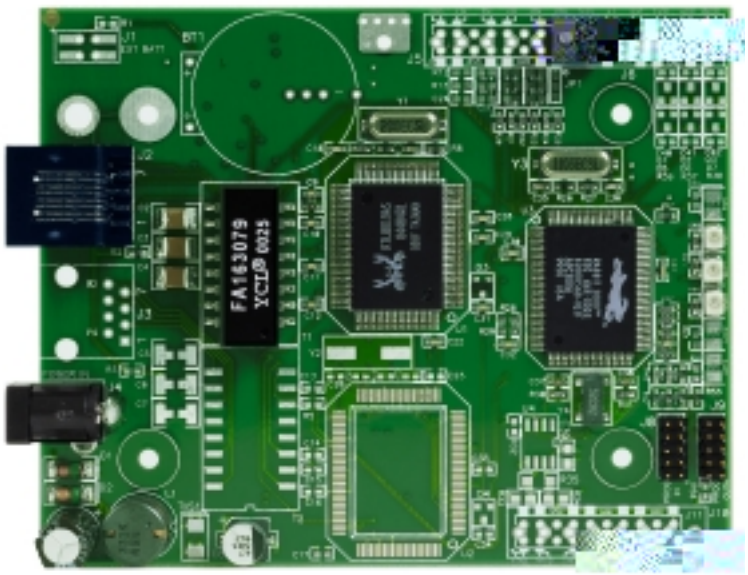


RabbitLink (EG2100)

Network Programming Gateway

User's Manual

010131 - A

RabbitLink (EG2100) User's Manual

Part Number 019-0090 • 010131 - A • Printed in U.S.A.

Copyright

Trademarks

- Dynamic C® is a registered trademark of Z-World, Inc.
- Windows® is a registered trademark of Microsoft Corporation
- Rabbit 2000™ is a trademark of Rabbit Semiconductor

Notice to Users

When a system failure may cause serious consequences, protecting life and property against such consequences with a backup system or safety device is essential. The buyer agrees that protection against consequences resulting from system failure is the buyer's responsibility.

This device is not approved for life-support or medical systems.

All Z-World products are 100 percent functionally tested. Additional testing may include visual quality control inspections or mechanical defects analyzer inspections. Specifications are based on characterization of tested sample units rather than testing over temperature and voltage of each unit. Z-World may qualify components to operate within a range of parameters that is different from the manufacturer's recommended range. This strategy is believed to be more economical and effective. Additional testing or burn-in of an individual unit is available by special arrangement.

Company Address

Z-World, Inc.
2900 Spafford Street
Davis, California 95616-6800
USA

Telephone: (530) 757-3737
Facsimile: (530) 753-5141
Web site: http://www.zworld.com
E-mail: zworld@zworld.com

Rabbit Semiconductor
2932 Spafford Street
Davis, California 95616-6800
USA

Telephone: (530) 757-8400
Facsimile: (530) 757-8402
Web site: http://www.rabbitsemiconductor.com
E-mail: sales@rabbitsemiconductor.com

TABLE OF CONTENTS

1. INTRODUCTION

Chapter 1 introduces the RabbitLink network programming gateway and describes the RabbitLink's features. The tool kit containing the hardware essentials to begin using the RabbitLink is described, and the software highlights are presented.

Rabbit-based embedded systems are normally programmed via a direct connection between a PC and the programming port that is a part of any standard Rabbit-based embedded control system. The RabbitLink network programming gateway provides a connection between an Ethernet-based network and the programming port to allow you to program and debug remotely over the Ethernet or the Internet via TCP/IP.

Figure 1 illustrates the concept of remote network access via the RabbitLink.

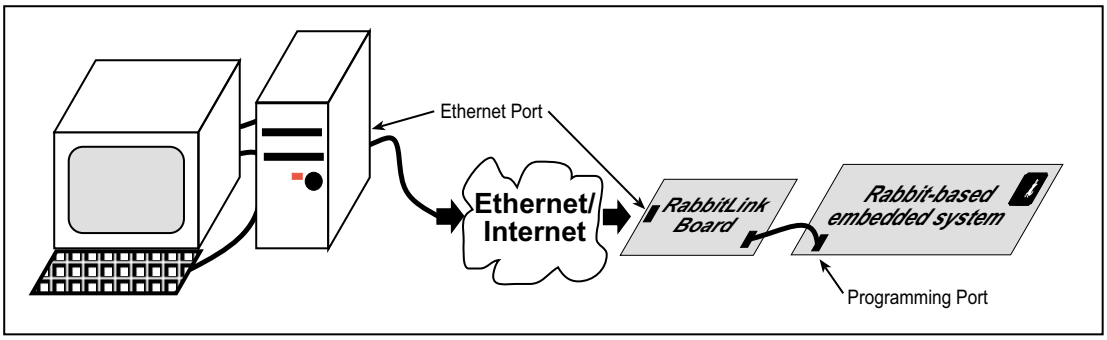


Figure 1. Remote Network Access via RabbitLink Network Gateway

The RabbitLink gateway allows you to program, monitor, and debug your Rabbit-based system from your desk or from anywhere in the world via an Ethernet-based connection. You use Dynamic C (version 7.03 or later) on your PC to communicate with the RabbitLink card via the network. Alternatively, the Rabbit Field Utility (version 2.0 or later) allows you or a customer to download new software via the RabbitLink to a remote system without using Dynamic C.

The RabbitLink gateway can also be used to provide a portal to the Internet for any Rabbit-based embedded system or any other device with a serial port. Firmware resident on the RabbitLink allows your embedded system to send E-mail or generate Web pages via simplified versions of standard network protocols.

1.1 RabbitLink Features

- Rabbit 2000™ microprocessor operating at 22.1 MHz.
- One RJ-45 Ethernet port compliant with IEEE 802.3 standard for 10Base-T Ethernet protocol.
- two serial ports—one to configure or update RabbitLink, one for networking target systems.
- Three status LEDs—download, link, and active.
- Optional plastic enclosure and LED light pipes (enclosure and light pipes are included with the tool kit, and are also sold separately).
- Password protection—password prompt each time a new Dynamic C session is initiated with the RabbitLink.
- 128K static RAM and 512K flash memory.
- Firmware already installed ready to run, easy setup with simple console commands.
- Remote program downloading, debugging, onboard Web server, and SMTP E-mail client.

