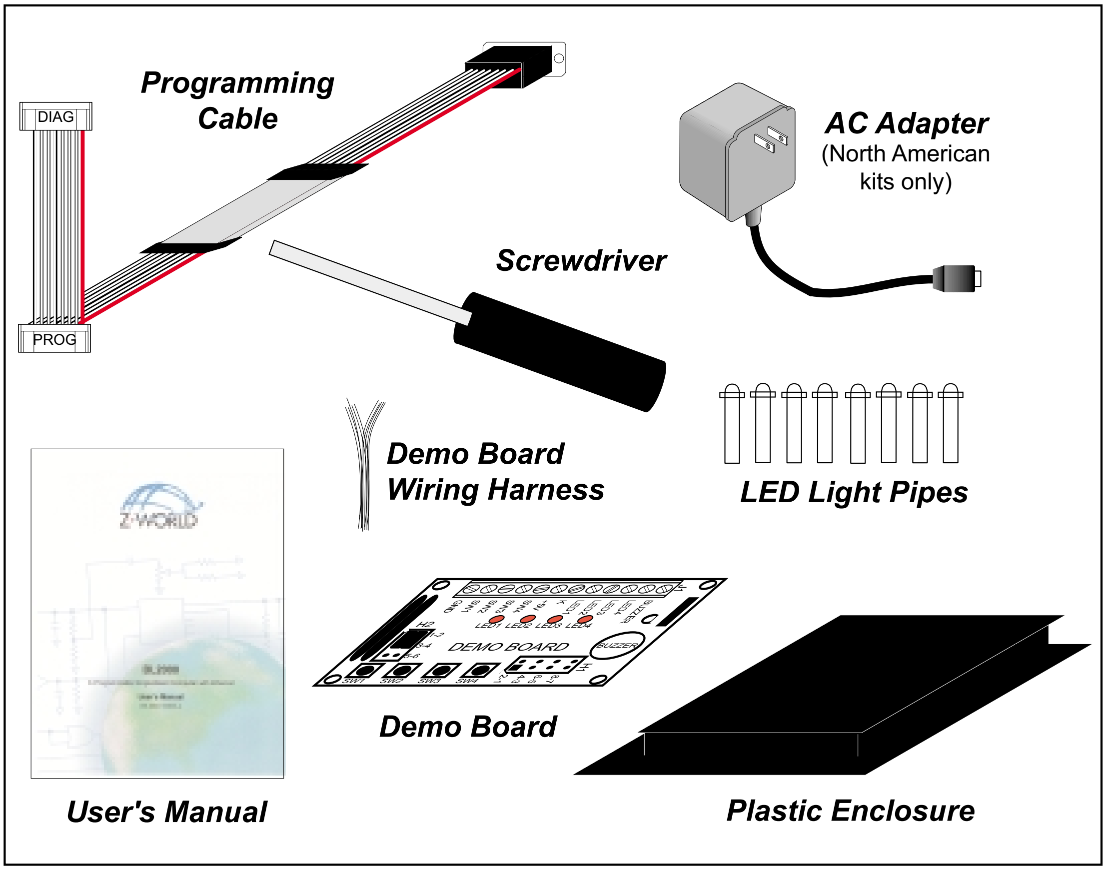


Figure 1. BL2000 Tool Kit

1.3.2 Software

1.3.3 Software

The BL2000 is programmed using version 7.04 or later of Z-World's Dynamic C Premier, an integrated development environment that includes an editor, a C compiler, and a debugger. Library functions provide an easy-to-use interface for the BL2000. Software drivers for TCP/IP, I/O, and serial communication are included with Dynamic C Premier.

The programming cable has a level converter board in the middle of the cable since the BL2000 programming port supports CMOS logic levels, and not the higher voltage RS-232 levels that are used by PC serial ports. When the programming cable is connected, Dynamic C running on the PC can hard-reset the BL2000 and cold-boot it. The cold boot includes compiling and downloading a BIOS program that stays resident while you work. If you crash the target, Dynamic C will automatically reboot and recompile the BIOS if it senses that a target communication error occurred or that the BIOS source code has changed.

You have a choice of doing your software development in the flash memory or in the static RAM included on the BL2000. The advantage of working in RAM is to save wear on the flash memory, which is limited to about 100,000 write cycles.

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.

The disadvantage of using flash memory for debug is that interrupts must be disabled for approximately 5 ms whenever a break point is set in the program. This can crash fast interrupt routines that are running while you stop at a break point or single-step the program. Flash memory or RAM is selected on the **Options** > **Compiler** menu.

Dynamic C Premier provides a number of debugging features. You can single-step your program, either in C, statement by statement, or in assembly language, instruction by instruction. You can set break points, where the program will stop, on any statement. You can evaluate watch expressions. A watch expression is any C expression that can be evaluated in the context of the program. If the program is at a break point, a watch expression can view any expression using local or external variables. If a periodic call to **`runwatch()`** is included in your program, you will be able to evaluate watch expressions by hitting **<Ctrl-U>** without stopping the program.

2. Getting Started

Chapter 2 explains how to connect the programming cable and power supply to the BL2000. Basic Ethernet network connections are shown, and instructions for configuring the network parameters on the BL2000 are included.

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.

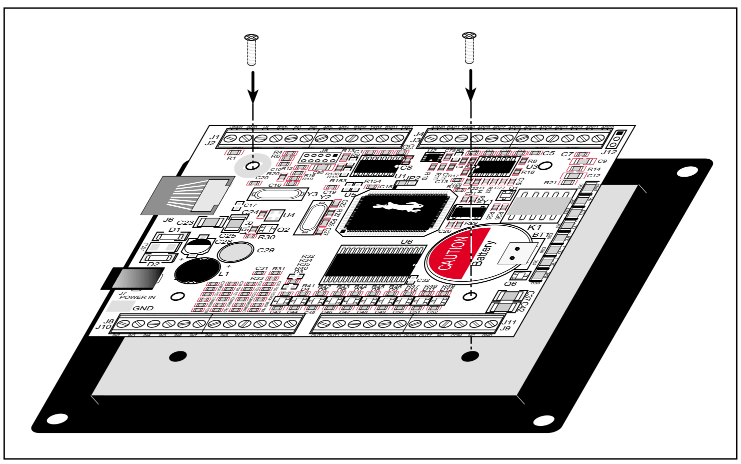


Figure 2. Attach BL2000 to Plastic Enclosure Base

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 monitoring only, as explained in Appendix E, "Programming Cable.") Connect the other end of the programming cable to a COM port on your PC. Make a note of the port to which you connect the cable, as Dynamic C will need to have this parameter configured. Note that COM1 on the PC is the default COM port used by Dynamic C Premier.

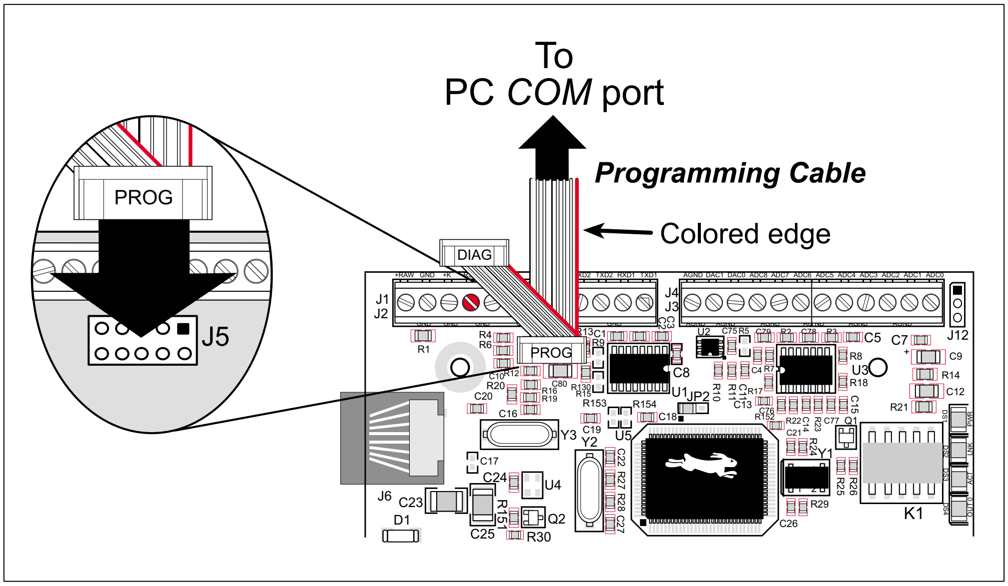


Figure 3. Programming Cable Connections

3. Connect the power supply.

Plug the DC end of the power supply into jack J7, which is labeled **POWER IN**, as shown in Figure 4.

4. Apply power.

Plug in the AC adapter. The orange **PWR** LED and the red **BAD** LED 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.

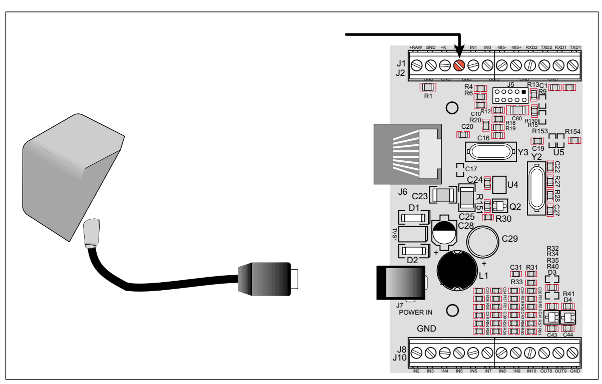


Figure 4. Power Supply Connection

2.2 Installing Dynamic C Premier

If you have not yet installed Dynamic C version 7.04 or later, do so now by inserting the Dynamic C Premier 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 Premier **`setup.exe`** file on your CD drive. Click **OK** to begin the installation once you have selected the **`setup.exe`** file.

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

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 or by double-clicking on the **`.exe`** file associated with **`DcRab`** in the Dynamic C directory.

Dynamic C assumes, by default, that you are using serial port COM1 on your PC. If you ***are*** using COM1, then Dynamic C should detect the BL2000 board and go through a sequence of steps to cold-boot the BL2000 and to compile the BIOS. If the error message "Rabbit Processor Not Detected" appears, you have probably connected to a different PC serial port such as COM2, COM3, or COM4. You can change the serial port used by Dynamic C with the **OPTIONS** menu, then try to get Dynamic C to recognize the BL2000 by selecting **Reset Target/Compile BIOS** on the **Compile** menu. Try the different COM ports in the **OPTIONS** menu until you find the one you are connected to. If you still can't get Dynamic C to recognize the target on any port, then the hookup may be wrong or the COM port might not working on your PC.

If you receive the "BIOS successfully compiled …" message after pressing **<Ctrl-Y>** or starting Dynamic C, and this message is followed by a communications error message, it is possible that your PC cannot handle the 115,200 bps baud rate. Try changing the baud rate to 57,600 bps as follows.

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

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

   to read as follows.

   ```
   #define USE115KBAUD 0   // set to 0 to use 57600 baud
   ```
3. Locate the **Serial options** dialog in the Dynamic C **Options** menu. Change the baud rate to 57,600 bps, then press **<Ctrl-Y>**.
4. Save the changes using **File > Save**.

When you receive the "BIOS successfully compiled …" message and do not receive a communications error message, the target is now ready to compile a program.

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 5.8, "Run the PINGME.C Demo," tests the TCP/IP portion of the board.

2.5 Where Do I Go From Here?

If there are any problems at this point, call Z-World Technical Support at (530)757-3737.

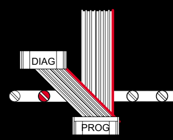
If the sample program ran fine, you are now ready to go on to explore other BL2000 features and develop your own applications.

Chapter 3, "Subsystems," provides a description of the BL2000's features, Chapter 4, "Software," describes the Dyt thmic 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.

- Switching Between Program Mode and Run Mode
- BL2000 Subsystems
- Digital I/O
- Relay Outputs
- Serial Communication
- A/D Converter Inputs
- D/A Converter Outputs
- Memory
- External Interrupts


DIAG
PROG

3.2 BL2000 Subsystems

Figure 6 shows the Rabbit-based subsystems designed into the BL2000.

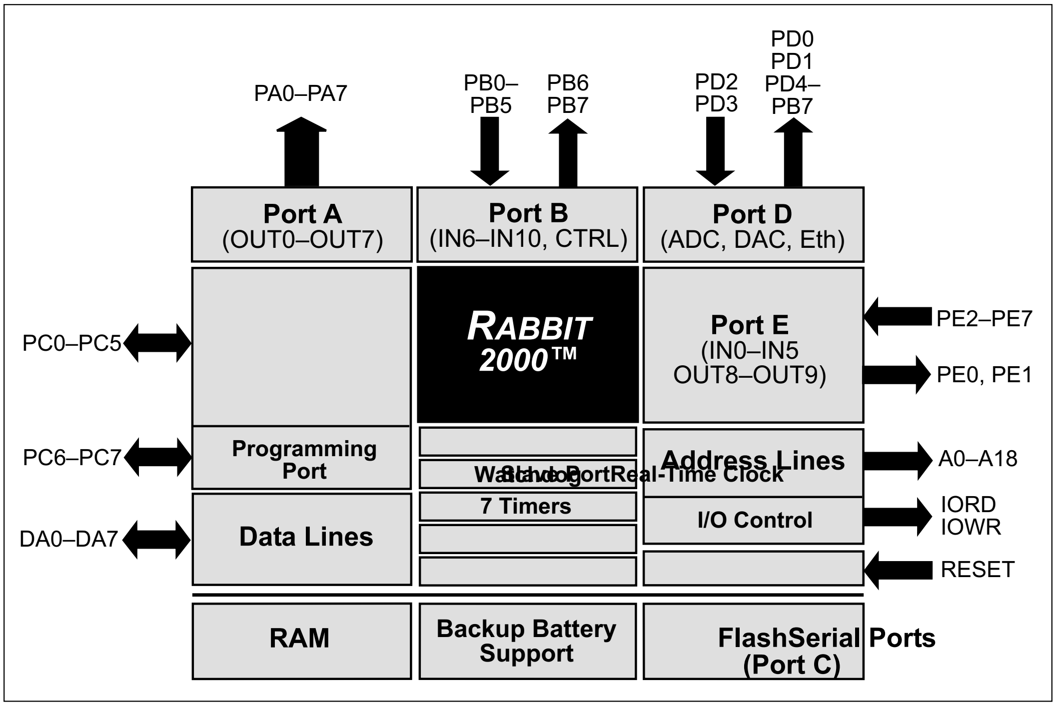


Figure 6. BL2000 Rabbit-Based Subsystems

Table 2 lists the Rabbit 2000 parallel ports and their use in the BL2000.

Table 2. 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 2. 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.

The BL2000 pinouts are shown in Figure 7.

