



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



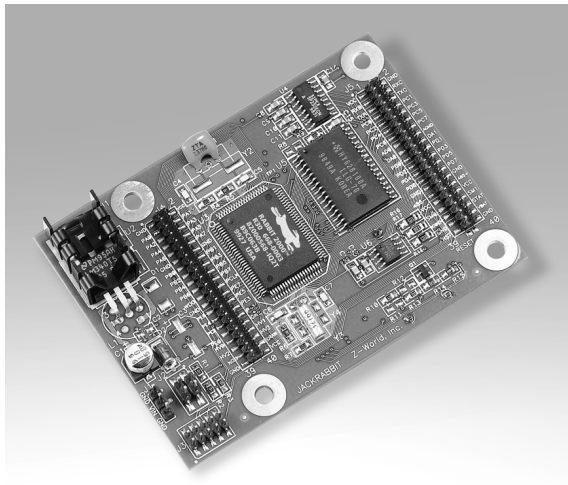
Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China





Rabbit[®] 2000
Microprocessor Development Kit
Getting Started Manual

019-0068 • 050515-F

Rabbit 2000 Development Kit Getting Started Manual

Part Number 019-0068 • 050515-F • Printed in U.S.A.

©2000–2005 Z-World Inc. • All rights reserved.

Z-World reserves the right to make changes and improvements to its products without providing notice.

Trademarks

Rabbit is a registered trademark of Rabbit Semiconductor.

Rabbit 2000 is a trademark of Rabbit Semiconductor.

Z-World and Dynamic C are registered trademarks of Z-World Inc.

Rabbit Semiconductor

2932 Spafford Street
Davis, California 95616-6800
USA

Telephone: (530) 757-8400
Fax: (530) 753-8402

www.rabbitsemiconductor.com

TABLE OF CONTENTS

Chapter 1. Introduction	1
1.1 Development Kit Contents	1
1.2 Development Software	2
1.3 How to Use This Manual	2
1.3.1 Additional Product Information	2
1.3.2 Additional Reference Information	2
1.3.3 Using Online Documentation.....	3
1.4 CE Compliance	4
1.4.1 Spectrum Spreader	5
1.4.2 Design Guidelines	5
1.4.3 Interfacing the BL1800 to Other Devices	6
Chapter 2. Detailed Installation Instructions	7
2.1 Software Installation	7
2.2 Prototyping Board.....	8
2.2.1 Prototyping Board Features	9
2.3 Development Hardware Connections	10
2.3.1 Attach BL1810 to Prototyping Board	11
2.3.2 Connect Programming Cable	12
2.3.3 Connect Power	13
2.4 Start Dynamic C.....	14
2.5 Run a Sample Program	14
2.5.1 Troubleshooting	14
2.6 Where Do I Go From Here?	15
2.6.1 Technical Support	15
Chapter 3. Sample Programs	17
3.1 Running Sample Program DEMOJR1.C	18
3.1.1 Other Sample Programs Illustrating Digital I/O	22
3.1.2 RS-232 Serial Communication Sample Programs	24
3.1.3 RS-485 Serial Communication Sample Program.....	25
3.2 Cooperative Multitasking	26
3.2.1 Advantages of Cooperative Multitasking	28
3.3 Switching Between Program Mode and Run Mode	29
3.3.1 Detailed Instructions: Changing from Program Mode to Run Mode.....	29
3.3.2 Detailed Instructions: Changing from Run Mode to Program Mode.....	29
Chapter 4. Software Reference	31
4.1 An Overview of Dynamic C	31
4.1.1 Upgrading Dynamic C	33
4.1.2 Add-On Modules.....	33
4.2 BL1810 Function Calls.....	34
4.2.1 I/O Drivers	34
4.2.2 Serial Communication Drivers.....	38

Appendix A. Reference Information	39
A.1 Electrical and Mechanical Specifications.....	40
A.2 Header Pinout.....	41
A.3 Jumper Configurations	42
A.4 Use of Rabbit 2000 Parallel Ports	44
Notice to Users	45
Index	47
Schematics	49

1. INTRODUCTION

The Rabbit[®] 2000 is a new and powerful microprocessor. Both hardware and software design are easy with the Rabbit.

This Development Kit has the essentials that you need to design your own a microprocessor-based system, and includes a complete software development system (Dynamic C). This kit contains a powerful single-board computer (the BL1810). With this kit you will be able to write and test complex software. You will be able to prototype circuits that interface to a Rabbit 2000 microprocessor.