1.2 Development and Evaluation Tools

1.2.1 Tool Kit

A program download cable included with each RabbitLink is used to connect a target controller to the RabbitLink board. A tool kit is contains the hardware essentials you will need to create and use your own RabbitLink remote programming gateway.

The items in the tool kit and their use are as follows:

- ***RabbitLink (EG2100) User's Manual*** with schematics (this document).
- Programming cable, used to connect your PC serial port to the RabbitLink to set up the serial communication parameters.
- AC adapter, used to power the RabbitLink. An AC adapter is supplied with tool kits sold in the North American market. If you are using another power supply, a minimum of 9 V at 120 mA is recommended.
- Plastic enclosure with four customer-installable light pipes.
- Screwdriver.
- CD with RabbitLink firmware and firmware source code.

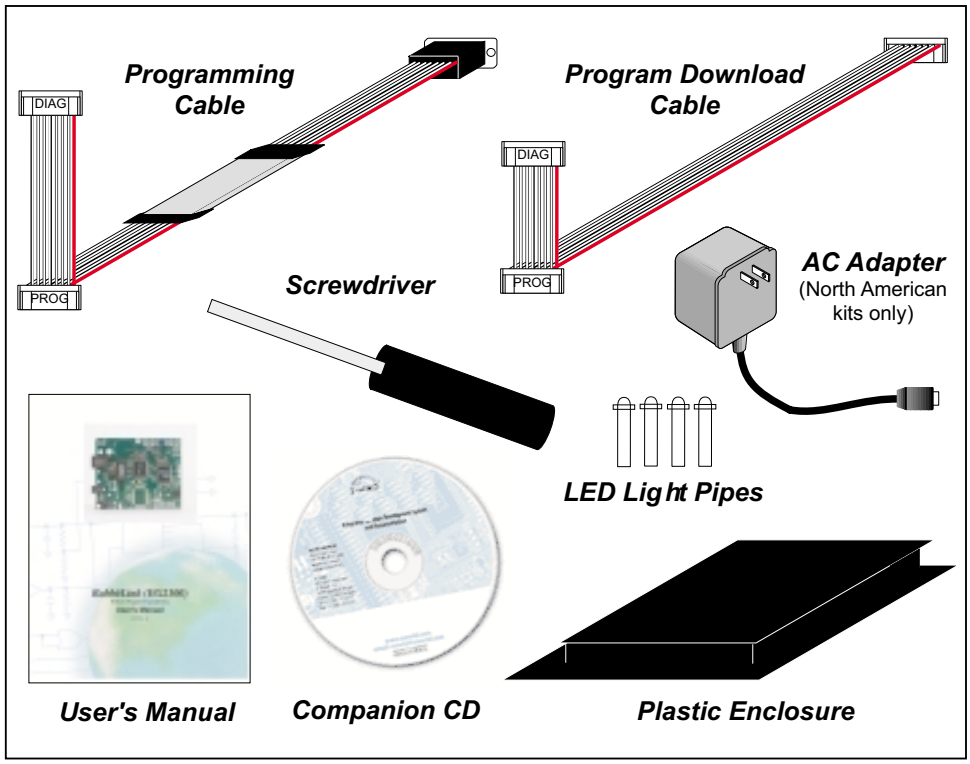


Figure 2. RabbitLink Development Tools
(only the program download cable is supplied with each RabbitLink board)

1.2.2 Software

The RabbitLink gateway is shipped with firmware already installed in the flash memory, and can be used without modification once some basic network parameters are set. In addition, the firmware source code is provided on the companion CD supplied with the tool kit along with binary images of the software and a utility to load the image to the board. The utility allows users to modify the software using Dynamic C (version 7.03 or later), or simply receive firmware updates in the form of binary images.

The programming cable used to configure the RabbitLink's network parameters has a level converter board in the middle of the cable since the **PROG IN** port on the RabbitLink supports CMOS logic levels, and not the higher voltage RS-232 levels that are used by PC serial ports. The network parameters are configured with the PC running a terminal program such as Hyperterminal or TeraTerm.

Attaching the RabbitLink gateway to the Ethernet and a Rabbit-based board allows a user to compile, run, and debug programs on the remote board from a network-connected PC running Dynamic C 7.03 or later. All the standard features of Dynamic C are available over the remote interface, including the Dynamic C **STDIO** window, watch expressions, and the ability to step through C and assembly code. Dynamic C can be used exactly the same way as it is for a board connected to the PC's serial port.

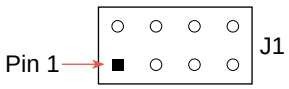
Every time Dynamic C initiates a new RabbitLink session, the RabbitLink requests an encrypted password from the user to ensure the security of the programs and data on the remote boards. This password is stored by Dynamic C during that session so it only needs to be entered the first time Dynamic C contacts the RabbitLink gateway.

In addition, the RabbitLink can be used as an Ethernet interface for any board with an available serial port or Rabbit programming port. Simple versions of a Web server and an SMTP E-mail client are built in and can be controlled by any embedded system via the RabbitLink's serial port, allowing the target board to provide information to the network through easily updated static HTML pages and E-mail. The simple serial protocol used by the RabbitLink serial console makes this interface compatible with an existing board's programming.

1.3 Manual Conventions

Pin Number 1

A black square indicates pin 1 of all headers.



2. GETTING STARTED

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

2.1 RabbitLink Connections

1. Attach the RabbitLink board to the plastic enclosure base.

Position the RabbitLink board over the plastic enclosure base as shown below in Figure 3. Attach the RabbitLink board to the base at the top left and bottom right positions using the two 4-40 × ¼ screws supplied.