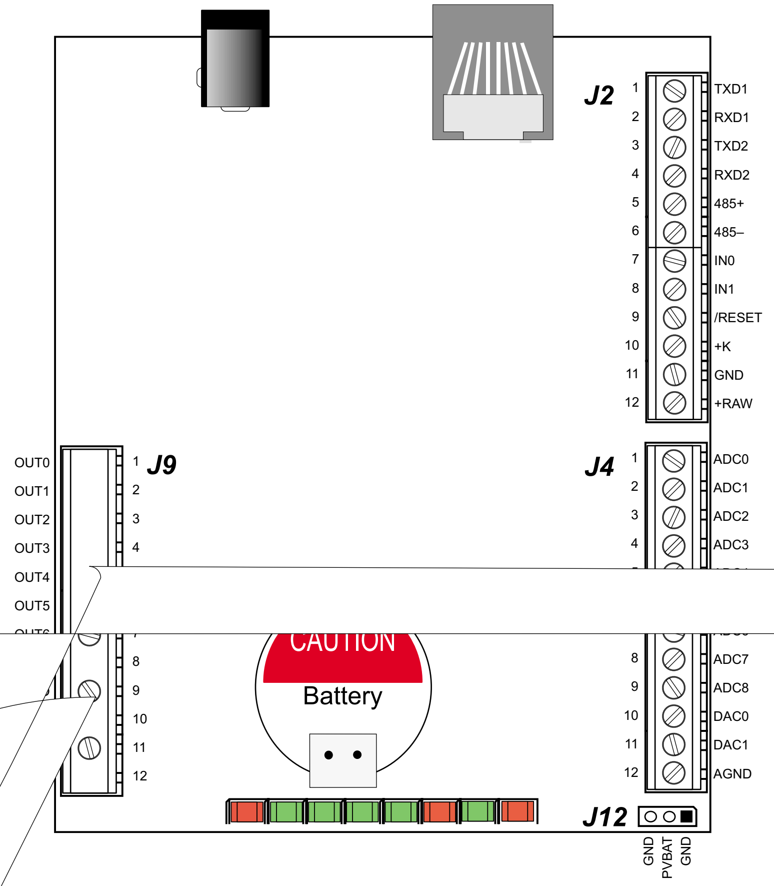


Figure 7. BL2000 Pinouts

...aders and Screw Terminals

...000 models are equipped with 1 × 12 screw terminal strips and a 2-pin power ... The BL2000 and BL2010 also have the RJ-45 Ethernet jack.

...e is provision on the circuit board to accommodate 2 × 17 male headers instead of the ... 12 screw terminal strips supplied, and 2 × 17 male headers can be factory-installed by ...pecial request for volume orders. Please contact your Z-World Sales Representative at +1(530)757-3737 for more information.

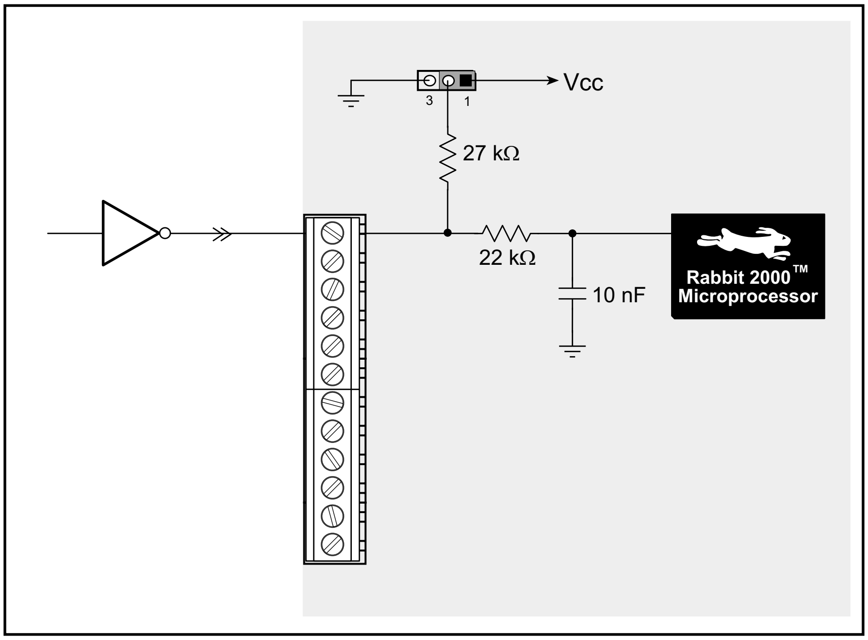


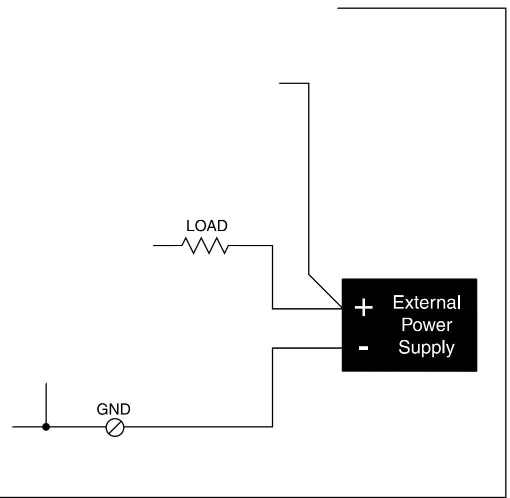
Figure 8(c). Example of Logic Gate Driving BL2000 Digital Input

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.

ık-
ıia
ıge of
ıts.

ıf no pull-
ıors are located
screw terminal



ɔc UT—Special Order

Vcc

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

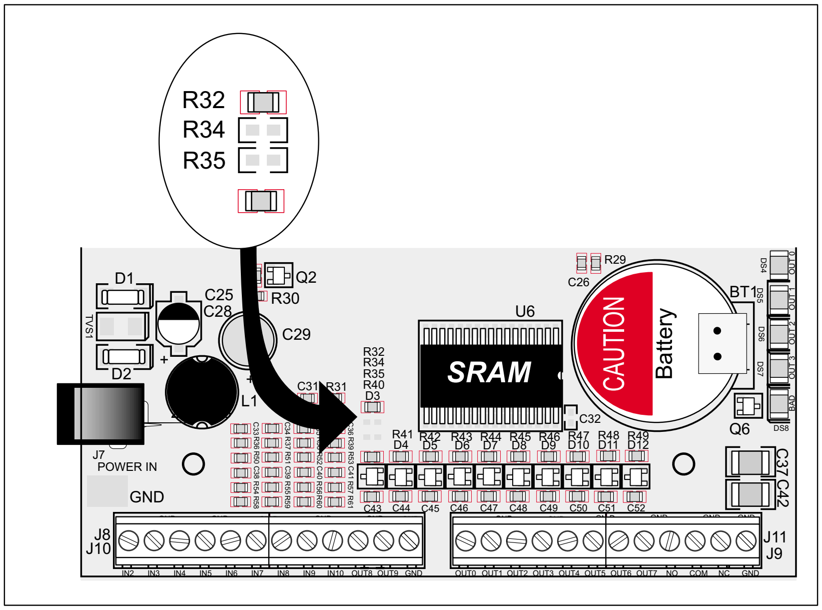


Figure 10. Locations of Resistors R32, R34, and R35

All BL2000 models are factory-configured with sinking outputs and pull-up resistors tied to Vcc via a 0 Ω resistor at R32. Large-volume orders can be factory-configured to your individual needs. Contact your Z-World Sales Representative at +1(530)757-3737 for more information.

3.4 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.

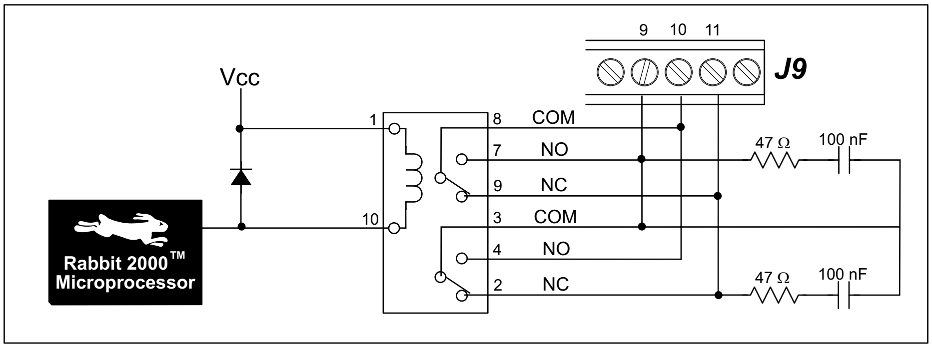


Figure 11. BL2000 Relay Output Contact Connections

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.

3.5 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 3 summarizes the options.

Table 3. 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.

3.5.1 RS-232

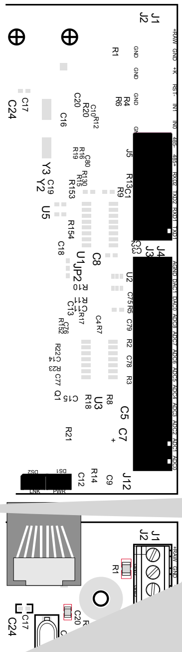
The BL2000 RS-232 serial communication is supported by an RS-232 transceiver, U1. U1 provides the voltage output, slew rate, and input voltage immunity required to meet the RS-232 serial communication protocol. Basically, the chip translates the Rabbit 2000's CMOS/TTL signals to RS-232 signal levels. 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.

3.5.2 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.


J1
J2
R1
C24
C17
C20
C16
Y3
Y2
U5
C8
U3
C5
C7
J12
R21
R14
C12
C9

The BL2000 comes with a 220 Ω termination resistor and 680 Ω bias resistors already installed, as shown in Figure 13.

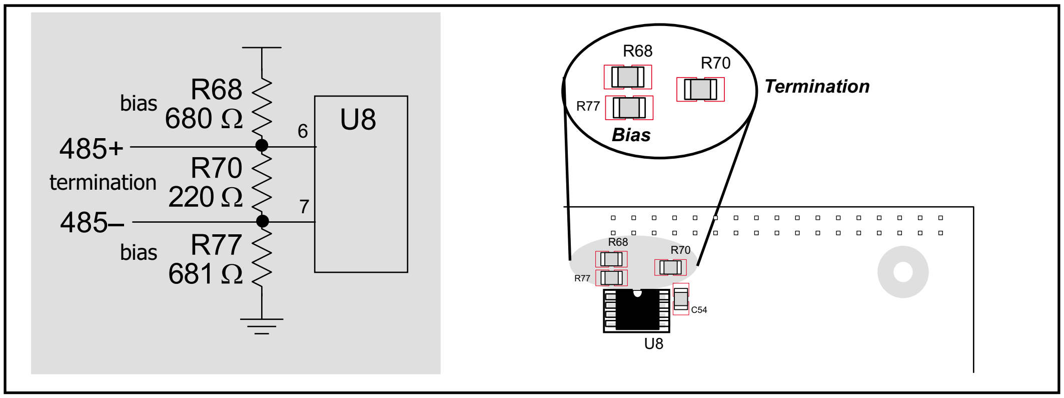


Figure 13. RS-485 Termination and Bias Resistors

The load these bias and termination resistors present to the RS-485 transceiver (U10) limits the number of BL2000 units in a multidrop network to one master and nine slaves, unless the bias and termination resistors are removed. When using more than ten BL2000 units in a multidrop network, leave the termination and bias resistors in place only on the BL2000 unit at each end of the network.

3.5.3 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. The Rabbit 2000 startup-mode pins (SMODE0, SMODE1) are presented to the programming port so that an externally connected device can force the BL2000 to start up in an external bootstrap mode.

> **NOTE:** Refer to the ***Rabbit 2000 Microprocessor User's Manual*** for more information related to the bootstrap mode.

The programming port is used to start the BL2000 in a mode where the BL2000 will download a program from the port and then execute the program. The programming port transmits information to and from a PC while a program is being debugged.

The BL2000 can be reset from the programming port via the **/EXT_RSTIN** line.

The Rabbit 2000 status pin is also presented to the programming port. The status pin is an output that can be used to send a general digital signal.

3.5.4 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.

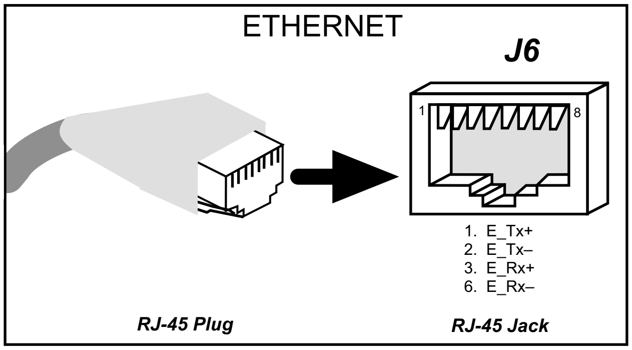


Figure 14. RJ-45 Ethernet Port Pinout

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.

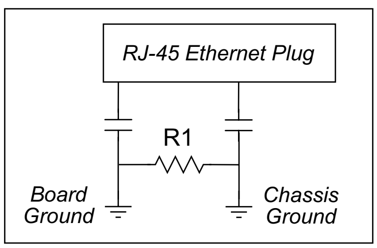


Figure 15. Isolation Resistor R1

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.

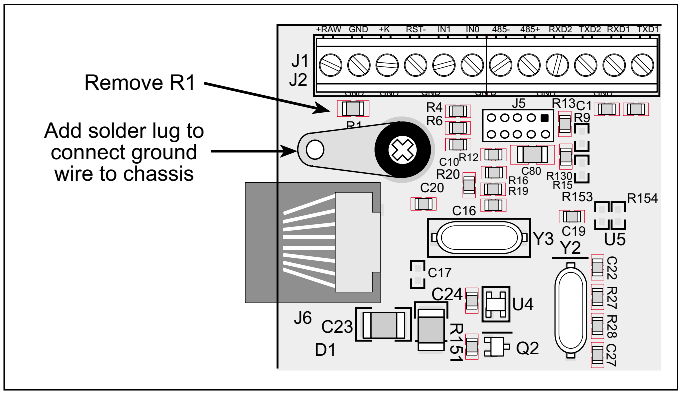


Figure 16. Recommended Location for Ring Lug

3.6 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.