1.1 Development Kit Contents

The Rabbit 2000 Development Kit contains the following items:

- BL1810 single-board computer.
- Prototyping Board.
- AC adapter, 12 V DC, 1 A. (Included only with Development Kits sold for the North American market. A header plug leading to bare leads is provided to allow overseas users to connect their own power supply with a DC output of 7.5–25 V.)

NOTE: The linear voltage regulator becomes rather hot for voltages above 15 V.

- 10-pin header to DE9 programming cable with integrated level-matching circuitry.
- *Dynamic C* CD-ROM, with complete product documentation on disk.
- This *Getting Started* manual.
- A bag of accessory parts for use on the Prototyping Board.
- Screwdriver.
- *Rabbit 2000 Processor Easy Reference* poster.
- Registration card.

1.2 Development Software

The BL1810 in the Development Kit uses the Dynamic C development environment for rapid creation and debugging of runtime applications. Dynamic C provides a complete development environment with integrated editor, compiler and source-level debugger. It interfaces directly with the target system, eliminating the need for complex and unreliable in-circuit emulators.

Dynamic C must be installed on a Windows workstation with at least one free serial USB or COM port for communication with the target system. See Chapter 2., “Detailed Installation Instructions,” for complete information on installing Dynamic C.

1.3 How to Use This Manual

This *Getting Started* manual is intended to give users a quick but solid start with the Rabbit 2000 microprocessor. It does not contain detailed information on the hardware capabilities or the Dynamic C development environment. Most users will want more detailed information on some or all of these topics in order to put the Rabbit 2000 microprocessor to effective use.

1.3.1 Additional Product Information

Detailed information about the BL1810 will be found in the *Jackrabbit (BL1800) User’s Manual*, provided on the accompanying CD-ROM in both HTML and Adobe PDF format.

TIP: We recommend that anyone not thoroughly familiar with Z-World embedded control systems at least read through the rest of this manual to gain the necessary familiarity to make use of the more advanced information.

1.3.2 Additional Reference Information

In addition to the product-specific information contained in the *Jackrabbit (BL1800) User’s Manual*, several higher level reference manuals are provided in HTML and PDF form on the accompanying CD-ROM. Advanced users will find these references valuable in developing systems based on the BL1810:

- *Dynamic C User’s Manual*
- *Dynamic C Function Reference Manual*
- *Rabbit 2000 Microprocessor User’s Manual*

1.3.3 Using Online Documentation

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

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

NOTE: The most current version of Adobe Acrobat Reader can always be downloaded from Adobe's web site at <http://www.adobe.com>. We recommend that you use version 4.0 or later.

Providing this documentation in electronic form saves an enormous amount of paper by not printing copies of manuals that users don't need.

Finding Online Documents

The online documentation is installed along with Dynamic C, and an icon for the documentation menu is placed on the workstation's desktop. Double-click this icon to reach the menu. If the icon is missing, create a new desktop icon that points to **default.htm** in the **docs** folder, found in the Dynamic C installation folder.

The latest versions of all documents are always available for free, unregistered download from our web sites as well.

Printing Electronic Manuals

We recognize that many users prefer printed manuals for some uses. Users can easily print all or parts of those manuals provided in electronic form. The following guidelines may be helpful:

- Print from the Adobe PDF versions of the files, not the HTML versions.
- If your printer supports duplex printing, print pages double-sided.
- If you do not have a suitable printer or do not want to print the manual yourself, most retail copy shops (e.g. Kinkos, AlphaGraphics, etc.) will print the manual from the PDF file and bind it for a reasonable charge—about what we would have to charge for a printed and bound manual.

1.4 CE Compliance

Equipment is generally divided into two classes.

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

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

The BL1800 single-board computer has been tested and was found to be in conformity with the following applicable immunity and emission standards. The BL1810 and BL1820 single-board computers are also CE qualified as they are sub-versions of the BL1800 single-board computer. Boards that are CE-compliant have the CE mark.



NOTE: Earlier versions of the BL1800 sold before 2002 that do not have the CE mark are *not* CE-complaint.

Immunity

The BL1800 series of single-board computers meets the following EN55024/1998 immunity standards.

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

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

Emissions

The BL1800 series of single-board computers meets the following emission standards with the Rabbit 2000 spectrum spreader turned on and set to the normal mode. The spectrum spreader is only available with Rev. C or higher of the Rabbit 2000 microprocessor. This microprocessor is used in all BL1800 series boards that carry the CE mark.

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

In order for the BL1800s to meet these EN55022:1998 Class B standards, you must add ferrite absorbers to the serial I/O cables used for RS-232 and RS-485 serial communication. Depending on your application, you may need to add ferrite absorbers to the digital

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

NOTE: If no ferrite absorbers are fitted, the BL1800s will still meet EN55022:1998 Class A requirements as long as the spectrum spreader is turned on.