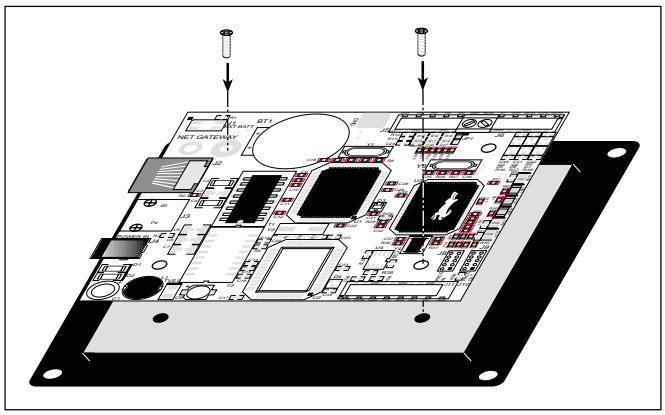


Figure 3. Attach RabbitLink Board to Plastic Enclosure Base

The plastic enclosure base facilitates handling the RabbitLink during development. The plastic enclosure is offered as a separate option when individual RabbitLink boards are purchased.

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

2. Connect the programming cable to configure network parameters from your PC.

Connect the 10-pin **DIAG** connector of the programming cable to header J8, which is labled **PROG IN**, on the RabbitLink. Ensure that the colored edge lines up with pin 1 as shown. Connect the other end of the programming cable to a COM port on your PC. Note that COM1 on the PC is the default COM port in Dynamic C Premier.

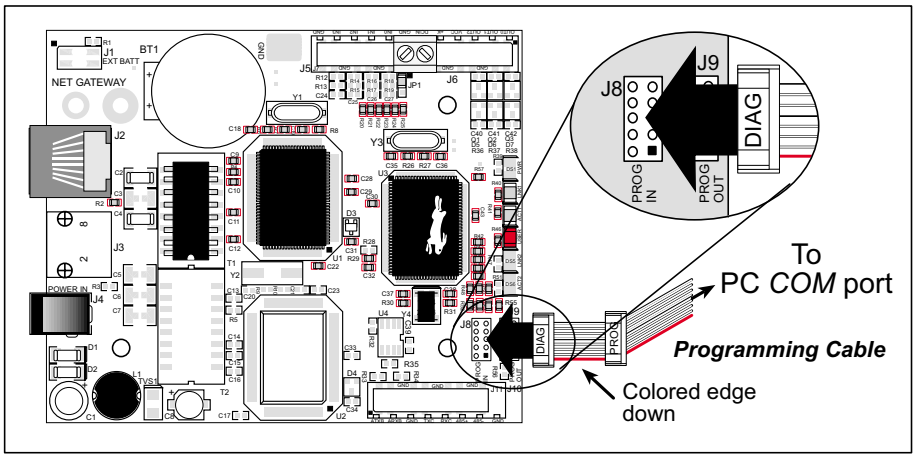


Figure 4. Programming Cable Connections

3. Conne

Two optic Link—via an AC adapter to power sup-
ply jack J /J6. Option 1 is most convenient in a
desktop e onnect the RabbitLink to an existing
power su ED comes on once power is applied
successfu