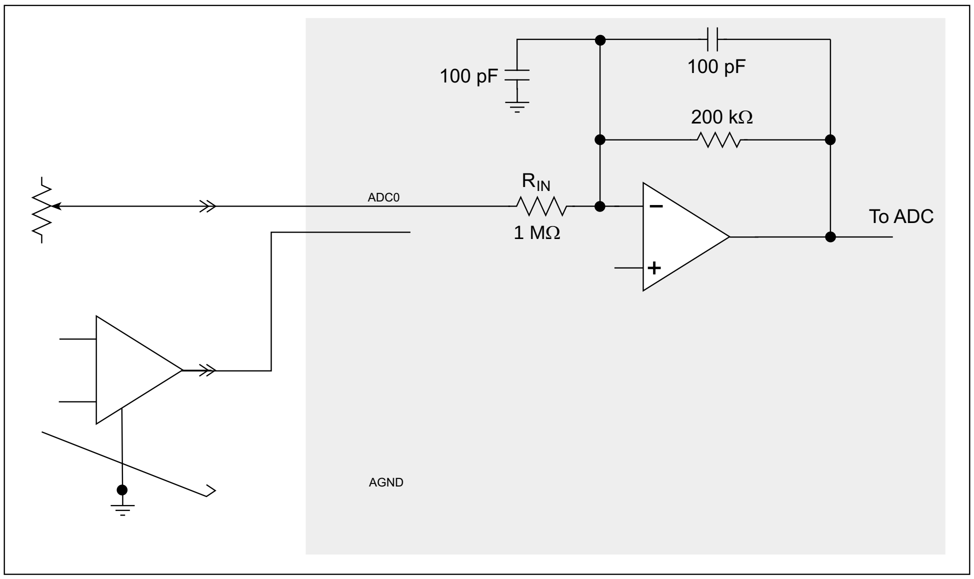


Figure 17. Buffered A/D Converter Inputs

The op-amp is powered from the +V supply. The 1 MΩ and 200 kΩ resistors set the gain (scale factor), which is 5 in this case. This results in a dynamic input range of 5 × 4.096 V 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.

The analog inputs can also be used as digital inputs when required. In this case a lower quality 8-bit D/A converter can be used and the software would assign a 1 or 0 to a voltage based upon whether it is above or below a particular threshold. See the **`digIn`** function description for more information.

3.7 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

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.

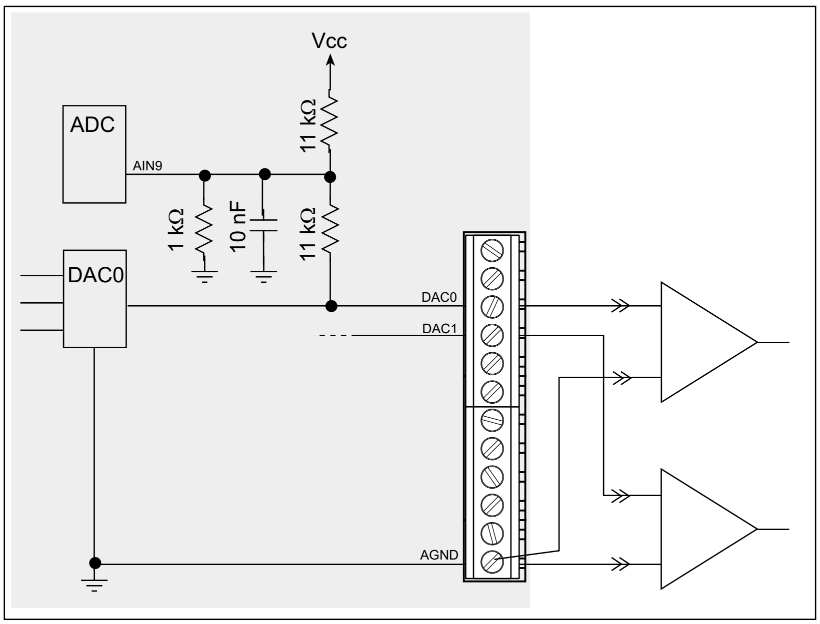


Figure 19. D/A Converter Outputs

3.8 Memory

3.8.1 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 4 lists the jumper settings for the jumpers used to set the SRAM size. The “jumpers” are 0 Ω surface-mounted resistors.

Table 4. Memory Jumper Selections

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

3.8.2 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 4 lists the jumper settings for the jumpers used to set the SRAM size. The “jumpers” are 0 Ω surface-mounted resistors.

> **NOTE:** Z-World 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.

3.9 External Interrupts

The BL2000 does not normally come configured to support external interrupts. However, should this functionality be required, it is possible to modify the board to support one external interrupt. To gain the external interrupt requires giving up the use of two of the 11 digital input lines.

> **NOTE:** There is a potential problem when using external interrupts with the Rabbit 2000 microprocessor. Refer to Technical Note TN301, ***Rabbit 2000 Microprocessor Interrupt Problem***, for more information. The modification given here for the BL2000 takes into account the workaround fix recommended in TN301.

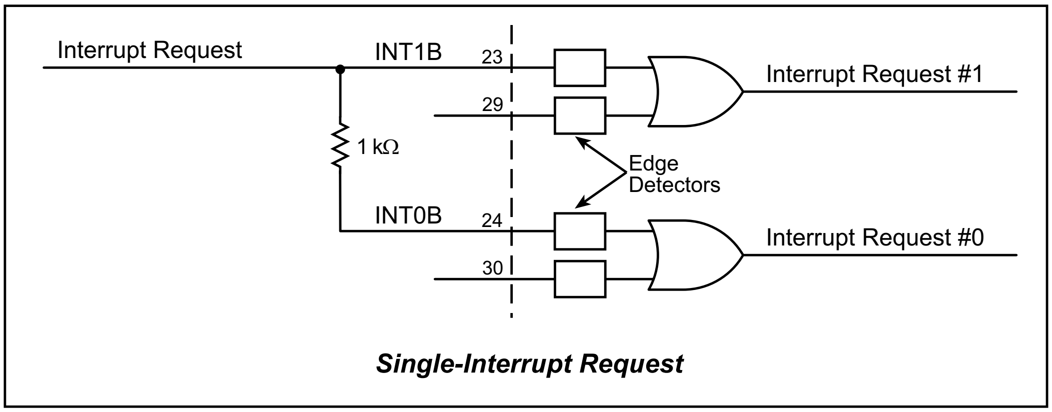


Figure 20. Use of Rabbit 2000 External Interrupts

Remove the following components associated with digital inputs IN2 and IN3: R36, R50, R54, R58, C33, and C38. Install 0 Ω resistors or a jumper across the pads of R54 and R36. Add a 1 kΩ leaded resistor between pins 1 and 2 of screw terminal header J8. The external interrupt is applied at pin 2 of screw terminal header J8.

Figure 21 shows the locations of these components.

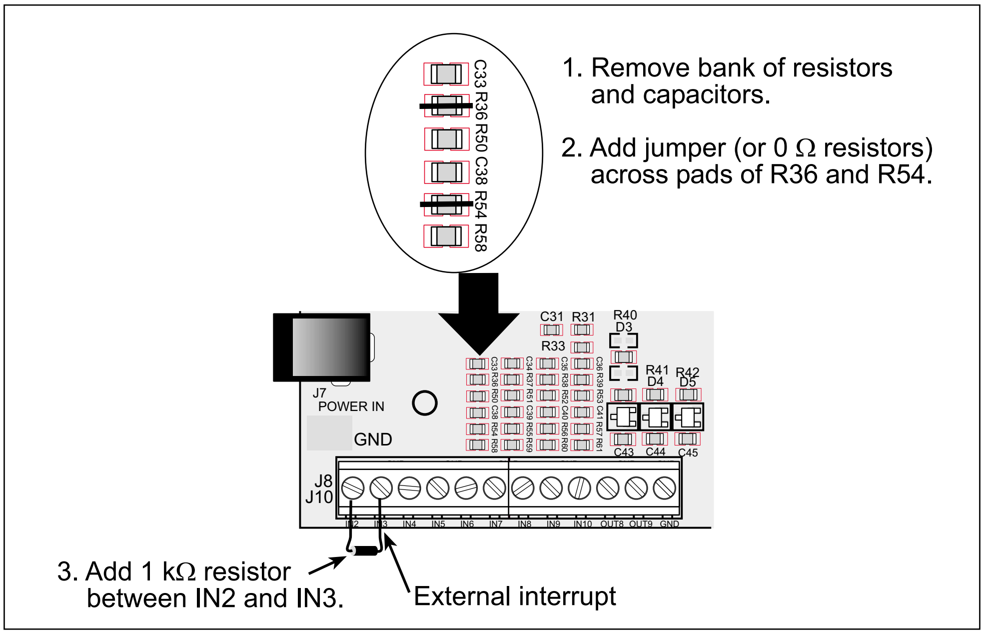


Figure 21. Location of Components to Change for External Interrupt

NOTE: This modification removes the circuit protection provided on digital inputs IN2 and IN3. The external interrupt should therefore not be allowed to transition outside the range -0.5 V to +5.5 V. When Vcc is 5 V, the switching threshold is at 2.5 V ± 0.2 V.

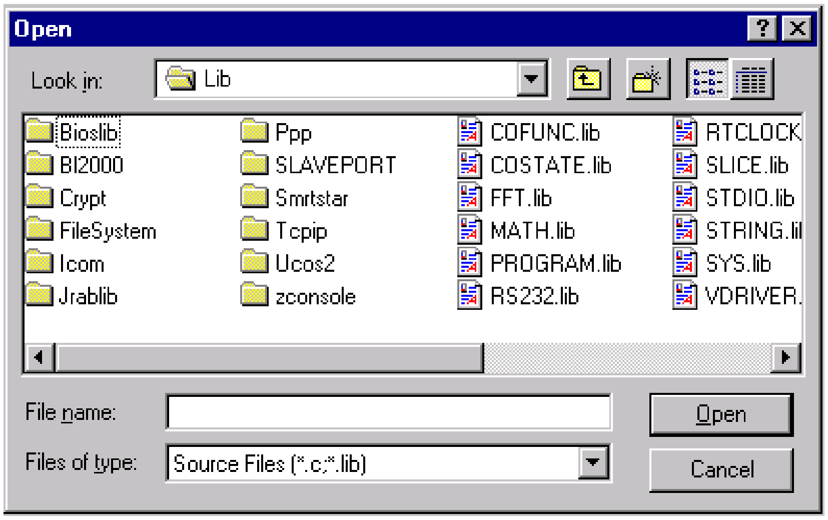
4. SOFTWARE

Dynamic C Premier is an integrated development system for writing embedded software. It runs on an IBM-compatible PC and is designed for use with Z-World 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 BL2000 Libraries

With Dynamic C running, click **File** > **Open**, and select **`Lib`**. The following list of Dynamic C libraries and library directories will be displayed.



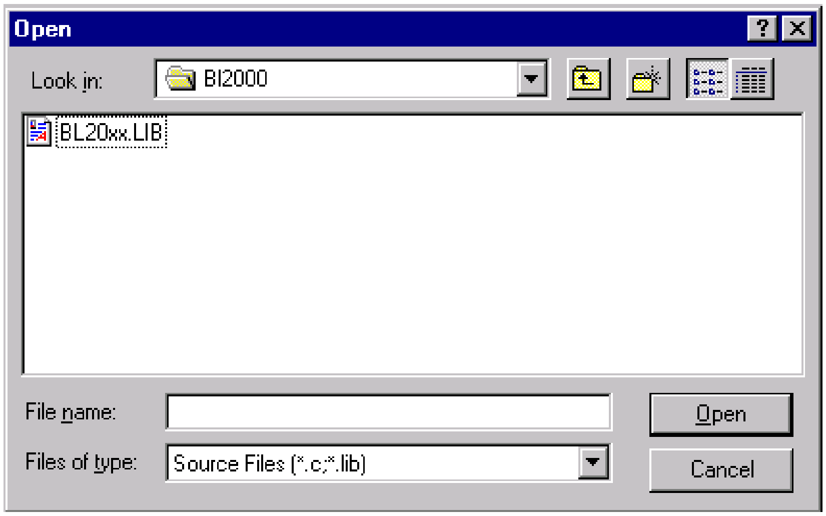
Two library directories are used to develop applications for the BL2000.

- **`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 Premier User's Manual***.

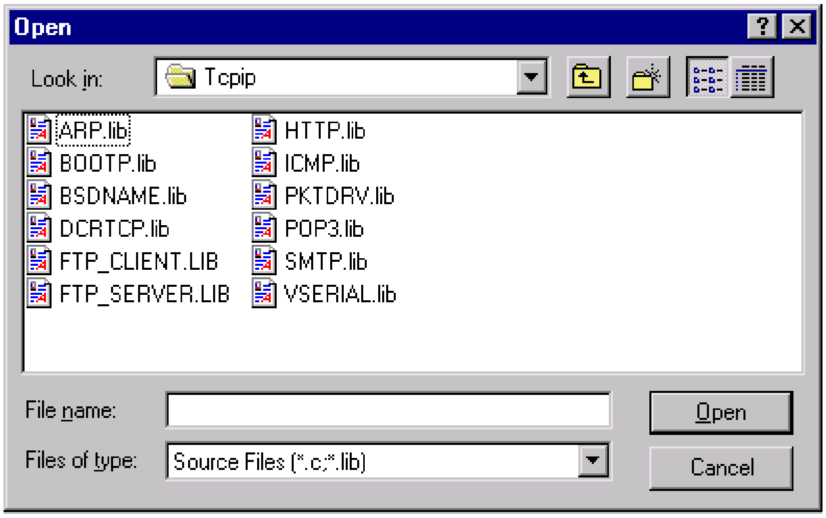
4.1.1 Library Directories

The **BL2000** directory contains the libraries required to operate the BL2000.



- **BL20xx.LIB**—This library supports all BL2000 series boards. The functions in this library are described in this chapter.

The **TCPIP** directory contains libraries with generic TCP/IP functions.



- **ARP.LIB**—address resolution protocol functions.
- **BOOTP.LIB**—bootstrap protocol functions.
- **BSDNAME.LIB**—BSD-style socket routines.
- **DCRTCP.LIB**—TCP/IP functions.
- **FTP_CLIENT.LIB**—FTP client functions.
- **FTP_SERVER.LIB**—FTP server functions.
- **HTTP.LIB**—HTTP handler.
- **ICMP.LIB**—ICMP handler.
- **PKTDRV.LIB**—packet driver functions.
- **POP3.LIB**—POP3 functions.
- **SMTP.LIB**—SMTP handler.
- **VSERIAL.lib**—virtual Telnet functions.

The functions in these libraries are described in the ***Dynamic C TCP/IP User's Manual*** included in the manual set with the ***Dynamic C Premier User's Manual***.

4.2 BL2000 Function APIs

4.2.1 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
PC2	Output	RTS/TXC RS-232	Inactive high

Port	I/O	Function	Output Function State
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`

4.2.2 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. Z-World 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`

4.2.3 Serial Communication

`int serMode(int mode);`

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

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`

4.2.4 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. Z-World 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. Z-World 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`

4.2.5 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.

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.

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`

4.2.6 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.

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.

PARAMETER

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

RETURN VALUE

0 if successful.