The spectrum spreader is on by default for the Jackrabbit model BL1810 included with the Rabbit 2000 Development Kit.

1.4.1 Spectrum Spreader

BL1800s that carry the CE mark have a Rabbit 2000 microprocessor that features a spectrum spreader, which helps to mitigate EMI problems. By default, the spectrum spreader is on automatically for BL1810 boards that carry the CE mark when used with Dynamic C 7.32 or later versions so as to maintain CE compliance, but the spectrum spreader may also be turned off or set to a stronger setting. The means for doing so is through a simple change to the following BIOS line.

```
#define ENABLE_SPREADER 1 // Set to 0 to disable spectrum spreader
                          // 1 to enable normal spreading, or
                          // 2 to enable strong spreading.
```

NOTE: The strong spectrum-spreading setting is not needed for any BL1810.

There is no spectrum spreader functionality for BL1800s that do not carry the CE mark or when using any BL1800 with a version of Dynamic C prior to 7.30.

1.4.2 Design Guidelines

Note the following requirements for incorporating the BL1800 series of single-board computers into your application to comply with CE requirements.

General

- The power supply provided with the Development Kit is for development purposes only. It is the customer's responsibility to provide a CE-compliant power supply for the end-product application.
- When connecting the BL1800 single-board computer to outdoor cables, the customer is responsible for providing CE-approved surge/lightning protection.
- Rabbit Semiconductor recommends placing digital I/O or analog cables that are 3 m or longer in a metal conduit to assist in maintaining CE compliance and to conform to good cable design practices. Rabbit Semiconductor also recommends using properly shielded I/O cables in noisy electromagnetic environments.
- When installing or servicing the BL1800, it is the responsibility of the end-user to use proper ESD precautions to prevent ESD damage to the BL1800.

Safety

- For personal safety, all inputs and outputs to and from the BL1800 series of single-board computers must not be connected to voltages exceeding SELV levels (42.4 V AC peak, or 60 V DC). Damage to the Rabbit 2000 microprocessor may result if voltages outside the design range of 0 V to 5.5 V DC are applied directly to any of its digital inputs.
- The lithium backup battery circuit on the BL1800 single-board computer has been designed to protect the battery from hazardous conditions such as reverse charging and excessive current flows. Do not disable the safety features of the design.

1.4.3 Interfacing the BL1800 to Other Devices

Since the BL1800 series of single-board computers is designed to be connected to other devices, good EMC practices should be followed to ensure compliance. CE compliance is ultimately the responsibility of the integrator. Additional information, tips, and technical assistance are available from your authorized Rabbit Semiconductor distributor, and are also available on the Z-World Web site at www.zworld.com.



2. DETAILED INSTALLATION INSTRUCTIONS

Chapter 2 contains detailed instructions for installing the software on your PC and for connecting the BL1810 to your PC in order to run sample programs.

2.1 Software Installation

You will need approximately 200 megabytes of free space on your hard disk for a complete installation. The software can be installed on your C drive or any other convenient drive.

Insert the Dynamic C CD-ROM in the drive on your PC. If autorun is enabled, the CD installation will begin automatically.

If autorun is disabled or the installation otherwise does not start, use the Windows **Start | Run** menu or Windows Disk Explorer to launch **SETUP.EXE** from the root folder of the CD-ROM.

The installation program will guide you through the installation process.

2.2 Prototyping Board

The Prototyping Board included in the Development Kit makes it easy to connect a BL1810 to a power supply and a PC workstation for development. It also provides some basic I/O peripherals (switches and LEDs), as well as a prototyping area for more advanced hardware development.

For the most basic level of evaluation and development, the Prototyping Board can be used without modification.

As you progress to more sophisticated experimentation and hardware development, modifications and additions can be made to the board without modifying or damaging the BL1810 itself.

The Prototyping Board is shown below in Figure 1, with its main features identified.