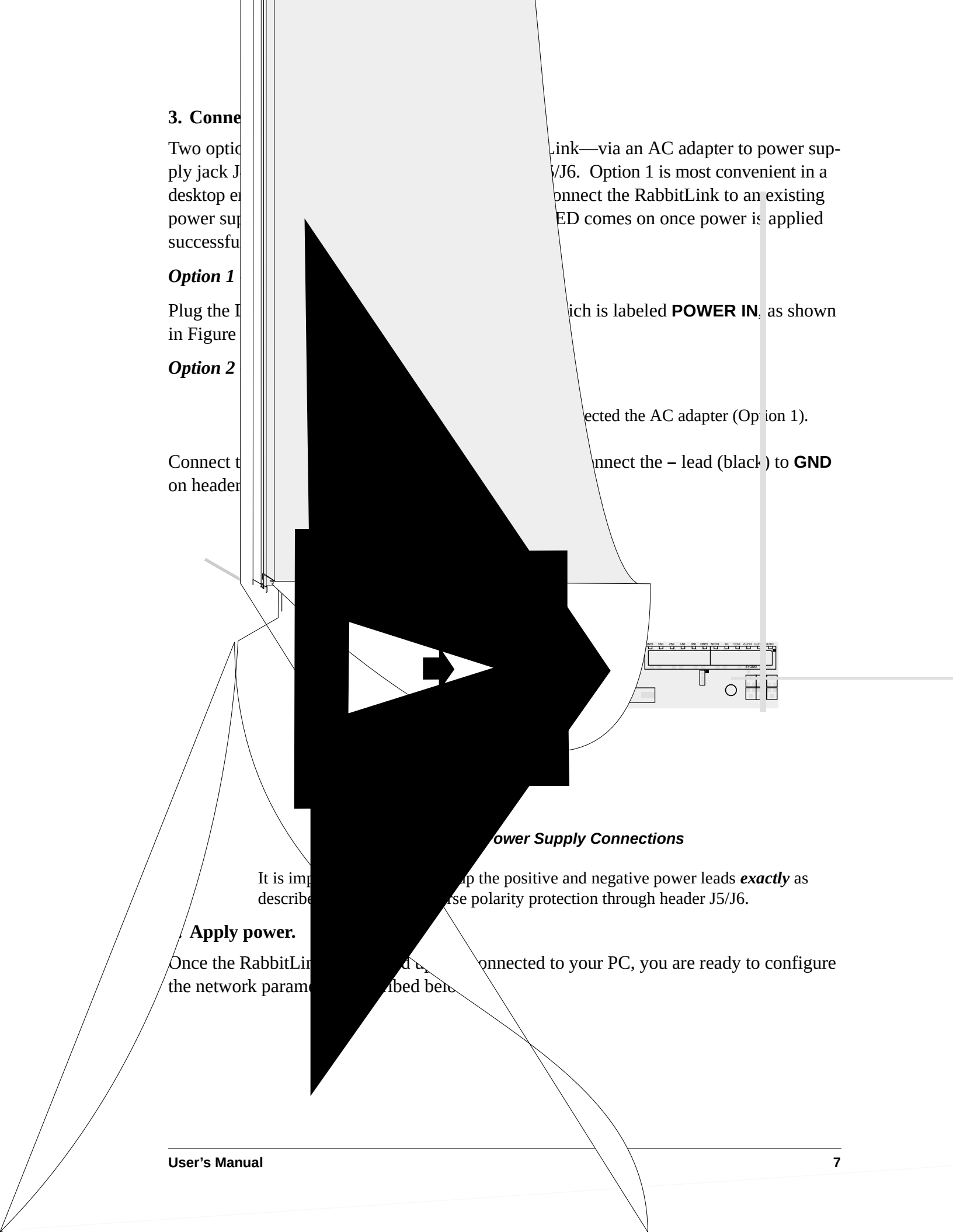
Option 1

Plug the I ich is labeled **POWER IN**, as shown
in Figure

Option 2

ected the AC adapter (Op ion 1).

Connect t nnect the **–** lead (black) to **GND**
on header



ower Supply Connections

It is imp p the positive and negative power leads ***exactly*** as
describe rse polarity protection through header J5/J6.

4. Apply power.

Once the RabbitLi d t onnected to your PC, you are ready to configure
the network parame ibed bel

2.2 Configuring RabbitLink Network Parameters from Your PC

1. Open a terminal emulator such as Tera Term or Windows Hypertermal on your PC. Configure the terminal emulator as follows.

Parameter	Setting
COM	port (COM1 or COM2) to which programming cable is connected
Baud Rate	57,600 bps
Data Bits	8
Parity	None
Stop Bits	1
Flow Control	None

Click OK once these parameters have been set. You should now be able to type commands.

Tip If no characters appear when you type, press return, then type **echo on** to turn on the echoing of characters.

2. Configure the RabbitLink network parameters. You will need to know your network parameters. Ask your network administrator for assistance, if necessary.

Tip Type **show** to get a list of all assignable variables and their settings.

Set IP Address—Type **set ip x.x.x.x**, where **x.x.x.x** is the IP address you wish to assign to the RabbitLink board. Factory default: **10.10.1.100**.

Set Gateway—Type **set gateway x.x.x.x**, where **x.x.x.x** is the gateway you wish to assign to the RabbitLink board. Factory default: **10.10.1.1**.

Set Netmask—Type **set netmask x.x.x.x**, where **x.x.x.x** is the netmask you wish to assign to the RabbitLink board. Factory default: **255.255.255.0**.

Set sender's E-mail address—Type **set mail from <username@domain.ca>**, where **<username@domain.ca>** is the E-mail address of the target system "sender" (used only if the RabbitLink is being configured for E-mail).

Set mail server IP address—Type **set mail server x.x.x.x**, where **x.x.x.x** is the IP address of the RabbitLink mail server (used only with E-mail).

Set Port—Type **set port x**, where x is the TCP port number. The default TCP port used is 4244. If you are using the RabbitLink across a firewall, and the firewall changes TCP port values, it may be necessary to change the TCP port number.

Set Name—Type **set hostname <name>**, where **<name>** is a unique name (up to 40 characters) that you assign to the RabbitLink board. This name is not tied to any of the network addresses, but does show up in the Dynamic C Premier discovery window. This feature allows the RabbitLink board to retain a given name even if its IP or other network address changes.

Set Password—Type **set passphrase**, and you will be prompted for a password. This feature protects the RabbitLink from unwanted Internet connections. The password is ***not*** limited to 8 characters, and indeed the password should be as long as possible (up to 255 characters) to increase the security of the system.

2.2.1 Other Useful Commands

Review Settings—Type **show** to get a list of all assignable variables and their settings (except the password).

Reset—Type **reset** to reset the values (except the password) to their factory defaults.

Help—Type **help** to get online help. You may limit the topics listed by typing **help ...** to limit the help information to the topic specified.

Appendix F, "Serial Console Commands," provides a complete list of the serial console commands.

2.3 Target System Connections to the RabbitLink

Once the RabbitLink network parameters have been configured, the RabbitLink is ready to be used to link any Rabbit-based system to an Ethernet-based network.

1. If you have not already done so, disconnect and set aside the programming cable used to connect the COM port on your PC to header J8, which is labeled **PROG IN**, on the RabbitLink board.
2. Connect the 10-pin **PROG** connector of the program download cable to the programming header of the Rabbit-based system you are going to interact with. Check that the colored edge of the cable lines up with pin 1 of the programming header as shown in Figure 6.
3. Connect the unmarked connector of the program download cable to header J9, which is labeled **PROG OUT**, on the RabbitLink. Check that the colored edge of the cable lines up with pin 1 of the **PROG OUT** header as shown in Figure 6.
4. The RabbitLink must be connected to an Ethernet-based network via an Ethernet hub.