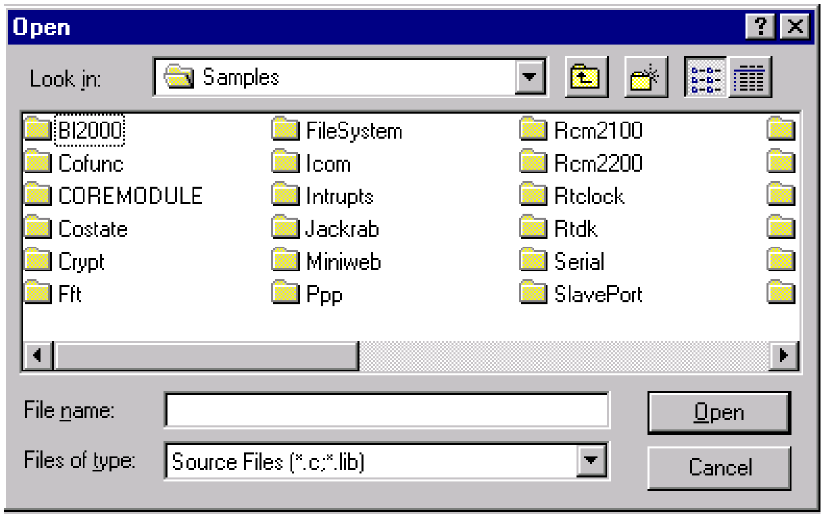
-1 if address or range is invalid.

SEE ALSO

`anaOutEERd, brdInit`

4.3 Sample Programs

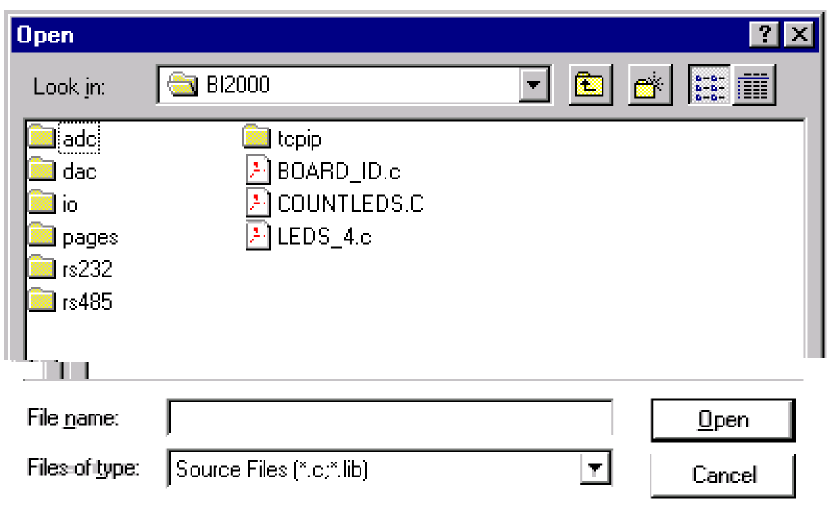
Sample programs are provided in the Dynamic C **Samples** folder, which is shown below.



The various folders contain specific sample programs that illustrate the use of the corresponding Dynamic C libraries. For example, the sample program **PONG.C** demonstrates the output to the **STDIO** window.

The **BL2000** and **TCPIP** folders provide 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.

Let's take a look at the **BL2000** folder.



- **BOARD_ID.C**—This program is used to identify the model of BL2000 being used, and display 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.

4.3.1 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.
- **RELAY.C**—Demonstrates how to control the relay on the BL2000.

4.3.2 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.

4.3.3 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 knownvoltages 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.

4.3.4 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.

4.3.5 TCP/IP Sample Programs

TCP/IP sample programs are described in Chapter 5.

4.4 Using Dynamic C

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 3.1, “Switching Between Program Mode and Run Mode,”) 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 Premier User’s Manual***. TCP/IP specific functions are described in the ***Dynamic C TCP/IP User’s Manual***. Information on using the TCP/IP features and sample programs is provided in Section 5, “Using the TCP/IP Features.”

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 Z-World in a TCP/IP tool kit. More information is available at www.zworld.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 22.

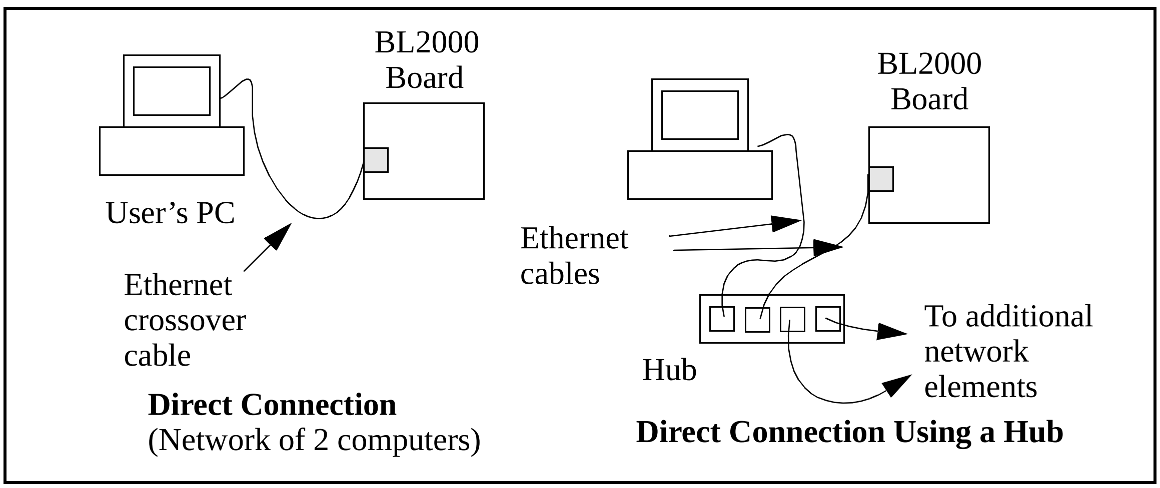


Figure 22. Ethernet Connections

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 Running 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 the user connect his 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.

Obtaining IP addresses to interact over an existing, operating, network can involve a number of complications, and must usually be done with cooperation from your ISP and/or network systems administrator (if your company has one). For this reason, it is suggested that the user begin instead by using a direct connection between a PC and the BL2000 using an Ethernet crossover cable or a simple arrangement with a hub. (A crossover cable should not be confused with regular straight through cables.) The hub and a wide variety of cables can also be purchased from a local computer store.

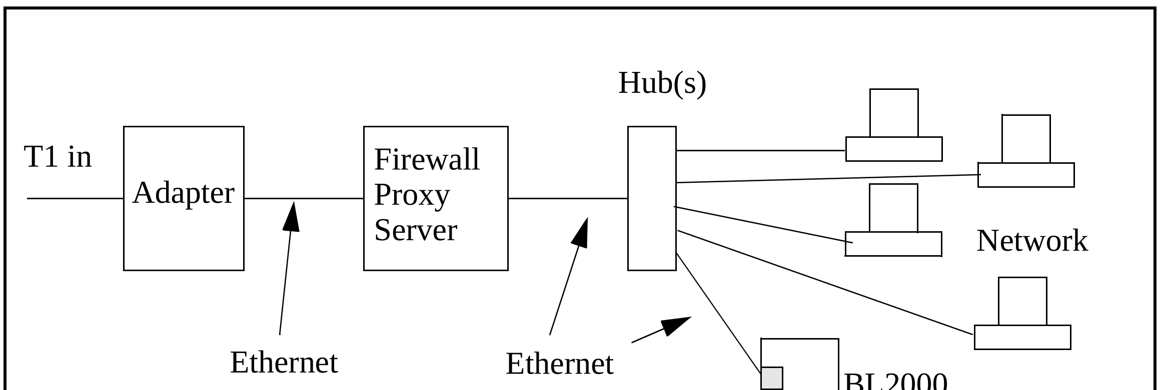
In order to set up this direct connection, the user will have to use a virgin PC (right out of the box), or disconnect a PC from the corporate network, or as yet another approach install a second Ethernet adapter and set up a separate private network attached to the second Ethernet adapter. Disconnecting your PC from the corporate network may be easy or nearly impossible, depending on how it is set up. Mobile PCs, such as laptops, are designed to be connected and disconnected, and will present the least problem. If your PC boots from the network or is dependent on the network for some or all of its disks, then it probably should not be disconnected. If a second Ethernet adapter is used, be aware that Windows TCP/IP will send messages to one adapter or the other, depending on the IP address and the binding order in Microsoft products. Thus you should have different ranges of IP addresses on your private network from those used on the corporate network. If both networks service the same IP address, then Windows may send a packet intended for your private network to the corporate network. A similar situation will take place if you use a dial-up line to send a packet to the Internet. Windows may try to send it via the local Ethernet network if it is also valid for that network.

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

The BL2000 uses a 10Base-T type of Ethernet connection, which is the most common scheme. The RJ-45 connectors are similar to U.S. style telephone connectors, are larger and have 8 contacts.

An alternative to the direct connection using a crossover cable is a direct connection using a hub. The hub relays packets received on any port to all of the ports on the hub. Hubs are low in cost and are readily available. The BL2000 uses 10 Mbps Ethernet, so the hub or Ethernet adapter must be either a 10 Mbps unit or a 10/100 unit that adapts to either 10 or 100 Mbps.

In a corporate setting where the Internet is brought in via a high-speed line, there are typically machines between the outside Internet and the internal network. These machines include a combination of proxy servers and firewalls that filter and multiplex Internet traffic. In the configuration below, the BL2000 could be given a fixed address so any of the computers on the local network would be able to contact it. It may be possible to configure the firewall or proxy server to allow hosts on the Internet to contact the BL2000 directly, but it would probably be easier to place the BL2000 directly on the external network outside the firewall. This avoids some of the configuration complications by sacrificing some security.



Typical Corporate Network

If your system administrator can give you an Ethernet cable along with its IP address, the netmask and the gateway address, then you may be able to run the sample programs without having to setup a direct connection between your computer and the BL2000. You will also need the IP address of the nameserver, the name or IP address of your mail server, and your domain name for some of the sample programs.

5.3 IP Addresses Explained

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

216.103.126.155

10.1.1.6

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

255.255.255.0

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

216.103.126.0

216.103.126.1

216.103.126.2

etc.

216.103.126.254

216.103.126.255

The lowest and highest address are reserved for special purposes. The lowest address (216.102.126.0) is used to identify the local network. The highest address (216.102.126.255) is used as a broadcast address. Usually one other address is used for the address of the gateway out of the network. This leaves 256 - 3 = 253 available IP addresses for the example given.

5.4 How IP Addresses are Used

The actual hardware connection via an Ethernet uses Ethernet adapter addresses (also called MAC addresses). These are 48-bit addresses and are unique for every Ethernet adapter manufactured. In order to send a packet to another computer, given the IP address of the other computer, it is first determined if the packet needs to be sent directly to the other computer or to the gateway. In either case, there is an IP address on the local network to which the packet must be sent. A table is maintained to allow the protocol driver to determine the MAC address corresponding to a particular IP address. If the table is empty, the MAC address is determined by sending an Ethernet broadcast packet to all devices on the local network asking the device with the desired IP address to answer with its MAC address. In this way, the table entry can be filled in. If no device answers, then the device is nonexistent or inoperative, and the packet cannot be sent.

IP addresses are arbitrary and can be allocated as desired provided that they don’t conflict with other IP addresses. However, if they are to be used with the Internet, then they must be numbers that are assigned to your connection by proper authorities, generally by delegation via your service provider.

5.5 Dynamically Assigned Internet Addresses

In many instances, IP addresses are assigned temporarily. This is the normal procedure when you use a dial-up Internet service provider (ISP). Your system will be provided with an IP address that it can use to send and receive packets. This IP address will only be valid for the duration of the call, and further may not actually be a real Internet IP address. Such an address works for browsing the Web, but cannot be used for transactions originating elsewhere since no other system has any way to know the Internet address except by first receiving a packet from you. (If you want to find the IP address assigned by a dial-up ISP, run the program **winipcfg** while connected and look at the address for the ppp adapter under Windows 98.)

In a typical corporate network that is isolated from the Internet by a firewall and/or proxy server using address translation, the IP addresses are not usually actual Internet addresses and may be assigned statically or dynamically. If they are assigned statically, you only have to get an unused IP address and assign it to the BL2000. If the IP addresses are assigned dynamically, then you will have to get an IP address that is valid but outside of the range of IP addresses that are assigned dynamically. This will enable you to communicate from a PC on the network to the BL2000. If you want to communicate to the BL2000 from the external Internet, then an actual Internet IP address must be assigned to the BL2000. It may be possible to setup the firewall to pass a real IP address, or it may be necessary to connect the BL2000 in front of the firewall to accomplish this.

5.6 How to Set IP Addresses in the Sample Programs

Most of the sample programs use macros to define the IP address assigned to the board and the IP address of the gateway, if there is a gateway.

```
#define MY_IP_ADDRESS "216.112.116.155"
#define MY_NETMASK "255.255.255.248"
#define MY_GATEWAY "216.112.116.153"
```

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 "255.255.255.248"
// #define MY_GATEWAY "216.112.116.153"
```

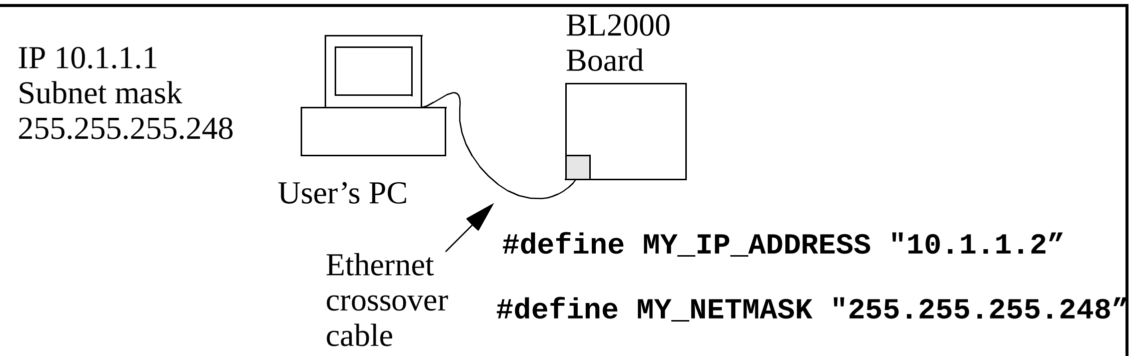
In this case, the gateway is not used and is commented out. The IP address of the board is defined to be 10.1.1.2. The IP address of you PC can be defined as 10.1.1.1.

5.7 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 10.1.1.1 with the subnetmask 255.255.255.248 under Windows 98, do the following.

Click on **Start > Settings > Control Panel** to bring up the Control Panel, and then double-click the Network icon. In the window find the line of the form **TCP/IP > Ethernet adapter name**. Double-click on this line to bring up the TCP/IP properties dialog box. You can edit the IP address directly and the subnet mask. (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.

The method of setting the IP address may differ for different versions of Windows, such as 95, NT or 2000.



Direct Connection PC to BL2000

5.8 Run the `PINGME.C` Demo

In order to run this program, edit the IP address and netmask in the **`PINGME.C`** program (**`SAMPLES\TCPIP\ICMP`**) to the values given above (10.1.1.2 and 255.255.255.248). Compile the program and start it running under Dynamic C. The crossover cable is connected from your computer's Ethernet adapter to the BL2000's RJ-45 Ethernet connector. 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 ping program:

```
ping 10.1.1.2
```

or by **Start** > **Run**

and typing the command

```
ping 10.1.1.2
```

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.

5.9 Running More Demo 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. In order to run these sample programs, edit the IP address as for the pingme program, compile the program and start it executing. Then bring up your Web browser and enter the following server address: http://10.1.1.2.

This should bring up the Web page served by the sample program.

The sample program **`SMTP.C`** (**`SAMPLES\BL2000\TCPIP\`**) allows you to send an E-mail when a switch on the Demonstartion 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. To run this program, edit the IP address, compile the program, and start it running. Run the Telnet program on your PC (**Start > Run** **`telnet 10.1.1.2`**). 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.

5.10 Where Do I Go From Here?

If there are any problems at this point, call Z-World Technical Support at +1(530)757-3737.

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***.

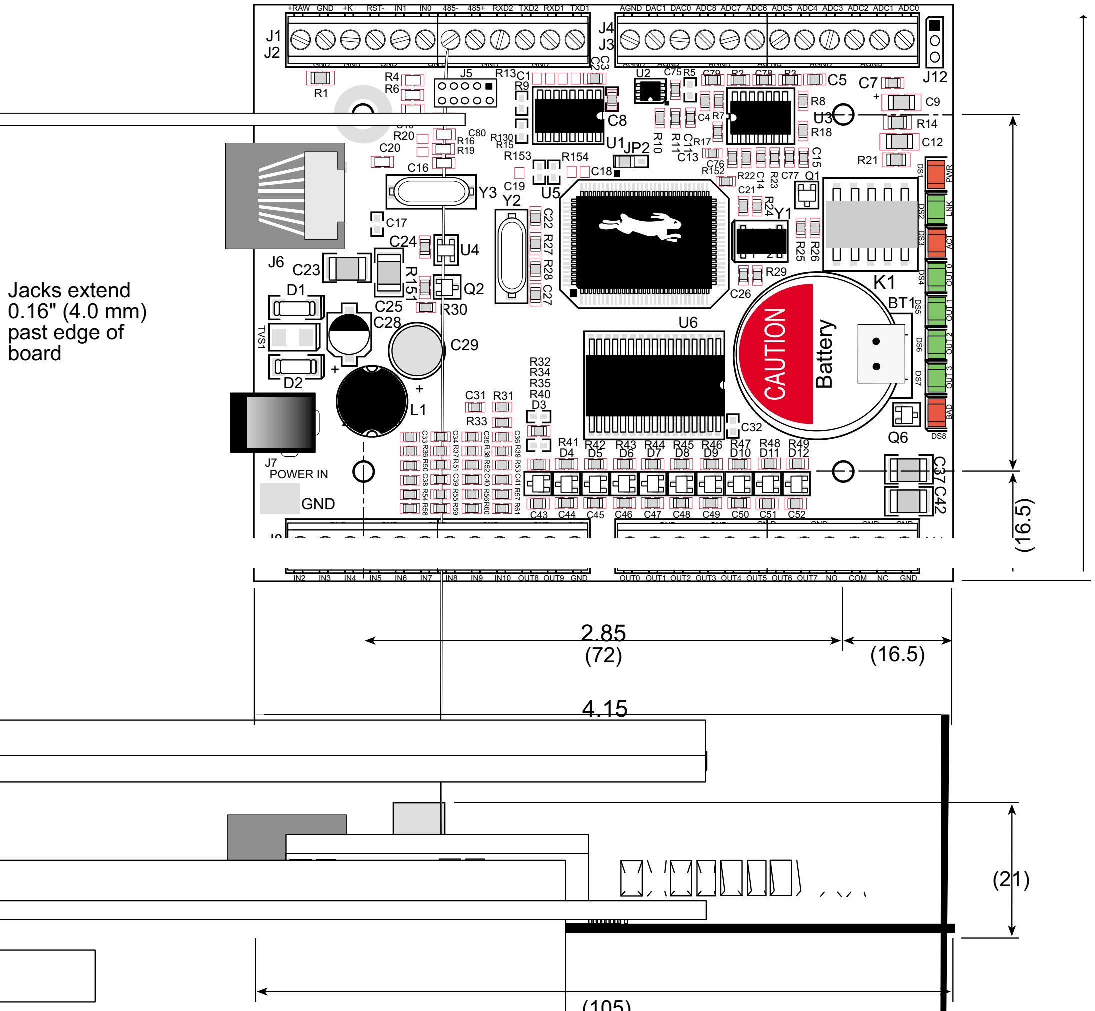
Refer to the ***Dynamic C TCP/IP User's Manual*** to develop your own applications. ***An Introduction to TCP/IP*** provides background information on TCP/IP, and is available on Z-World's Web site.

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.



ions

followed by millimeters enclosed in parentheses.

Table A-1 lists the electrical, mechanical, and environmental specifications for the BL2000.

Table A-1. BL2000 Specifications

Parameter	Specification
Board Size	3.43" × 4.15" × 0.82" (87 mm × 105 mm × 21 mm)
Connectors	one RJ-45 (Ethernet) one 2 × 5, 2 mm pitch (serial programming port) one power jack for AC adpater four 12-terminal screw connectors (18 to 26 AWG wire) for analog and digital I/O, relay
Ethernet Interface	Direct connection to 10Base-T Ethernet networks via RJ-45 connection
Temperature	–40°C to +70°C
Humidity	5% to 95%, noncondensing
External Input Voltage	9 V to 40 V DC or 22 V to 26 V AC
Power Consumption	1.5 W maximum
Onboard Voltage Regulator	Surface-mount switching regulator sources 5 V at up to 1 A
Digital I/O	Up to 28 digital I/O: 11 inputs: hardware-configurable pull-up or pull-down, ± 36 V DC, switching threshold 2.4 V typical up to 7 dual-purpose unbuffered analog inputs that may be software-configured for use as digital inputs, 0 V to 48 V DC, switch threhold may be set in software 10 outputs: sinking or sourcing, +40 V DC, 200 mA maximum per channel
Analog Inputs	Four 12-bit A/D converter inputs, ± 10 V DC, 1 MΩ input impedance Five 12-bit dual-purposeA/D converter inputs, 0 V to 48 V DC, 12 kΩ input impedance Two additional 12-bit dual-purpose A/D converter inputs, 0 V to 48 V DC, 12 kΩ input impedance (BL2010 and BL2030)
Analog Outputs (BL2000 and BL2020)	Two 12-bit D/A converter outputs, 0 V to 4.000 V DC, 1 mA max.
Relay Output	SPDT (N.O., N.C., COM) with snubbers: 1 A @ 30 V DC, 300 mA @ 120 V AC max. contact settling time 4 ms
Microprocessor	Rabbit 2000™
Clock	22.1 MHz
SRAM	128K, surface mount
Flash EPROM	256K, surface mount

Table A-1. BL2000 Specifications (continued)

Parameter	Specification
Timers	Five 8-bit timers, one 10-bit timer with two match registers, five timers are cascadable
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	Maximum standard asynchronous 230,400 bps
Watchdog/Supervisor	Yes
Time/Date Clock	Yes
Backup Battery	Yes: 3 V lithium coin type, 950 mA·h standard solder-in; optional 3 V, 160 mA·h, for battery using onboard battery holder; external battery connecor

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

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.

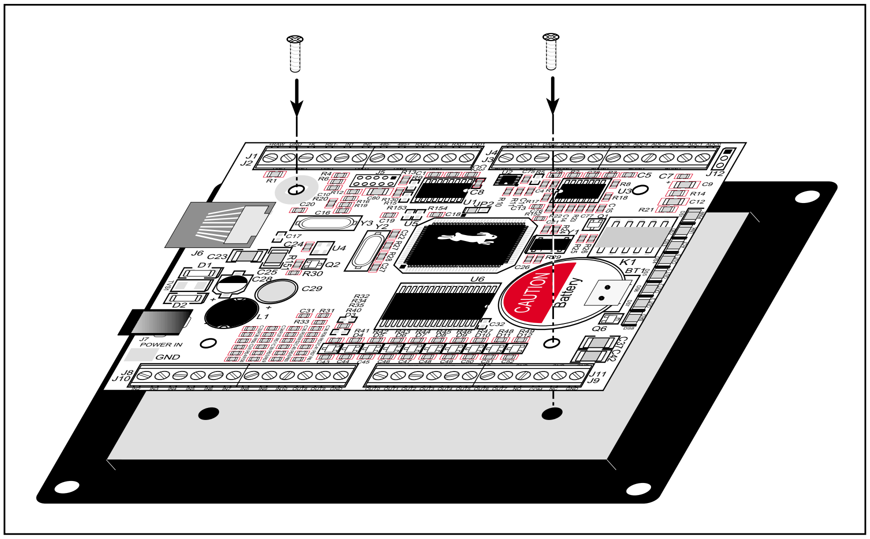


Figure B-1. Attach BL2000 to Plastic Enclosure Base

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.

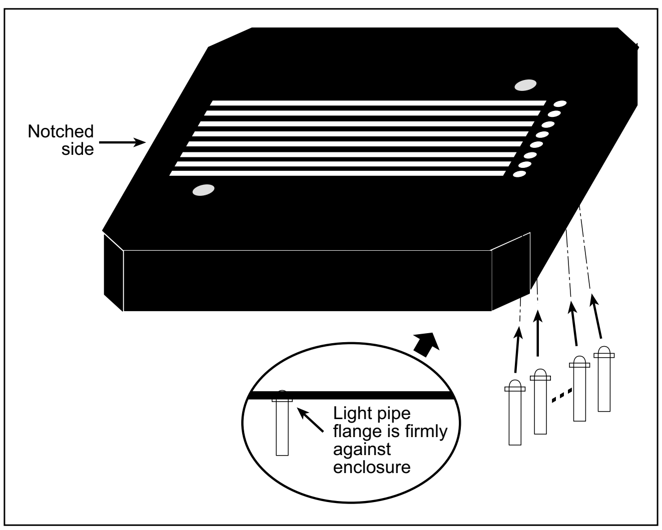


Figure B-2. Install Light Pipes in Enclosure Top

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. Att[redacted]e top to the base.

Positi[redacted]top over the plastic enclosure base as shown below in Figure B-3. Attach[redacted]op to the base using the two 4-40 × ½ screws supplied. If you install[redacted]s, be sure they are aligned over the LEDs as shown.

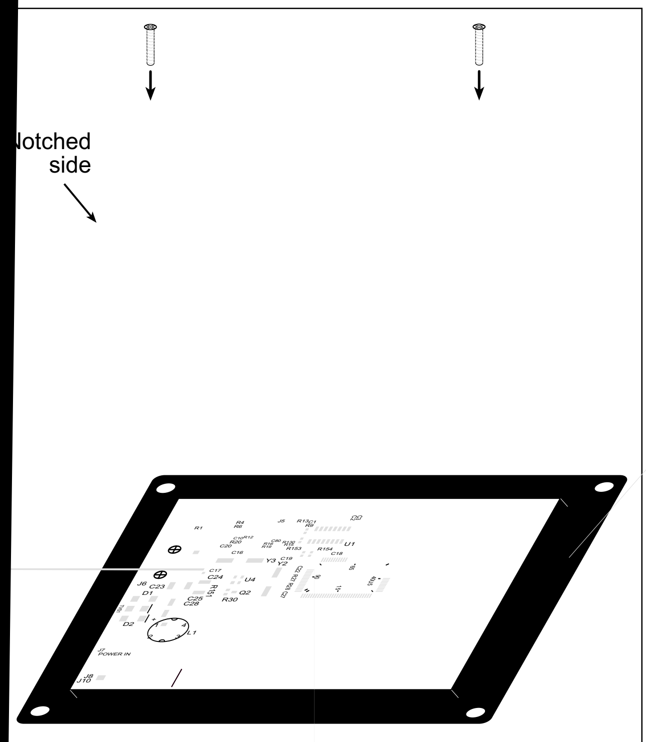


Figure B-3. Attach Enclosure Top

4. M[redacted]losure (optional).

Use f[redacted]to attach the assembled plastic enclosure to the surface on which it will b[redacted]s step applies to production versions of BL2000 boards once development [redacted]leted.

B.2 Dimensions

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

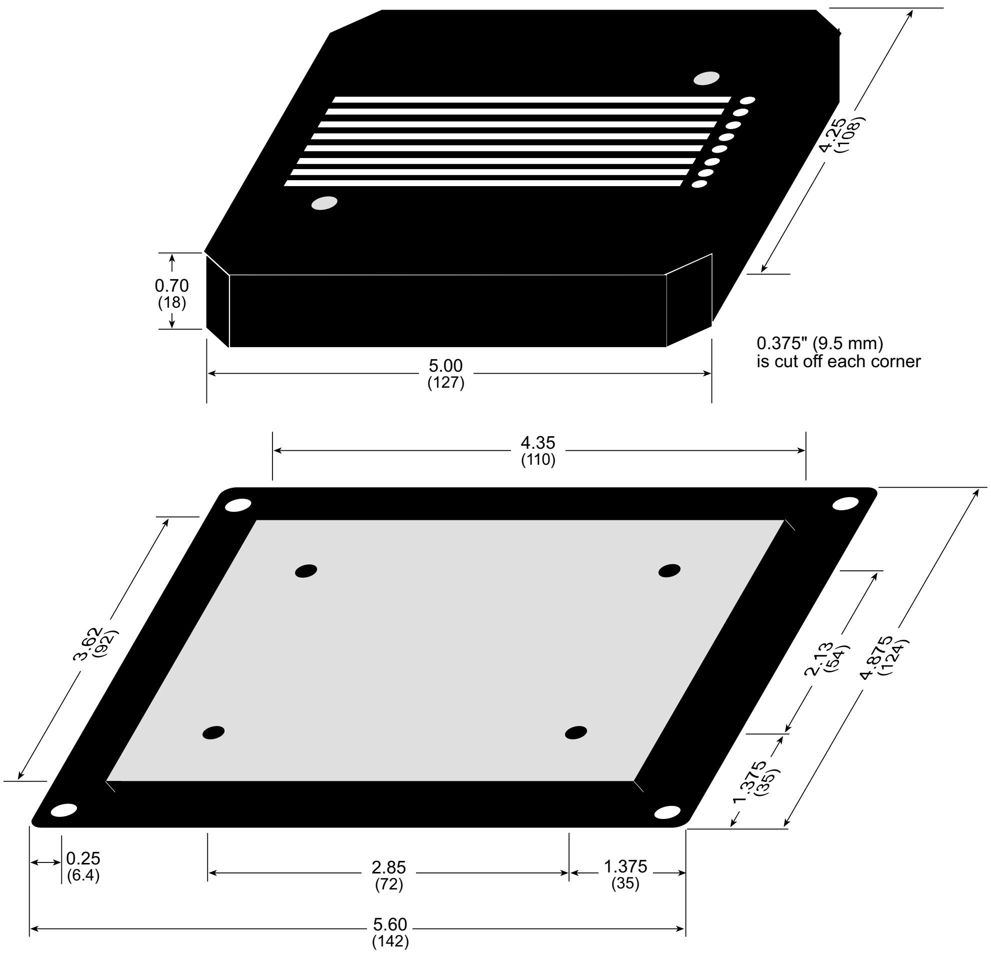


Figure B-4. Plastic Enclosure Dimensions