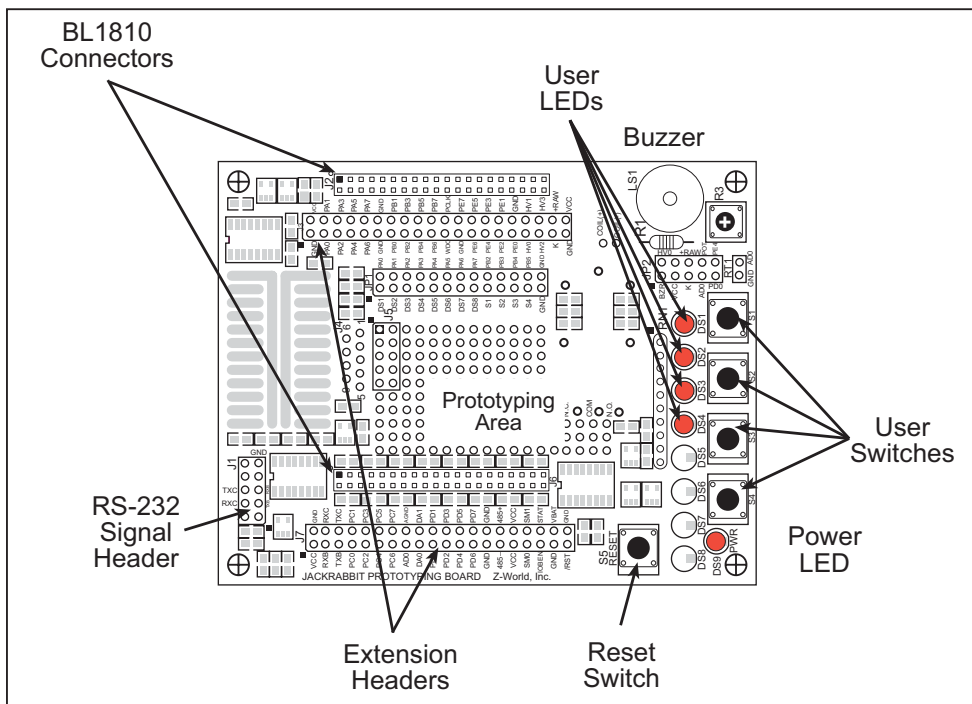


Figure 1. Prototyping Board

2.2.1 Prototyping Board Features

- **Power LED**—The power LED lights whenever power is connected to the Prototyping Board.
- **Reset Switch**—A momentary-contact, normally open switch is connected directly to the BL1810's **/RESET_IN** pin. Pressing the switch forces a hardware reset of the system.
- **I/O Switches and LEDs**—Four momentary-contact, normally open switches are connected to the PB2–PB5 pins of the Rabbit 2000 microprocessor on the BL1810, and may be read as inputs by sample applications.

Four LEDs are connected to the PA1–PA4 pins of the of the Rabbit 2000 microprocessor on the BL1810, and may be driven as output indicators by sample applications.

- **Prototyping Area**—A generous prototyping area has been provided for the installation of through-hole components. Several areas for surface-mount devices are also available. (Note that there are SMT device pads on both top and bottom of the Prototyping Board.)
- **Extension Headers**—The complete pin sets of the BL1810 are duplicated at these two sets of headers. Developers can solder wires directly into the appropriate holes, or, for more flexible development, 40-pin header strips can be soldered into place. See Figure A-1 for the header pinouts.
- **RS-232**—Two 3-wire or one 5-wire RS-232 serial port are available on the Prototyping Board. Refer to the Prototyping Board schematic (090-0088) for additional details.

A 10-pin 0.1-inch spacing header strip is installed at J1 to permit connection of a ribbon cable leading to a standard DE-9 serial connector.

2.3 Development Hardware Connections

There are three steps to connecting the Prototyping Board for use with Dynamic C and the sample programs:

1. Attach the BL1810 to the Prototyping Board.
2. Connect the programming cable between the BL1810 and the workstation PC.
3. Connect the power supply to the BL1810.

2.3.1 Attach BL1810 to Prototyping Board

To attach the BL1810 to the Prototyping Board, turn the BL1810 over so that the battery is facing up. Plug the pins from headers J4 and J5 on the bottom side of the BL1810 into the header sockets at J2 and J6 on the Prototyping Board as indicated in Figure 2.