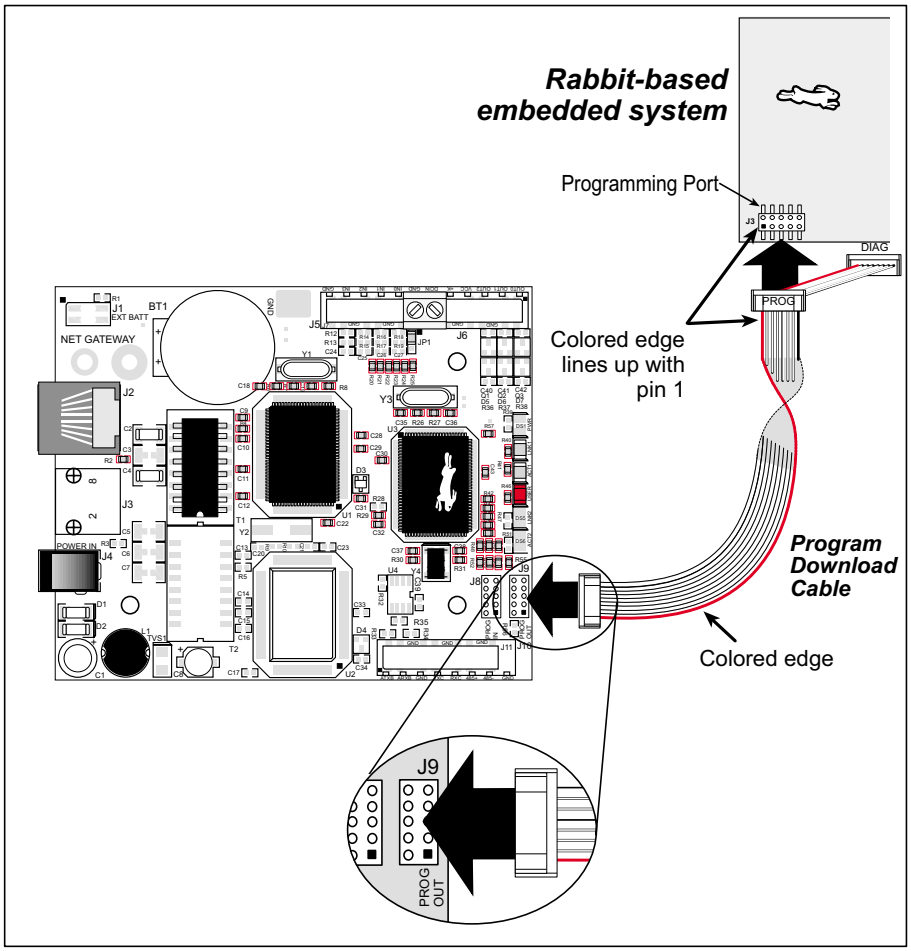


Figure 6. Connect Rabbit-Based Target System to RabbitLink

2.4 Ethernet Connections

Before proceeding, you will need to have either two straight-through Ethernet cables and an Ethernet hub or one Ethernet crossover cable. Your PC must have an RJ-45 Ethernet jack.

You can identify the RJ-45 Ethernet jack by looking for an 8-connector (as opposed to a 6-connector RJ-11 phone jack) jack labeled **ETHERNET** or **<···>** that usually has an LED or two on it. If your PC does not have Ethernet access, you will need to install at least a 10BaseT Ethernet card (available from your favorite computer supplier).

The Ethernet cables and Ethernet hub are available from Rabbit Semiconductor or Z-World in a TCP/IP tool kit. More information is available at www.rabbitsemiconductor.com or at www.zworld.com.

Connect your PC and the RabbitLink board to an Ethernet hub as shown in Figure 7. The Ethernet hub may also be connected to your network.

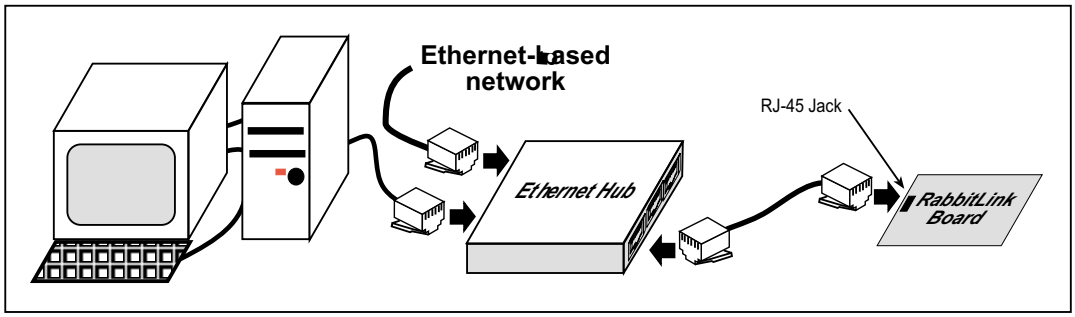


Figure 7. Connect PC and RabbitLink Board to Ethernet Hub

Alternatively, you may use the crossover Ethernet cable to connect the RabbitLink directly to a PC RJ-45 Ethernet jack.

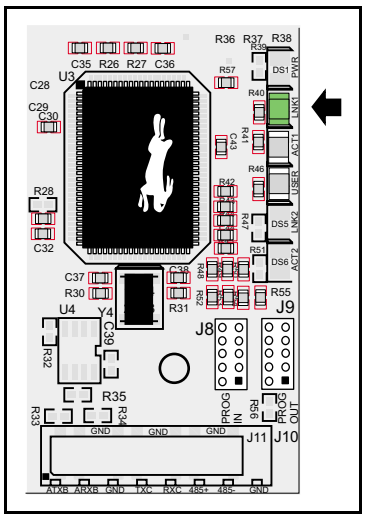


Figure 8. Link LED Lights Up When Connections Made

The green **LINK** LED should light up once all the connections have been made correctly.

Dynamic C (version 7.03 or later) or the Rabbit Field Utility (version 2.0 or later) on your PC may now be used to download a program to the Rabbit-based target system. The red **USER** LED on the RabbitLink blinks while the download or a debug session is in progress.

The Rabbit Field Utility is included on the companion CD. The companion CD also contains a binary image of the firmware, and has the support documention for the RabbitLink.

2.5 Where Do I Go From Here?

If you have any problems, call Rabbit Semiconductor Technical Support at (530)757-8400 or Z-World Technical Support at (530)757-3737.

- Chapter 3, "RabbitLink Software," describes the firmware used to enable the RabbitLink to be a Web server or to originate E-mail. A brief description of Dynamic C for downloading/debugging via the RabbitLink is also included.
- Appendix A, "Specifications," provides specifications.
- Appendix B, "Plastic Enclosure," provides information on the optional plastic enclosure.
- Appendix C, "Install Firmware and Upgrades," provides instructions for installing firmware and firmware updates on your RabbitLink.
- Appendix D, "Subsystems," provides detailed information about the RabbitLink subsystems.
- Appendix E, "Programming Cable," provides additional information about the programming cable and the Rabbit 2000™ programming port..
- Appendix F, "Serial Console Commands," lists all the console commands.