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 Shottky diode at D1 as shown in Figure C-1. The Shottky diode has a low forward voltage drop, 0.3 V, which keeps the minimum DCIN required to power the BL2000 lower than a normal silicon diode would allow.

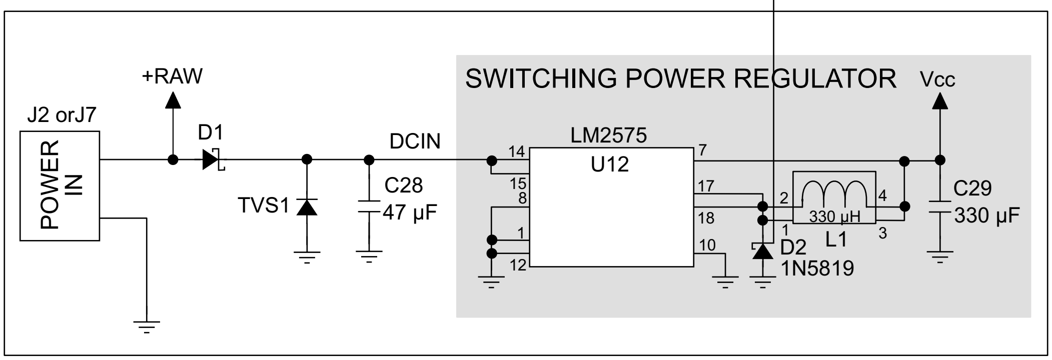


Figure C-1. BL2000 Power Supply

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 significant droop will be measured at DCIN, the voltage will never drop below as +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.

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. The battery and its associated support circuits can be disabled via jumpers.

A 950 mA·h lithium battery provides power to the real-time clock and SRAM when external power is removed from the circuit. The drain on the battery is typically less than 20 µA when there is no external power applied. In this case, the battery can last more than 5 years:

$$\frac{950\ \text{mA}\cdot\text{h}}{20\ \mu\text{A}} = 5.4\ \text{years.}$$

The drain on the battery is typically less than 4 µA when external power ***is*** applied. The battery life is now determined by its shelf life:

$$\frac{950\ \text{mA}\cdot\text{h}}{4\ \mu\text{A}} = 27\ \text{years (shelf life = 10 years).}$$

Since the shelf life of the battery is 10 years, the battery can last for its full shelf life when external power is applied continuously to the BL2000.

An external battery may be connected to the BL2000 via header J12. The existing battery does not have to be removed. Alternatively, for large-volume orders, a battery can be installed in a battery holder for convenience in replacing the battery. Contact your Z-World Sales Representative at +1(530)757-3737 for more information.

Figure C-2 shows the battery backup circuit.

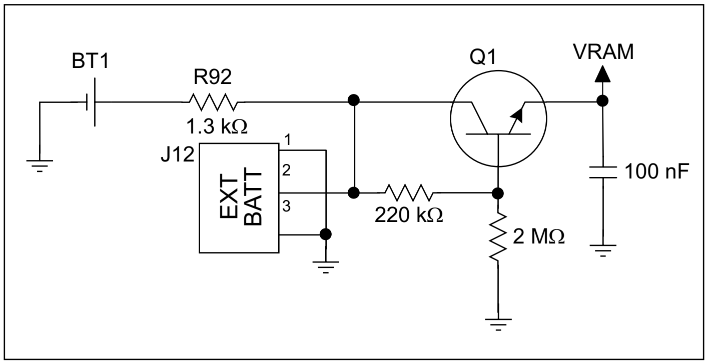


Figure C-2. BL2000 Backup Battery Circuit

The battery-backup circuit serves two 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. The regulator in Figure C-2 sets VRAM to approximately 2 V when Vcc power is lost.
- It ensures that current can flow only ***out*** of the battery to prevent charging the battery.

VRAM and Vcc are nearly equal (<100 mV, typically 10 mV) when power is supplied to the BL2000. R92 prevents any catastrophic failure of Q1 by limiting current from the battery.

C.2.1 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.

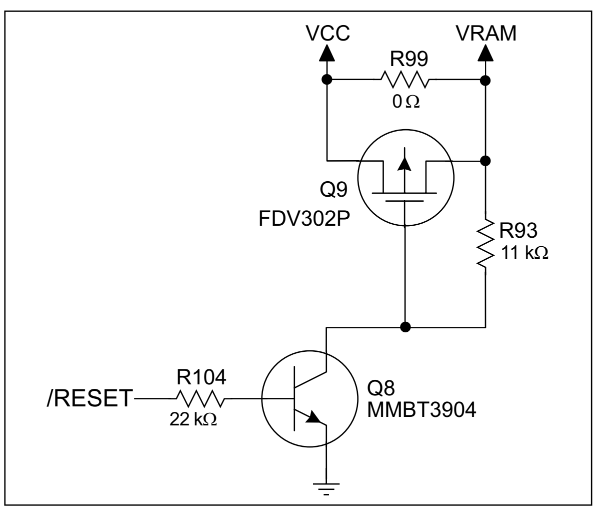


Figure C-3. VRAM Switch

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.2.2 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.

C.3 Chip Select Circuit

Figure C-4 shows a schematic of the chip select circuit.

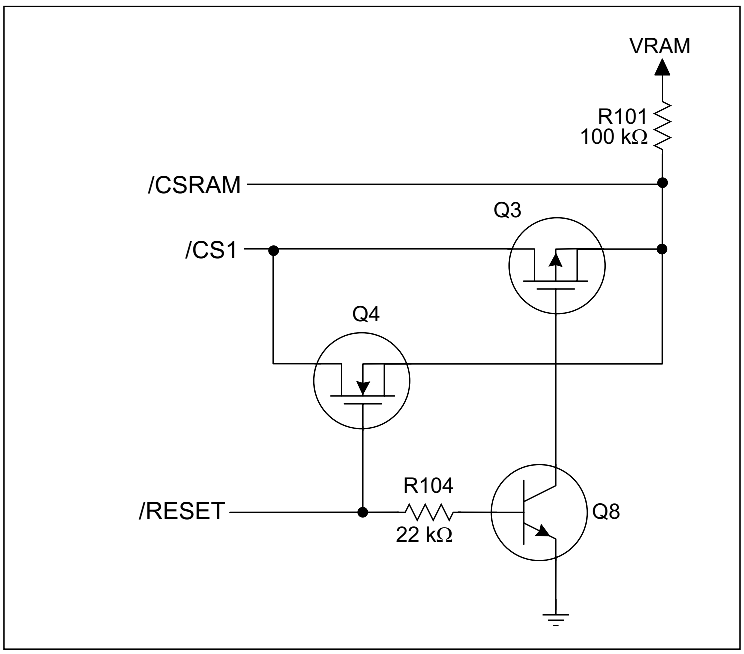


Figure C-4. Chip Select Circuit

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.

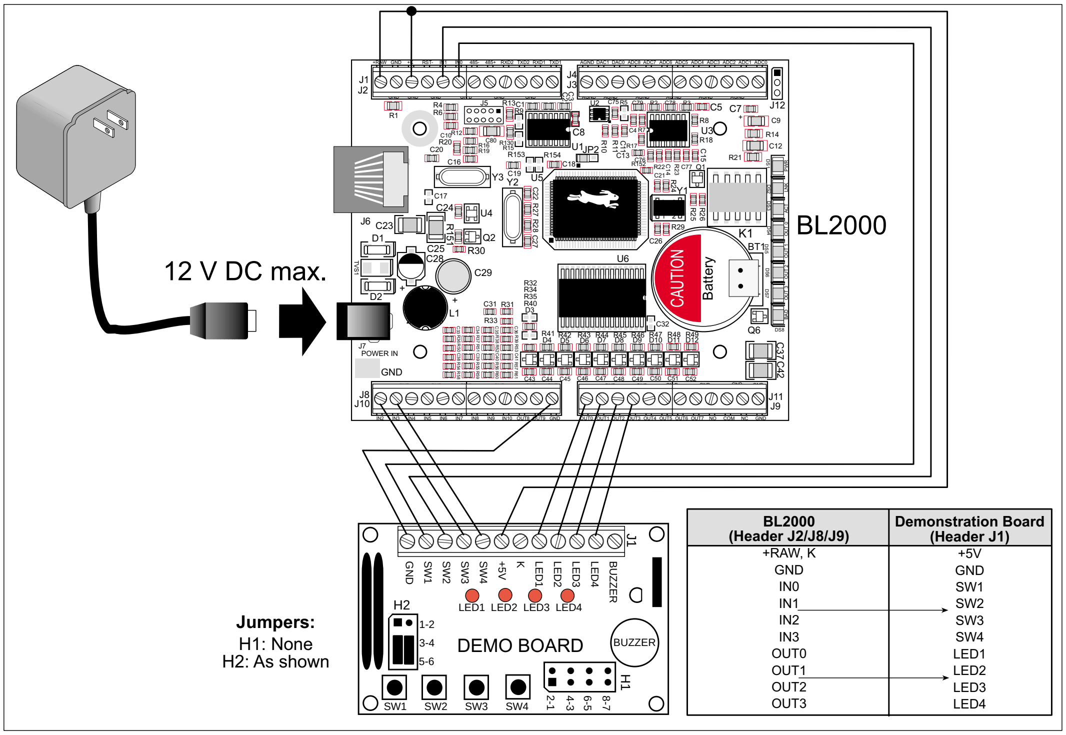
Appendix D. Demonstration Board

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

D.1 Connecting Demonstration Board

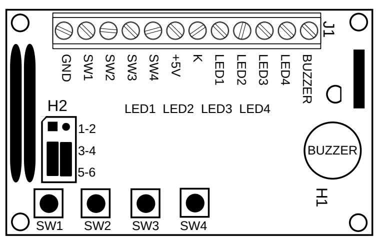
Before running sample programs based on the Demonstration Board, you will have to connect the Demonstration Board from the BL2000 Tool Kit to the BL2000 board. Proceed as follows.

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

APPENDIX E. PROGRAMMING CABLE

Appendix E provides additional theoretical information for the Rabbit 2000™ microprocessor when using the **DIAG** and **PROG** connectors on the programming cable. The **PROG** connector is used only when the programming cable is attached to the programming connector (header J5) while a new application is being developed. Otherwise, the **DIAG** connector on the programming cable allows the programming cable to be used as an RS-232 to CMOS level converter for serial communication, which is appropriate for monitoring or debugging a BL2000 system while it is running.

The programming port, which is shown in Figure E-1, can serve as a convenient communications port for field setup or other occasional communication need (for example, as a diagnostic port). There are several ways that the port can be automatically integrated into software. If the port is simply to perform a setup function, that is, write setup information to flash memory, then the controller can be reset through the programming port and a cold boot performed to start execution of a special program dedicated to this functionality.

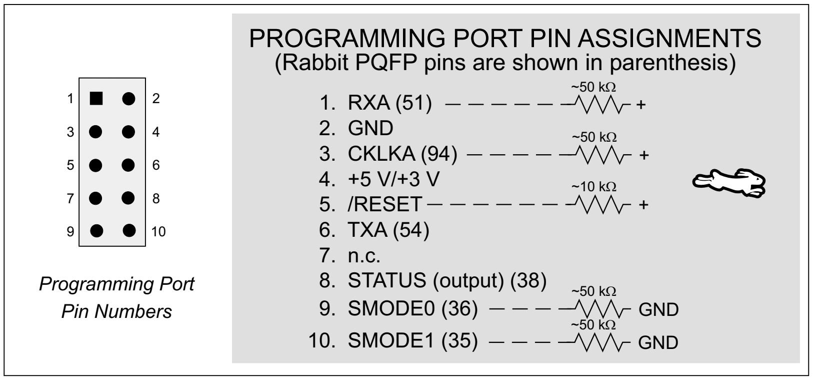


Figure E-1. Programming Port Pin Assignments

When the **PROG** connector is used, the /RESET line can be asserted by manipulating DTR and the STATUS line can be read as DSR on the serial port. The target can be restarted by pulsing reset and then, after a short delay, sending a special character string at 2400 bps. To simply restart the BIOS, the string 80h, 24h, 80h can be sent. When the BIOS is started, it can tell whether the programming cable is connected because the SMODE1 and SMODE0 pins are sensed as being high. This will cause the Rabbit 2000 to enter the bootstrap mode. The Dynamic C programming mode then can have an escape message that will enable the diagnostic serial port function.