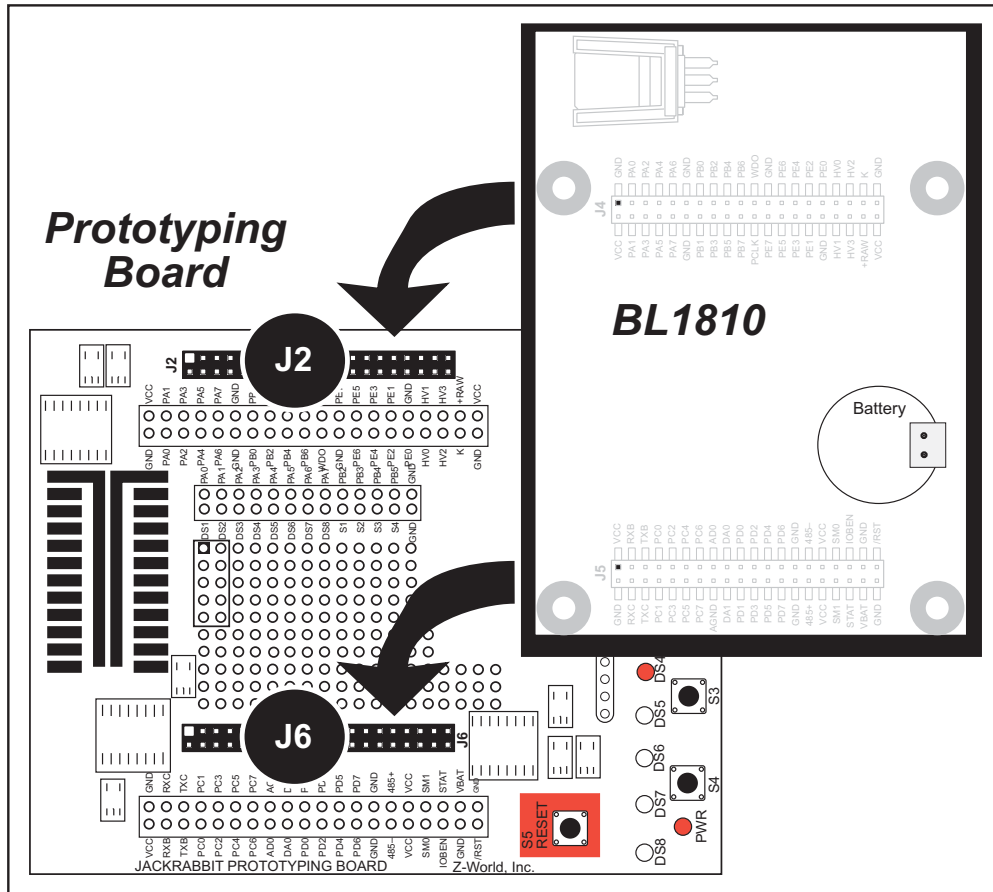


Figure 2. Attach BL1810 to Prototyping Board

NOTE: It is important that you line up the pins on headers J4 and J5 of the BL1810 exactly with the corresponding pins of header sockets J2 and J6 on the Prototyping Board. The header pins may become bent or damaged if the pin alignment is offset, and the BL1810 might not work. Permanent electrical damage to the BL1810 board may also result if a misaligned BL1810 is powered up.

Press the BL1810's pins firmly into the Prototyping Board headers.

2.3.2 Connect Programming Cable

The programming cable connects the BL1810 to the PC running Dynamic C to download programs and to monitor the BL1810 during debugging.

Connect the 10-pin connector of the programming cable labeled **PROG** to header J3 on the BL1810 as shown in Figure 3. Be sure to orient the marked (usually red) edge of the cable towards pin 1 of the connector. (Do not use the **DIAG** connector, which is used for a normal serial connection.)

NOTE: Use only the programming cable that has a red shrink wrap around the RS-232 level converter (Z-World part number 101-0513), which is supplied with the Development Kit. Other Z-World programming cables might not be voltage-compatible or their connector sizes may be different.