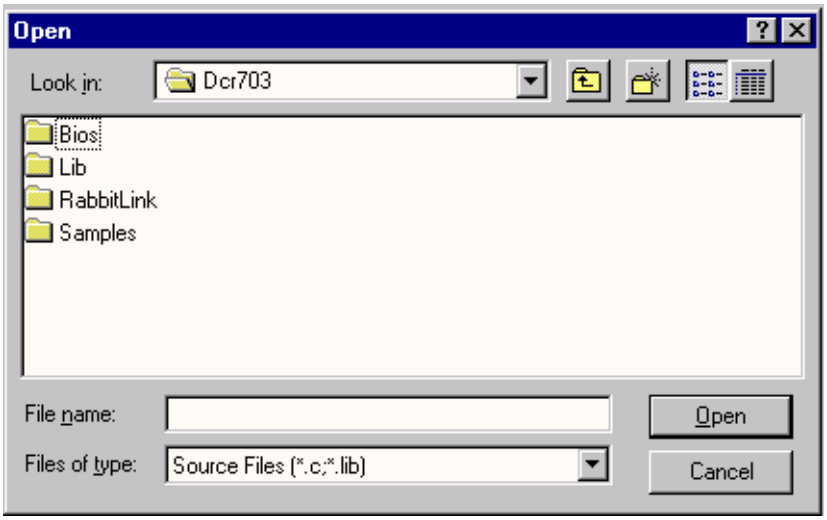
3. RABBITLINK SOFTWARE

All the firmware necessary to operate the RabbitLink with a Rabbit-based target system is already loaded—all you have to do is open a terminal emulator to interact with the serial console on the RabbitLink board, configure the board, and you are ready to go. Although no further application development is needed for the RabbitLink, the application running on the target system will have to be written to take into account the availability of the RabbitLink.

Chapter 3 describes the RabbitLink firmware and the Dynamic C features that enable the target system to be used with the RabbitLink.

3.1 RabbitLink Firmware

When Dynamic C first opens, the following list of Dynamic C root directories will be displayed.

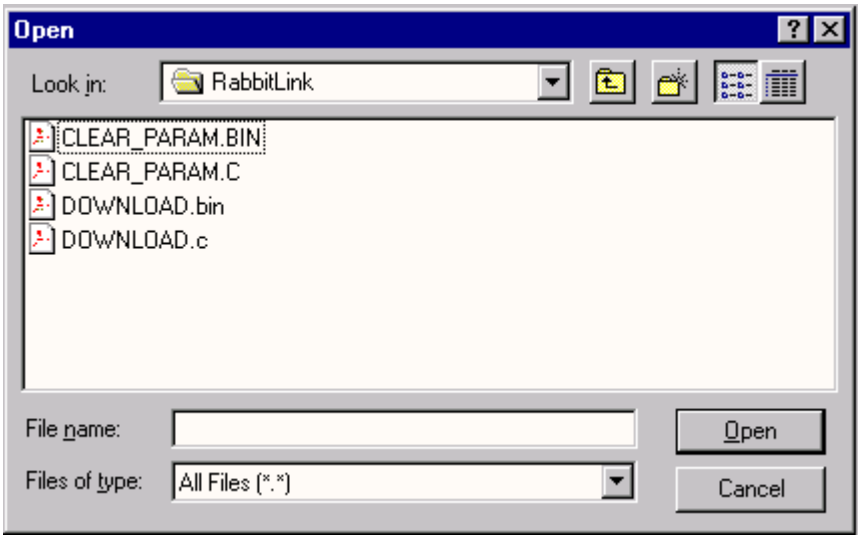


If you plan to install the RabbitLink firmware from the companion CD, you will be given the choice of a destination folder. You may wish to install the firmware as shown above in your Dynamic C directory.

The **RABBITLINK** directory is specific to the RabbitLink, and contains the source code and binary images associated with features specific to the RabbitLink.

The **STDIO.LIB** library in the **Lib** directory contains functions used by target boards to access the RabbitLink serial console. Other functions applicable to all devices based on the Rabbit 2000 microprocessor are described in the ***Dynamic C Premier User's Manual***.

All the firmware necessary to operate the RabbitLink with a Rabbit-based target system is already loaded at the factory. In case it becomes necessary to reload the firmware, the **RABBITLINK** directory contains the source code and the binary image to download.



- **CLEAR_PARAM.BIN**—Binary image to reset the network configuration parameters of the RabbitLink board (stored on the second flash) to default values. This binary file is used to reset the serial port if its operation has become undefined.
- **CLEAR_PARAM.C**—Source code to reset the network configuration parameters of the RabbitLink board (stored on the second flash) to default values.
- **DOWNLOAD.C**—RabbitLink firmware source code.
- **DOWNLOAD.BIN**—RabbitLink firmware binary image.

Refer to Appendix C.1, "Download Firmware," for complete instructions on how to load a firmware upgrade or reload the firmware.

3.2 Downloading and Debugging through the RabbitLink

3.2.1 Getting Started

Before attempting to do any downloading or debugging through the RabbitLink, you will need either the Rabbit Field Utility (version 2.0) or Dynamic C (version 7.03 or later) installed on your PC.

If you plan to use the Rabbit Field Utility, version 2.0 is available on the companion CD. Select the Rabbit Field Utility on your PC desktop after installing the software on the companion CD. Complete instructions for using the Rabbit Field Utility are included in the **Help** menu for the Rabbit Field Utility.

Note that pre-2.0 versions of the Rabbit Field Utility are unable to handle Ethernet transfers by the RabbitLink.

3.2.2 Downloading and Debugging

Regardless of whether you use the Rabbit Field Utility or Dynamic C, downloading across an Ethernet connection is essentially the same as across a serial connection.

Dynamic C can also be used for debugging across an Ethernet or an Internet connection. Once a program is successfully downloaded to the controller attached to the RabbitLink, Dynamic C can be used to debug the program in exactly the same way as it would be used via a serial link.

Refer to the ***Dynamic C Premier User's Manual*** for more detailed information on downloading and debugging.