Alternatively, the **DIAG** connector can be used to connect the programming port. The /RESET line and the SMODE1 and SMODE0 pins are not connected to this connector. The programming port is then enabled as a diagnostic port by polling the port periodically to see if communication needs to begin or to enable the port and wait for interrupts. The pull-up resistors on RXA and CLKA prevent spurious data reception that might take place if the pins floated.

If the clocked serial mode is used, the serial port can be driven by having two toggling lines that can be driven and one line that can be sensed. This allows a conversation with a device that does not have an asynchronous serial port but that has two output signal lines and one input signal line.

The line TXA (also called PC6) is zero after reset if the cold-boot mode is not enabled. A possible way to detect the presence of a cable on the programming port is for the cable to connect TXA to one of the SMODE pins and then test for the connection by raising PC6 (by configuring it as a general output bit) and reading the SMODE pin after the cold-boot mode has been disabled.

Once you establish that the programming port will never again be needed for programming, it is possible to use the programming port for additional I/O lines. Table E-1 lists the pins available for this alternate configuration.

Table E-1. BL2000 Programming Port Pinout Configurations

Pin	Pin Name	Default Use	Alternate Use	Notes
1	RXA	Serial Port A	PC6—Input	
2	GND			
3	CLKA		PB1—Bitwise or parallel programmable input	
4	VCC			
5	RESET			Connected to reset generator U4
6	TXA	Serial Port A	PC7—Output	
8	STATUS		Output	
9	SMODE0		Input	Must be low when BL2000 boots up
10	SMODE1		Input	Must be low when BL2000 boots up

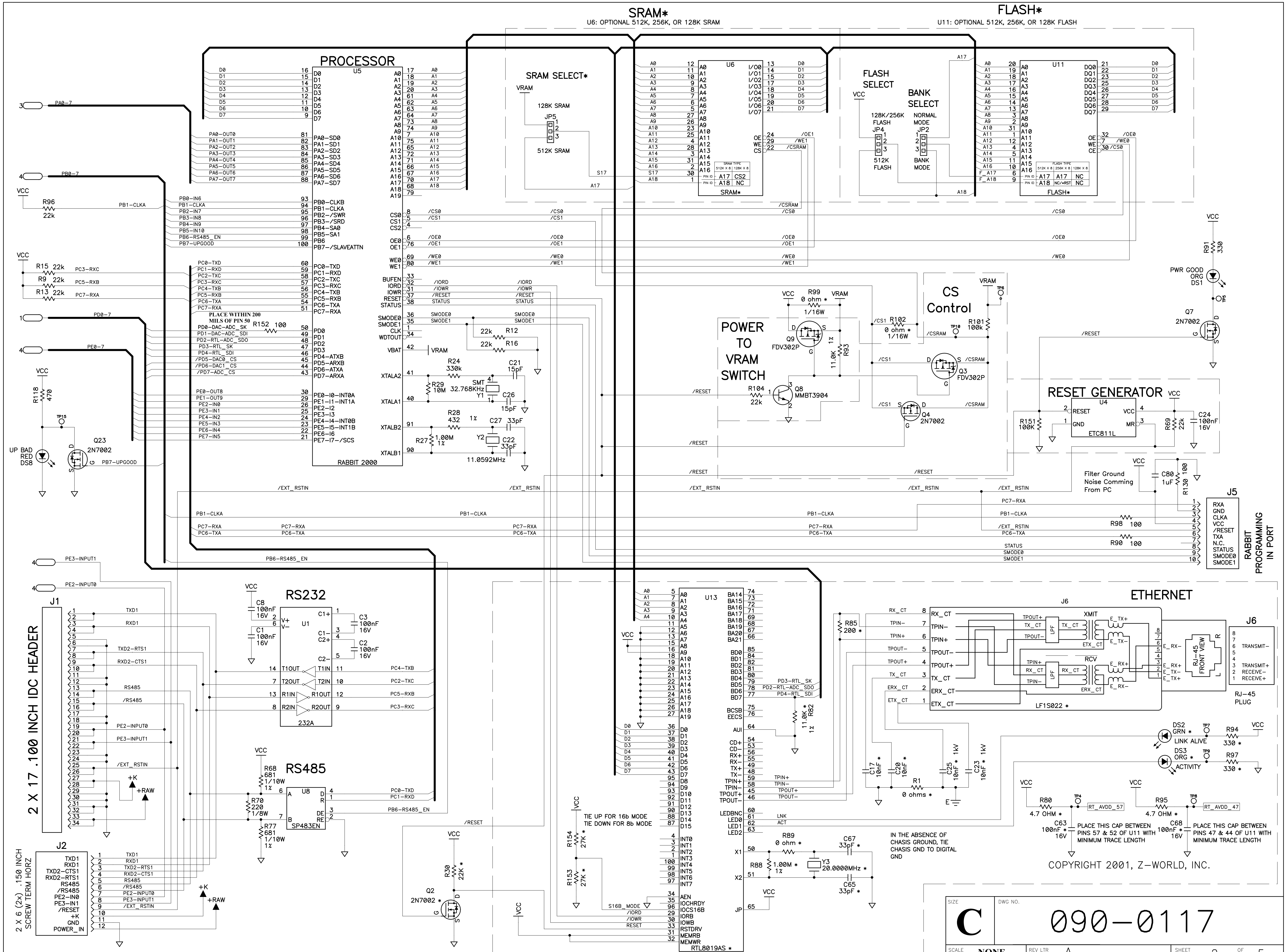
INDEX

SCHEMATICS

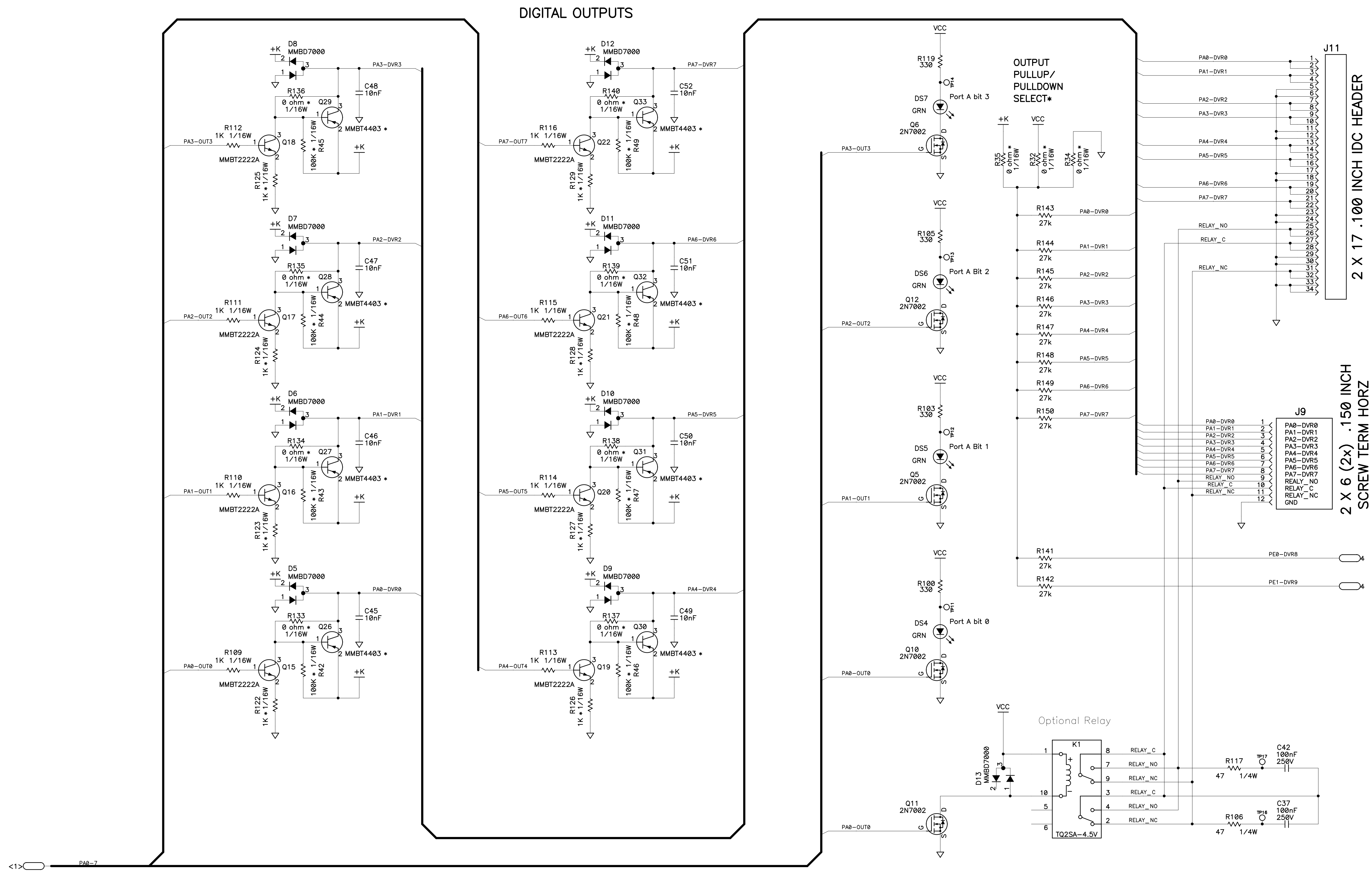
090-0117 BL2000 Schematic

090-0042 Demonstration Board Schematic

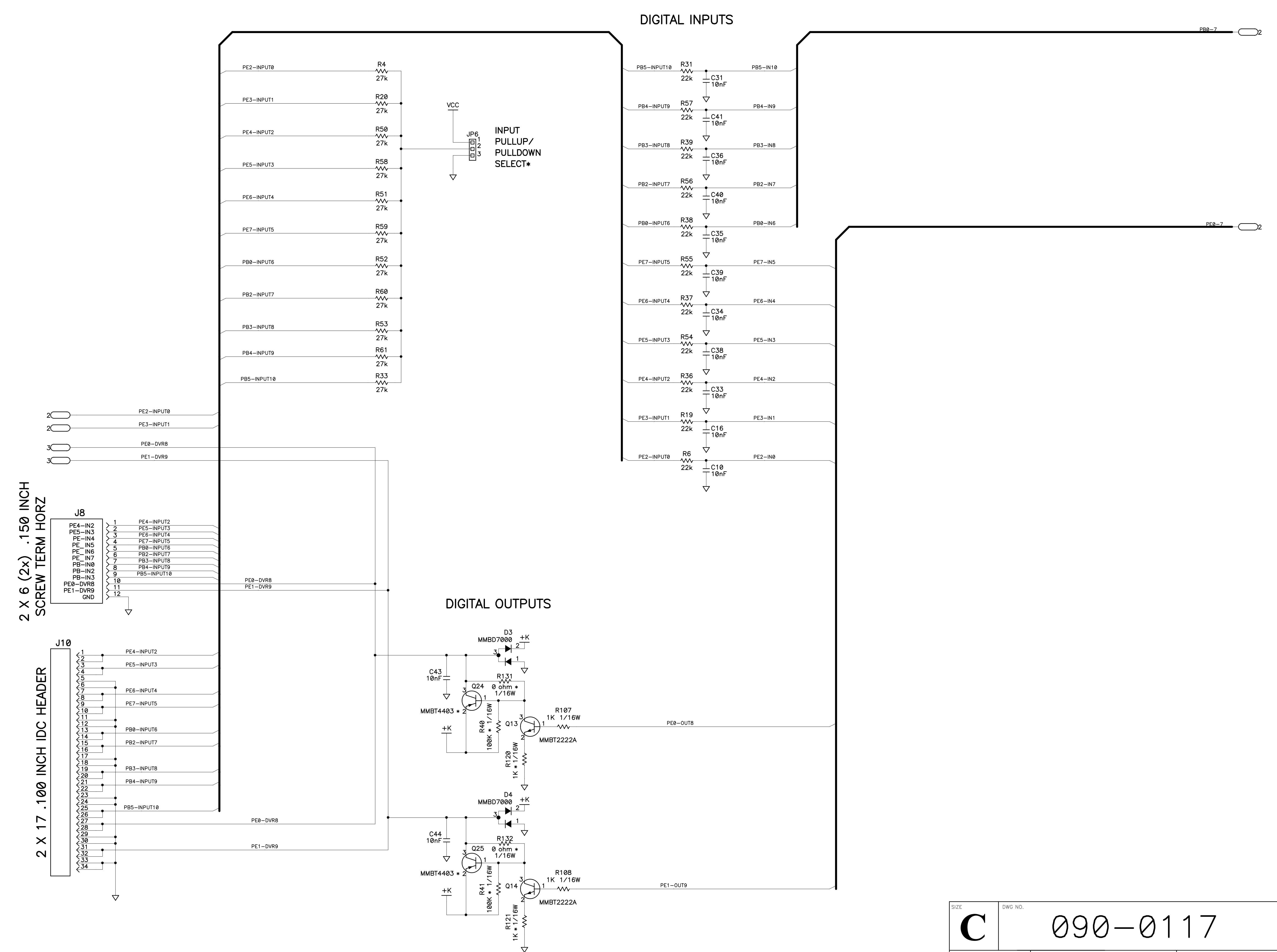
090-0128 Programming Cable Schematic



SRAM*
U6: OPTIONAL 512K, 256K, OR 128K SRAM
FLASH*
U11: OPTIONAL 512K, 256K, OR 128K FLASH
PROCESSOR
RABBIT 2000
SRAM SELECT*
FLASH SELECT
BANK SELECT
POWER TO VRAM SWITCH
CS Control
RESET GENERATOR
RABBIT PROGRAMMING PORT
ETHERNET
RS232
RS485
2 X 17 .100 INCH IDC HEADER
2 X 6 (2x) .150 INCH SCREW TERM HORZ
PLACE WITHIN 200 MILS OF PIN 50
Filter Ground Noise Comming From PC
TIE UP FOR 16b MODE
TIE DOWN FOR 8b MODE
IN THE ABSENCE OF CHASIS GROUND, TIE CHASIS GND TO DIGITAL GND
PLACE THIS CAP BETWEEN PINS 57 & 52 OF U11 WITH MINIMUM TRACE LENGTH
PLACE THIS CAP BETWEEN PINS 47 & 44 OF U11 WITH MINIMUM TRACE LENGTH
COPYRIGHT 2001, Z-WORLD, INC.
SIZE C
DWG NO. 090-0117
SCALE NONE
REV LTR A
SHEET 2 OF 5



SIZE	DWG NO.		
C	090-0117		
SCALE NONE	REV LTR A	SHEET 3	OF 5



SIZE	DWG NO.		
C	090-0117		
SCALE NONE	REV LTR A	SHEET 4	OF 5

STUFFING TABLE

	CIRCUIT	PART	OPTION INSTALLED	OPTION NOT INSTALLED
UNINSTALLED COMPONENTS	ADC INPUTS AIN_9 & AIN_10	R5		NOT INSTALLED
		R78		NOT INSTALLED
	CS CONTROL	R99		NOT INSTALLED
		R102		NOT INSTALLED
	ETHER-NET	R154		NOT INSTALLED
SRAM	SRAM DENSITY	U3	128K SRAM	256K/512K SRAM
	SRAM SELECT	JP5	ZERO ohm ACROSS PINS 1-2	ZERO ohm ACROSS PINS 2-3
FLASH	FLASH DENSITY	U5	128K/256K FLASH	512K FLASH
	FLASH SELECT	JP4	ZERO ohm ACROSS PINS 1-2	ZERO ohm ACROSS PINS 2-3
FLASH BANK SELECT	FLASH DENSITY	U5	256K SRAM	128K/512K SRAM
	BANK SELECT	JP2	DISABLED ZERO ohm ACROSS PINS 1-2	DISABLED ZERO ohm ACROSS PINS 1-2
			ENABLED ZERO ohm ACROSS PINS 2-3	
DACs	DAC_0	U7	AD5320	NOT INSTALLED
		C53	INSTALLED	NOT INSTALLED
	DAC_1	U2	AD5320	NOT INSTALLED
		C75	INSTALLED	NOT INSTALLED
	DAC POWER	JP3	4.096V ZERO ohm ACROSS PINS 2-3	5.0V ZERO ohm ACROSS PINS 1-2
	4.096V REF	U10	LM4040BIM3-4.1	NOT INSTALLED
		R81	NOT INSTALLED	453 ohms 1%
ADC	10 BIT ADC	U9	TLC1543	NOT INSTALLED
	12 BIT ADC	U9	TLC2543	NOT INSTALLED
ETHERNET	ETHERNET	U13	RTL8019AS	NOT INSTALLED
		Q2	2N7002	NOT INSTALLED
		J6	RJ45	NOT INSTALLED
		Y3	20MHz XTAL	NOT INSTALLED
		R30	22K	NOT INSTALLED
		R82	11K	NOT INSTALLED
		R88	1.02M	NOT INSTALLED
		R89	ZERO ohm	NOT INSTALLED
		R85	200 ohms	NOT INSTALLED
		R94	330 ohms	NOT INSTALLED
		R97	330 ohms	NOT INSTALLED
		R80	4.7 ohms	NOT INSTALLED
		R95	4.7 ohms	NOT INSTALLED
		R153	27K	NOT INSTALLED
		C67	33pf	NOT INSTALLED
		C65	33pf	NOT INSTALLED
		C17	10nf	NOT INSTALLED
		C20	10nf	NOT INSTALLED
		C25	10nf 1KV	NOT INSTALLED
		C23	10nf 1KV	NOT INSTALLED
		C63	56nf	NOT INSTALLED
		C68	56nf	NOT INSTALLED
		DS2	GRN LED	NOT INSTALLED
		DS3	ORG LED	NOT INSTALLED

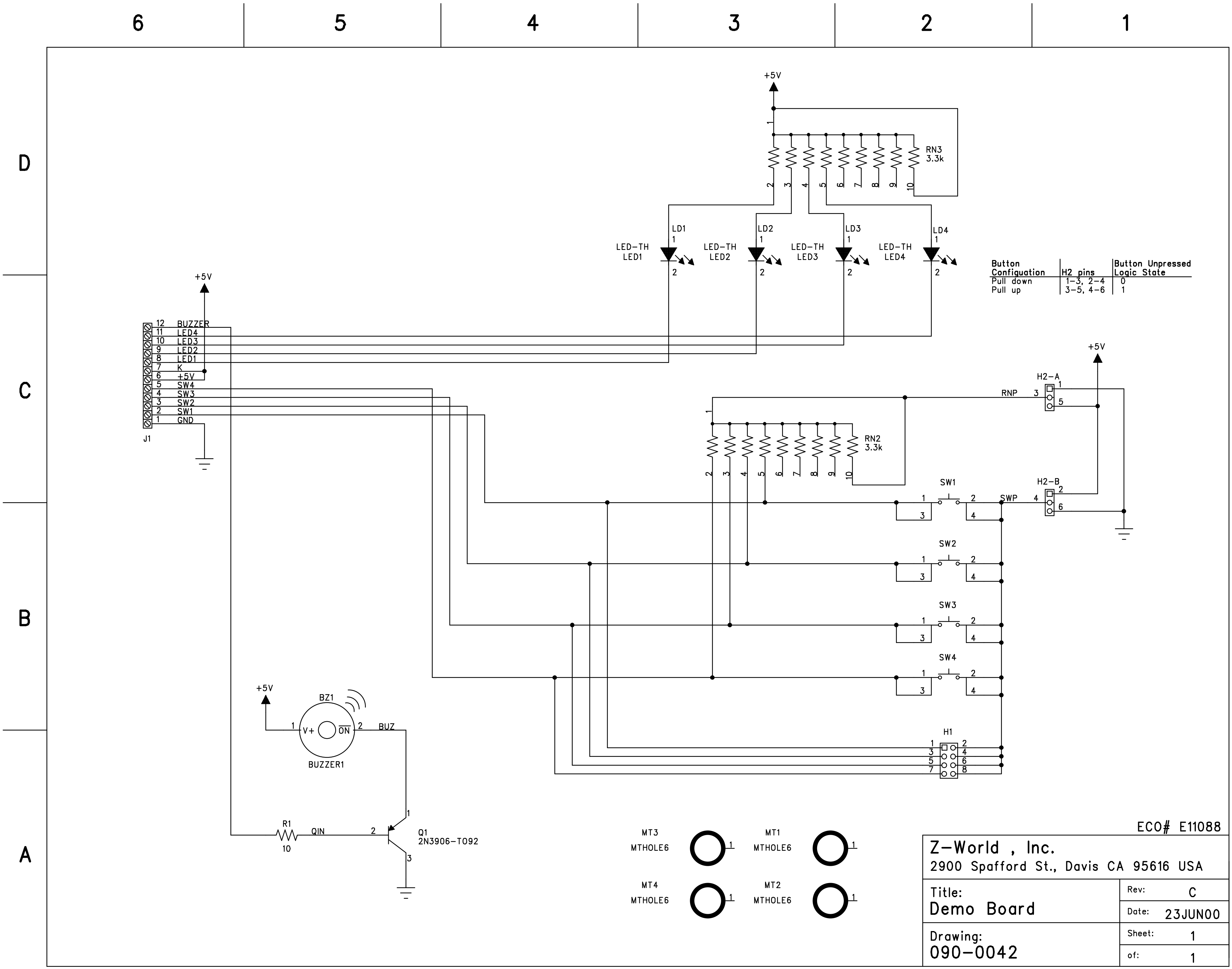