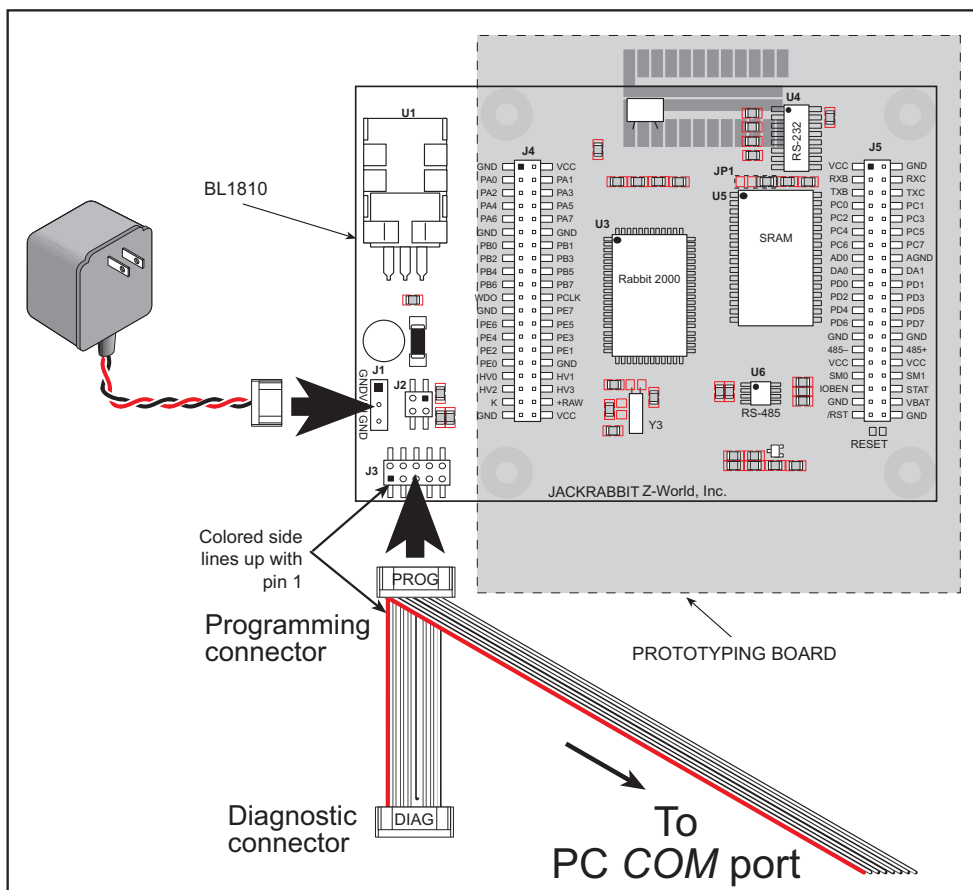


Figure 3. Power and Programming Cable Connections to the BL1810

Connect the other end of the programming cable to a COM port on your PC.

NOTE: Some PCs now come equipped only with a USB port. It may be possible to use an RS-232/USB converter with the programming cable supplied with the Rabbit 2000 Development Kit. An RS-232/USB converter is available through the Z-World Web store.

2.3.3 Connect Power

When all other connections have been made, you can connect power to the BL1810.

Hook up the connector from the wall transformer to header J1 on the BL1810 as shown in Figure 3. The orientation of this connector is not important since the VIN (positive) voltage is the middle pin, and GND is available on both ends of the three-pin header J1.

Plug in the wall transformer. The BL1810 and the Prototyping Board are ready to be used.

NOTE: A **RESET** button is provided on the Prototyping Board (see Figure 3) to allow hardware reset without disconnecting power.

To power down the BL1810, unplug the power connector from J1. You should disconnect power before making any circuit adjustments in the prototyping area, changing any connections to the board, or removing the BL1810 from the Prototyping Board.

2.3.3.1 Overseas Development Kits

Development kits sold outside North America include a header connector that may be connected to 3-pin header J1 on the BL1810. The connector may be attached either way as long as it is not offset to one side. The red and black wires from the connector can then be connected to the positive and negative connections on your power supply. The power supply should deliver 7.5 V–25 V DC at 5 W.

2.4 Start Dynamic C

Once the BL1810 is connected as described in Section 2.3, start Dynamic C by double-clicking on the Dynamic C icon or by double-clicking on `dcrabXXXX.exe` in the Dynamic C root directory, where `XXXX` are version-specific characters.

If you are using a USB port to connect your computer to the BL1810, choose **Options > Project Options** and select “Use USB to Serial Converter” under the **Communications** tab.

2.5 Run 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 shows that the CPU is working.

2.5.1 Troubleshooting

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

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

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

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

If there are any other problems:

- Check to make sure you are using the **PROG** connector, not the **DIAG** connector, on the programming cable.
- Check both ends of the programming cable to ensure that they are firmly plugged into the PC and the programming port on the BL1810.
- Ensure that the BL1810 is firmly and correctly installed in its connectors on the Prototyping Board.
- Select a different COM port within Dynamic C. From the **Options** menu, select **Project Options**, then select **Communications**. Select another COM port from the list, then click OK. Press **<Ctrl-Y>** to force Dynamic C to recompile the BIOS. If Dynamic C still reports it is unable to locate the target system, repeat the above steps until you locate the active COM port.

2.6 Where Do I Go From Here?

If everything appears to be working, we recommend the following sequence of action:

1. Run all of the sample programs described in Chapter 3 to get a basic familiarity with Dynamic C and the BL1810's capabilities.
2. For further development, refer to the *Jackrabbit (BL1800) User's Manual* for details of the board's hardware components.

A documentation icon should have been installed on your workstation's desktop; click on it to reach the documentation menu. You can create a new desktop icon that points to **default.htm** in the **docs** folder in the Dynamic C installation folder.

3. For advanced development topics, refer to the *Dynamic C User's Manual*, also in the online documentation set.

2.6.1 Technical Support

NOTE: If you purchased your Rabbit 2000 Development Kit through a distributor or through a Rabbit Semiconductor partner, contact the distributor or partner first for technical support.