3.2.2.1 RabbitLink Configuration Commands

Configure the following RabbitLink network parameters before using the RabbitLink.

set ip <IP address>

set gateway <address>

set netmask <address>

set port <TCP port number>

set hostname <name>

set passphrase <passphrase>

The configuration commands and procedures are described in detail in Section 2.2, "Configuring RabbitLink Network Parameters from Your PC,"

3.2.2.2 Connections

Connect the RabbitLink board as shown in Figure 9 with the **PROG** connector on the program download cable connected to the programming header of the Rabbit-based target system and the unmarked connector connected to the **PROG OUT** header of the RabbitLink board. The RabbitLink board is also connected to the Ethernet and should have been configured with its IP address, netmask, and gateway information.

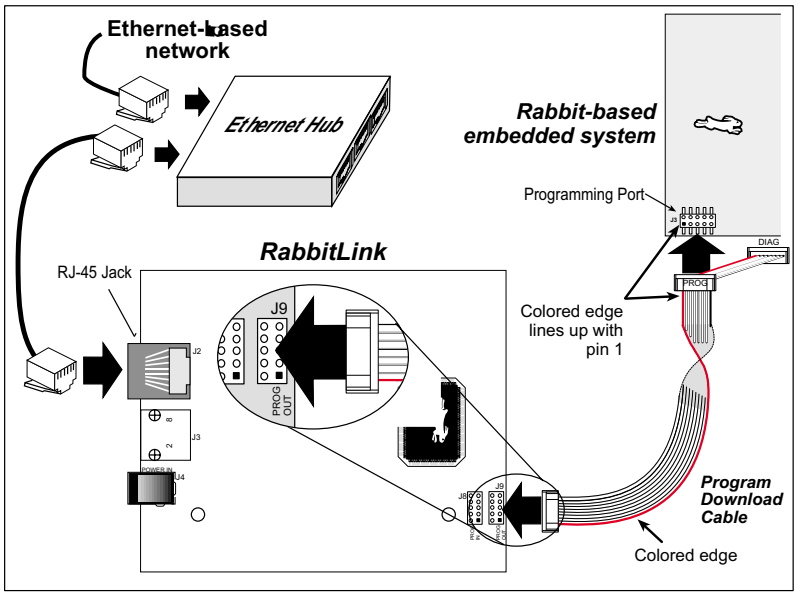


Figure 9. RabbitLink Connections for Download/Debug

Enter the information in the Dynamic C **Options > Communications** dialog box for the RabbitLink you are connecting to:

- The **Network Address** field should contain the IP address of the RabbitLink.
- The **Control Port** field should be set to the TCP port number that the RabbitLink uses to accept control commands from Dynamic C—the default value is 4244. If the RabbitLink is behind a firewall, the **Control Port** field should be set to the port on the firewall that is being forwarded to the RabbitLink.

If a RabbitLink or multiple RabbitLinks are attached to a local network, press the **Discover** button in the **Options > Communications** dialog box to have Dynamic C send a broadcast message to each RabbitLink attached to the network. The default port for discovery is 4242. Each RabbitLink will respond with its IP address, name, control port, current status, and MAC address. Selecting any line in the **Discover** window will cause the information for that RabbitLink to be placed into the appropriate fields in the **Options > Communications** dialog box.

3.2.3 Troubleshooting Tips

- If Dynamic C is unable to establish communication with the RabbitLink, make sure that the RabbitLink is powered on, and make sure that the Ethernet cable is firmly connected to the RabbitLink and that the **USER** and **ACTIVE** LEDs are on.
- If Dynamic C is still unable to establish communication with the RabbitLink, make sure that Dynamic C has the correct IP address and control port information.
- If Dynamic C is able to establish communication with the RabbitLink, but midway through the download process displays the error message "Error receiving Flash ID from target," check to make sure that the program download cable is plugged in correctly from the RabbitLink to the controller being programmed, and that the controller is powered on.

3.3 Accessing the RabbitLink Serial Console

The RabbitLink serial console provides the commands to set network parameters, to create and delete Web pages, and to send E-mail. The serial console can be accessed via the serial port using the **PROG IN** connector on the RabbitLink, as was done in Chapter 2 to configure the RabbitLink. This method can also be used by a non-Rabbit board with a RabbitLink to provide Web and E-mail access.

If a Rabbit 2000-based target board is connected to the **PROG OUT** port of the RabbitLink, then the target board can communicate with the RabbitLink console via the program download cable by using the following Dynamic C functions.

int PrintToConsole(char flag);

This function controls whether **STDIO** commands such as **printf** go to the RabbitLink serial console in addition to the Dynamic C **STDIO** window.

Parameter

flag 0–**printf** and related commands work as normal,
1–**printf** and related commands go to the RabbitLink serial console as well.

Return Value

0 if successful, -1 if not.

Library

STDIO.LIB

See also

getchar, **SendToConsole**, **ConsoleFinish**

int SendToConsole(char *data, int length);

This function writes a binary buffer of a specified length to the serial console on a RabbitLink. Any data are acceptable since the data will not show up in the Dynamic C **STDIO** window.

Parameters

data is a pointer to the data to be sent.

length is the length of the buffer passed in **data**.

Return Value

The actual number of bytes written to the console.

Library

STDIO.LIB

See also

getchar, **PrintToConsole**, **ConsoleFinish**

int ConsoleFinish(long timeout);

This function finishes receiving data from the RabbitLink serial console by blocking for an optional amount of time to do it. If the timeout is set to 0, the function will not receive any data, but will poll to determine whether there are more data to receive.

Parameters

timeout is the length of time to time out, in milliseconds, and is 0 for **ConsoleFinish** to determine whether there are more data to receive.

Return Value

0 if there are more data on the serial console, non-zero if all the data have been received.

Library

STDIO.LIB

See also

getchar, PrintToConsole, SendToConsole

Example

```
main() {
  // first method
  SendToConsole("set ip 10.10.2.102\n", 19);

  // second method
  PrintToConsole(1);
  printf("set gateway 10.10.2.1\n");
  PrintToConsole(0);
}
```

Additional examples are provided in the **Samples/RabbitLink** folder. Refer to Section 3.7, “Sample Programs,” for more information.