STUFFING TABLE

	CIRCUIT	PART	OPTION INSTALLED	OPTION NOT INSTALLED
RELAY	RELAY	K1	TQ2SA-4.5V	NOT INSTALLED
	RELAY SUPPORT	Q11	2N7002	NOT INSTALLED
		D13	MMBD7000	NOT INSTALLED
		R117	47 ohms 1/4W	NOT INSTALLED
		R106	47 ohms 1/4W	NOT INSTALLED
		C42	100nf 250V	NOT INSTALLED
		C37	100nf 250V	NOT INSTALLED
OUTPUT PULL-UP/DOWN	OUTPUTS PULLED UP TO +K	R35	ZERO ohm	
		R32		NOT INSTALLED
		R34		NOT INSTALLED
	OUTPUTS PULLED UP TO +5V	R35		NOT INSTALLED
		R32	ZERO ohm	
		R34		NOT INSTALLED
	OUTPUTS PULLED DOWN TO GROUND	R34		NOT INSTALLED
		R32		NOT INSTALLED
		R34	ZERO ohm	
OUTPUT DRIVERS 0-9			CURRENT SINK	CURRENT SOURCE
	DRIVER 0	Q26	NOT INSTALLED	MMBT4403
		R122	ZERO ohm	1K
		R133	ZERO ohm	NOT INSTALLED
		R42	NOT INSTALLED	100K
	DRIVER 1	Q27	NOT INSTALLED	MMBT4403
		R123	ZERO ohm	1K
		R134	ZERO ohm	NOT INSTALLED
		R43	NOT INSTALLED	100K
	DRIVER 2	Q28	NOT INSTALLED	MMBT4403
		R124	ZERO ohm	1K
		R135	ZERO ohm	NOT INSTALLED
		R44	NOT INSTALLED	100K
	DRIVER 3	Q29	NOT INSTALLED	MMBT4403
		R125	ZERO ohm	1K
		R136	ZERO ohm	NOT INSTALLED
		R45	NOT INSTALLED	100K
	DRIVER 4	Q30	NOT INSTALLED	MMBT4403
		R126	ZERO ohm	1K
		R137	ZERO ohm	NOT INSTALLED
		R46	NOT INSTALLED	100K
	DRIVER 5	Q31	NOT INSTALLED	MMBT4403
		R127	ZERO ohm	1K
		R138	ZERO ohm	NOT INSTALLED
		R47	NOT INSTALLED	100K
	DRIVER 6	Q32	NOT INSTALLED	MMBT4403
		R128	ZERO ohm	1K
		R139	ZERO ohm	NOT INSTALLED
		R48	NOT INSTALLED	100K
	DRIVER 7	Q33	NOT INSTALLED	MMBT4403
		R129	ZERO ohm	1K
		R140	ZERO ohm	NOT INSTALLED
		R49	NOT INSTALLED	100K
	DRIVER 8	Q24	NOT INSTALLED	MMBT4403
		R120	ZERO ohm	1K
		R131	ZERO ohm	NOT INSTALLED
		R40	NOT INSTALLED	100K
	DRIVER 9	Q25	NOT INSTALLED	MMBT4403
		R121	ZERO ohm	1K
		R132	ZERO ohm	NOT INSTALLED
		R41	NOT INSTALLED	100K
INPUT PULL-UP/DOWN	INPUTS PULLED UP TO +5V	JP6	ZERO ohm ACROSS PINS 1-2	
	INPUTS PULLED DOWN TO GROUND	JP6	ZERO ohm ACROSS PINS 2-3	

BL20XX OPTION TABLE

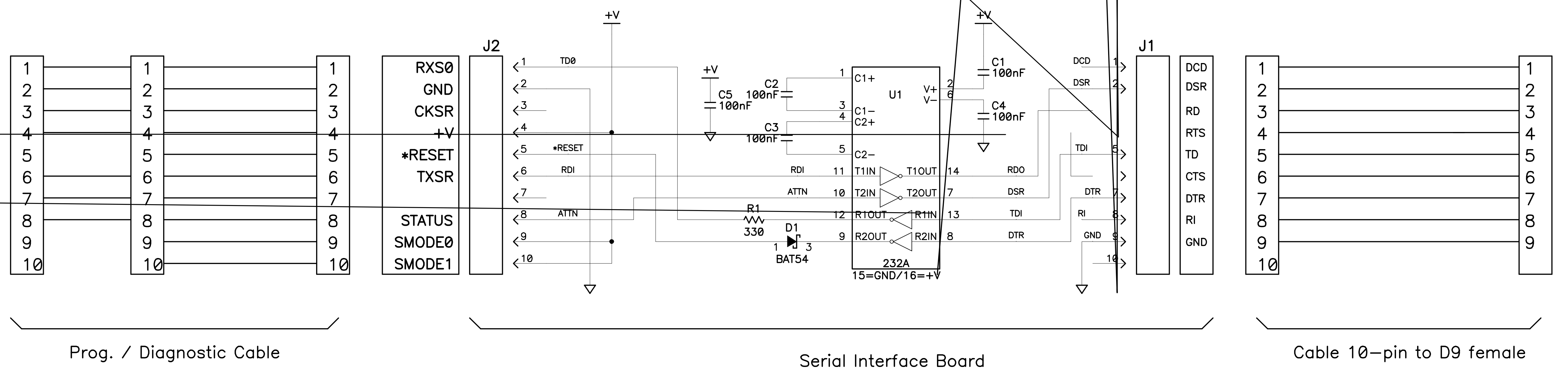
CIRCUIT OPTION	BL2000	BL2010	BL2020	BL2030	BL2040
FLASH DENSITY	256K	256K	256K	256K	256K
FLASH BANK SELECT	DISABLED	DISABLED	DISABLED	DISABLED	DISABLED
SRAM DENSITY	128K	128K	128K	128K	128K
DAC_0	INSTALLED	NOT INSTALLED	INSTALLED	NOT INSTALLED	NOT INSTALLED
DAC_1	INSTALLED	NOT INSTALLED	INSTALLED	NOT INSTALLED	NOT INSTALLED
DAC POWER	4.096V	4.096V	4.096V	4.096V	NOT INSTALLED
4.096V REF	INSTALLED	NOT INSTALLED	INSTALLED	NOT INSTALLED	NOT INSTALLED
8 BIT ADC	NOT INSTALLED	INSTALLED	NOT INSTALLED	INSTALLED	NOT INSTALLED
12 BIT ADC	INSTALLED	NOT INSTALLED	INSTALLED	NOT INSTALLED	NOT INSTALLED
ETHER-NET	INSTALLED	INSTALLED	NOT INSTALLED	NOT INSTALLED	NOT INSTALLED
OUTPUT CURRENT 0	SINKING	SINKING	SINKING	SINKING	SINKING
OUTPUT CURRENT 1	SINKING	SINKING	SINKING	SINKING	SINKING
OUTPUT CURRENT 2	SINKING	SINKING	SINKING	SINKING	SINKING
OUTPUT CURRENT 3	SINKING	SINKING	SINKING	SINKING	SINKING
OUTPUT CURRENT 4	SINKING	SINKING	SINKING	SINKING	SINKING
OUTPUT CURRENT 5	SINKING	SINKING	SINKING	SINKING	SINKING
OUTPUT CURRENT 6	SINKING	SINKING	SINKING	SINKING	SINKING
OUTPUT CURRENT 7	SINKING	SINKING	SINKING	SINKING	SINKING
OUTPUT CURRENT 8	SINKING	SINKING	SINKING	SINKING	SINKING
OUTPUT CURRENT 9	SINKING	SINKING	SINKING	SINKING	SINKING
OUTPUT PULL UP/DOWN	PULLED TO +5V	PULLED TO +5V	PULLED TO +5V	PULLED TO +5V	PULLED TO +5V
INPUT PULL UP/DOWN	PULLED TO +5V	PULLED TO +5V	PULLED TO +5V	PULLED TO +5V	PULLED TO +5V

SIZE	DWG NO.	
C	090-0117	
SCALE NONE	REV LTR A	SHEET 5 OF 5



6 5 4 3 2 1
D C B A
+5V
RN3
3.3k
LD1 LD2 LD3 LD4
LED-TH LED1
LED-TH LED2
LED-TH LED3
LED-TH LED4
Button Configuation | H2 pins | Button Unpressed Logic State
Pull down | 1-3, 2-4 | 0
Pull up | 3-5, 4-6 | 1
12 BUZZER
11 LED4
10 LED3
9 LED2
8 LED1
7 K
6 +5V
5 SW4
4 SW3
3 SW2
2 SW1
1 GND
J1
H2-A
RNP
H2-B
SWP
RN2
3.3k
SW1 SW2 SW3 SW4
H1
BZ1
V+
ON
BUZ
BUZZER1
R1
10
QIN
Q1
2N3906-TO92
MT3 MTHOLE6
MT1 MTHOLE6
MT4 MTHOLE6
MT2 MTHOLE6
ECO# E11088
Z-World , Inc.
2900 Spafford St., Davis CA 95616 USA
Title:
Demo Board
Rev: C
Date: 23JUN00
Drawing:
090-0042
Sheet: 1
of: 1

REVISION HISTORY			REVISION APPROVAL			
REV	ECO	DESCRIPTION	PROJECT ENGINEER	APPROVAL DATE	DOCUMENT CONTROL	APPROVAL DATE
A	E11523	INITIAL RELEASE OF SCHEMATIC				



NOTES: UNLESS OTHERWISE SPECIFIED;

1. ALL RESISTOR VALUES ARE IN OHMS, 1/10W, 5%
2. ALL CAPACITORS ARE 50VDC OR HIGHER.
3. THE ORIGINATION SOURCE OF A VOLTAGE IS REPRESENTED BY (VCC), AND ALL REFERENCES TO THAT VOLTAGE ARE REPRESENTED BY (VCC).

APPEND THE FOLLOWING DOCUMENTS WHEN CHANGING THIS DOCUMENT:	DRAWING CONTENT:		TITLE
	DRAWN BY: (INITIAL RELEASE) E. PEAK	14MAY01	SCHEMATIC DIAGRAM RABBIT PROG. CABLE
	REVISED BY:		
	APPROVALS: INITIAL RELEASE		
	PROJECT ENGINEER:		SIZE B — DWG NO. 090–0128
	ENGINEERING MANAGER:		
	SIGNATURES	DATE	SCALE NONE — RELEASE DATE — SHEET 1 OF 1

Z-WORLD
2900 SPAFFORD ST.
DAVIS, CA 95616
530 - 757 - 4616