If there are any problems at this point:

- Use the Dynamic C **Help** menu to get further assistance with Dynamic C.
- Check the Z-World/Rabbit Semiconductor Technical Bulletin Board at www.zworld.com/support/bb/.
- Use the Technical Support e-mail form at www.zworld.com/support/questionSubmit.shtml.

3. SAMPLE PROGRAMS

A series of sample programs is provided in the Dynamic C `SAMPLES/JACKRAB` folder. The sample programs are listed in Table 1.

Table 1. BL1810 Sample Programs

DEMOJR1.C
DEMOJR2.C
DEMOJR3.C
JRIOTEST.C
JR_FLOWCONTROL.C
JR_PARITY.C
JRIO_COF.C
LCD_DEMO.C
RABDB01.C
RABDB02.C

The first five sample programs provide a step-by-step introduction to the BL1810 board. Additional sample programs illustrate more advanced topics.

Each sample program has comments that describe the purpose and function of the program. Loading, executing and studying these programs will give you a solid hands-on overview of the BL1810's capabilities, as well as a quick start with Dynamic C as an application development tool. These programs are intended to serve as tutorials, but then can also be used as starting points or building blocks for your own applications.

NOTE: It is assumed in this section that you have at least an elementary grasp of ANSI C. If you do not, see the introductory pages of the *Dynamic C User's Manual* for a suggested reading list.

Before running any of these sample program, make sure that your BL1810 is connected to the Prototyping Board and to your PC as described in Section 2.3, "Development Hardware Connections."

To run a sample program, open it with the **File** menu (if it is not already open), then compile and run it by pressing **F9** or by selecting **Run** in the **Run** menu.

3.1 Running Sample Program DEMOJR1.C

This sample program can be used to illustrate some of the functions of Dynamic C.

First, open the file `DEMOJR1.C`, which is in the `SAMPLES/JACKRAB` folder. The program will appear in a window, as shown in Figure 4 below (minus some comments). Use the mouse to place the cursor on the function name `WrPortI` in the program and type `<Ctrl-H>`. This will bring up a documentation box for the function `WrPortI`. In general, you can do this with all functions in Dynamic C libraries, including libraries you write yourself. Close the documentation box and continue.

```
main() {  
    WrPortI (SPCR, NULL, 0x84);  
    WrPortI (PADR, &PADRShadow, 0xff);  
  
    while(1) {  
        BitWrPortI (PADR, &PADRShadow, 1, 2);  
        for(j=0; j<25000; j++);  
        BitWrPortI (PADR, &PADRShadow, 0, 2);  
        for(j=0; j<1000; j++);  
    } // end while(1)  
} // end of main
```

C programs begin with main

NULL is a macro for a zero pointer

write to SPCR register to initialize parallel port A

Write all 1's to port A to turn off all LEDs

Start a loop

Set bit 2 to a "1" LED DS3 off.

Time delay by counting to 25,000.

Set bit 2 to a "0" turning LED DS3 on

Count to 1000 for a shorter time delay

End of the endless loop

Note: See *Rabbit 2000 Microprocessor User's Manual* (Software Chapter) for details on the routines that read and write I/O ports.

Figure 4. Sample Program DEMOJR1.C

To run the program `DEMOJR1.C`, load it with the **File > Open** menu, compile it using the **Compile** menu, and then run it by selecting **Run** in the **Run** menu. The LED on the Development Board should start flashing if everything went well. If this doesn't work review the following points.

- The target should be ready, which is indicated by the message "BIOS successfully compiled..." If you did not receive this message or you get a communication error, recompile the BIOS by typing `<Ctrl-Y>` or select **Recompile BIOS** from the **Compile** menu.
- A message reports that "No Rabbit processor detected" in cases where the BL1810 and Prototyping Board are not connected together, the wall transformer is not connected, or is not plugged in. (The red power LED lights whenever power is connected.)

- The programming cable must be connected to the BL1810. (The colored wire on the programming cable is closest to pin 1 on header J3 on the BL1810, as shown in Figure 3.) The other end of the programming cable must be connected to the PC serial port. The COM port specified in the Dynamic C **Options** menu must be the same as the one the programming cable is connected to.
- To check if you have the correct serial port, select **Compile**, then **Compile BIOS**, or type **<Ctrl-Y>**. If the “BIOS successfully compiled ...” message does not display, try a different serial port using the Dynamic C **Options** menu until you find the one you are plugged into. Don’t change anything in this menu except the COM number. The baud rate should be 115,200 bps and the stop bits should be 1.

Single-Stepping

Compile or re-compile `DEMOJR1.C` by clicking the **Compile** button on the task bar. The program will compile and the screen will come up with a highlighted character (green) at the first executable statement of the program. Use the **F8** key to single-step. Each time the **F8** key is pressed, the cursor will advance one statement. When you get to the `for (j=0, j< . . .` statement, it becomes impractical to single-step further because you would have to press **F8** thousands of times. We will use this statement to illustrate watch expressions.

Watch Expression

Type **<Ctrl-W>** or chose **Add/Del Watch Expression** in the **Inspect** menu. A box will come up. Type the lower case letter `j` and click on *add to top* and *close*. Now continue single-stepping with **F8**. Each time you step, the watch expression (`j`) will be evaluated and printed in the watch window. Note how the value of `j` advances when the statement `j++` is executed.

Break Point

Move the cursor to the start of the statement:

```
for (j=0; j<1000; j++);
```

To set a break point on this statement, type **F2** or select **Breakpoint** from the **Run** menu. A red highlight will appear on the first character of the statement. To get the program running at full speed, type **F9** or select **Run** on the **Run** menu. The program will advance until it hits the break point. Then the break point will start flashing and show both red and green colors. Note that LED DS3 is now solidly turned on. This is because we have passed the statement turning on LED DS3. Note that `j` in the watch window has the value 25000. This is because the loop above terminated when `j` reached 25000.

To remove the break point, type **F2** or select **Toggle Breakpoint** on the **Run** menu. To continue program execution, type **F9** or select **Run** from the **Run** menu. Now the LED should be flashing again since the program is running at full speed.

You can set break points while the program is running by positioning the cursor to a statement and using the **F2** key. If the execution thread hits the break point, a break point will take place. You can toggle the break point off with the **F2** key and continue execution with the **F9** key. Try this a few times to get the feel of things.

Editing the Program

Click on the **Edit** box on the task bar. This will set Dynamic C into the edit mode so that you can change the program. Use the **Save as** choice on the **File** menu to save the file with a new name so as not to change the demo program. Save the file as **MYTEST.C**. Now change the number 25000 in the **for** (. . statement to 10000. Then use the **F9** key to recompile and run the program. The LED will start flashing, but it will flash much faster than before because you have changed the loop counter terminal value from 25000 to 10000.

Watching Variables Dynamically

Go back to edit mode (select edit) and load the program **DEMOJR2.C** using the **File** menu **Open** command. This program is the same as the first program, except that a variable **k** has been added along with a statement to increment **k** each time around the endless loop. The statement:

```
runwatch();
```

has been added. This is a debugging statement that makes it possible to view variables while the program is running.

Use the **F9** key to compile and run **DEMOJR2.C**. Now type **<Ctrl-W>** to open the watch window and add the watch expression **k** to the top of the list of watch expressions. Now type **<Ctrl-U>**. Each time you type **<Ctrl-U>**, you will see the current value of **k**, which is incrementing about 5 times a second.

As an experiment add another expression to the watch window:

```
k*5
```

Then type **<Ctrl-U>** several times to observe the watch expressions **k** and **k*5**.

Summary of Features

So far you have practiced using the following features of Dynamic C.

- Loading, compiling and running a program. When you load a program it appears in an edit window. You can compile by selecting **Compile** on the task bar or from the **Compile** menu. When you compile the program, it is compiled into machine language and downloaded to the target over the serial port. The execution proceeds to the first statement of main where it pauses, waiting for you to command the program to run, which you can do with the **F9** key or by selecting **Run** on the **Run** menu. If want to compile and start the program running with one keystroke, use **F9**, the run command. If the program is not already compiled, the run command will compile it first.
- Single-stepping. This is done with the **F8** key. The **F7** key can also be used for single-stepping. If the **F7** key is used, then descent into subroutines will take place. With the **F8** key the subroutine is executed at full speed when the statement that calls it is stepped over.

- Setting break points. The **F2** key is used to turn on or turn off (toggle) a break point at the cursor position if the program has already been compiled. You can set a break point if the program is paused at a break point. You can also set a break point in a program that is running at full speed. This will cause the program to break if the execution thread hits your break point.
- Watch expressions. A watch expression is a C expression that is evaluated on command in the watch window. An expression is basically any type of C formula that can include operators, variables and function calls, but not statements that require multiple lines such as *for* or *switch*. You can have a list of watch expressions in the watch window. If you are single-stepping, then they are all evaluated on each step. You can also command the watch expression to be evaluated by using the **<Ctrl-U>** command. When a watch expression is evaluated at a break point, it is evaluated as if the statement was at the beginning of the function where you are single-stepping. If your program is running you can also evaluate watch expressions with a **<Ctrl-U>** if your program has a **run-watch()** command that is frequently executed. In this case, only expressions involving global variables can be evaluated, and the expression is evaluated as if it were in a separate function with no local variables.