3.4 Web Serving

The Web server built into the RabbitLink firmware can be controlled via the RabbitLink console. HTML pages, images, and other files can be added and deleted from the file system on the second flash, and can be loaded by a browser elsewhere on the network at any time. See Appendix F, "Serial Console Commands," for detailed descriptions of the console commands.

3.4.1 RabbitLink Configuration Commands

The following RabbitLink parameters must be configured before the RabbitLink is used.

set ip <IP address>

set gateway <address>

set netmask <address>

The configuration commands and procedures are described in detail in Section 2.2, "Configuring RabbitLink Network Parameters from Your PC, "

3.4.1.1 Connections

Connect the RabbitLink board as shown in Figure 10 with the **PROG** connector on the program download cable connected to the programming header of the Rabbit-based target system and the unmarked connector connected to the **PROG OUT** header of the RabbitLink board.

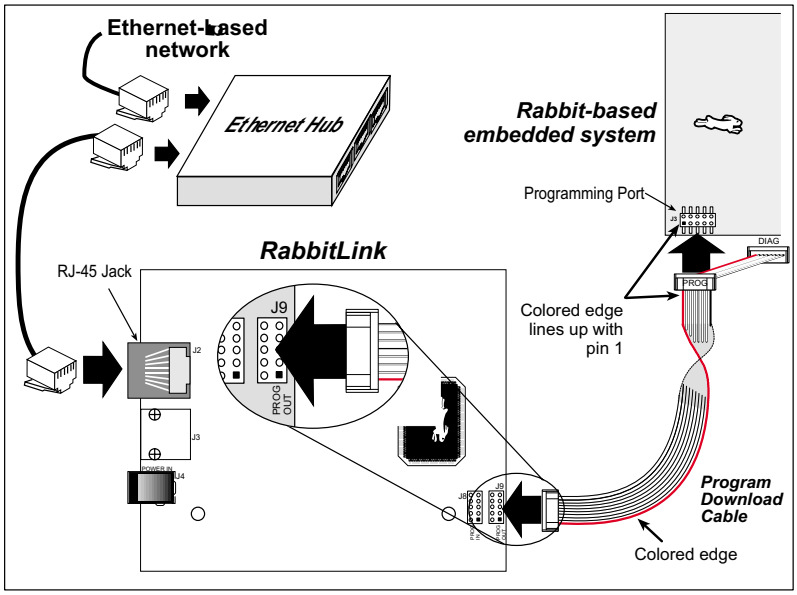


Figure 10. RabbitLink Connections for Web Serving (Rabbit-based embedded system)

Refer to Section 3.6, "Web Serving and E-mail With Non-Rabbit Boards, " for informatiom on connecting a non-Rabbit embedded system.

3.4.2 Creating Web Pages

The RabbitLink serial console contains commands for writing, reading, and deleting files from the flash file system.

3.4.2.1 ASCII Files

The target embedded system may place ASCII files (for example, an HTML file) on the RabbitLink. Use the **put** command shown below.

put <filename>, where **<filename>** is the file name (timeout 60 seconds)

Use a **<Ctrl-d>** or a **<Ctrl-z>** to end the command since **Enter(↵)** will be interpreted by the RabbitLink serial console as a carriage return (**\r**) or as a carriage return + line feed (**\r \n**) in the text or the file name being entered.

A timeout is built in to the **put** command so that data transfer will begin no later than after the timeout. The timeout countdown process does not begin until the RabbitLink senses there is no activity.

The following example shows how to create and update a simple Web interface.

First, create the HTTP file.

```
put status.html
<H2>Status:</H2>
Temperature: OK
<P>Humidity: OK
<P>See <A HREF="data.txt">data</A>
<Ctrl-d>
```

Note that the final line represents actually entering a **<Ctrl-d>** to end the file, not typing that text.

The RabbitLink responds **OK**.

Next, create the data file.

```
put data.txt
TIME  TEMP  HUM
0000  21    67
0600  25    75
1200  27    78
<CTRL-d>
```

The RabbitLink responds **OK**.

The files currently in the file system can be listed by the **list files** command.

```
list files
```

The RabbitLink responds

```
      86  status.html
      61  data.txt
OK
```

Use the **get** command to view a file at any time. For example, type

```
get status.html
```

The RabbitLink responds

```
<H2>Status:</H2>
Temperature: OK
<P>Humidity: OK
<P>See <A HREF="data.txt">data</A>
OK
```

Tip Type **get <filename> 0** to see the size, in bytes, of **<filename>**.

3.4.2.2 Binary Files and Images

Use the binary version of the **put** command to send a binary file (for example, GIF or JPG).

put <filename> <size>, where **<filename>** is the file name and **<size>** is the size, in bytes, of the file (timeout 10 seconds)

Specifying the number of bytes to be sent alerts the RabbitLink console that this file is not an ASCII file, and the RabbitLink will accept all the characters sent as raw data bytes, ignoring the special meaning of characters such as **<CTRL-d>** (04h) that would otherwise signify the end of an ASCII file.

A timeout is built in to the **put** command so that data transfer will begin no later than after the timeout even though the file size is less than the maximum or the specified size. The timeout countdown process does not begin until the RabbitLink senses there is no activity.

The following example shows how to create and update a simple Web interface.

First, create the HTTP file.

```
put data.bin 2034
[2034 bytes of raw data follow]
```

The RabbitLink responds **OK**.

The files currently in the file system can be listed by the **list files** command.

```
list files
```

The RabbitLink responds

```
      86  status.html
      76  data.txt
     203  data.bin
OK
```

Tip It is faster and more efficient to transfer all files, including ASCII files, as binary files.

Images (GIF or JPG files) are often very large binary files. It is easier to transfer these files from the PC directly to the RabbitLink, especially when the images will not change. This can be accomplished by connecting the RabbitLink directly to your PC, as was done to configure the RabbitLink (see Section 2.2, "Configuring RabbitLink Network Parameters from Your PC,"). Connect the **DIAG** end of the programming cable to the **PROG IN** port of the RabbitLink, and the other end to your PC's serial port as shown in Figure 11.

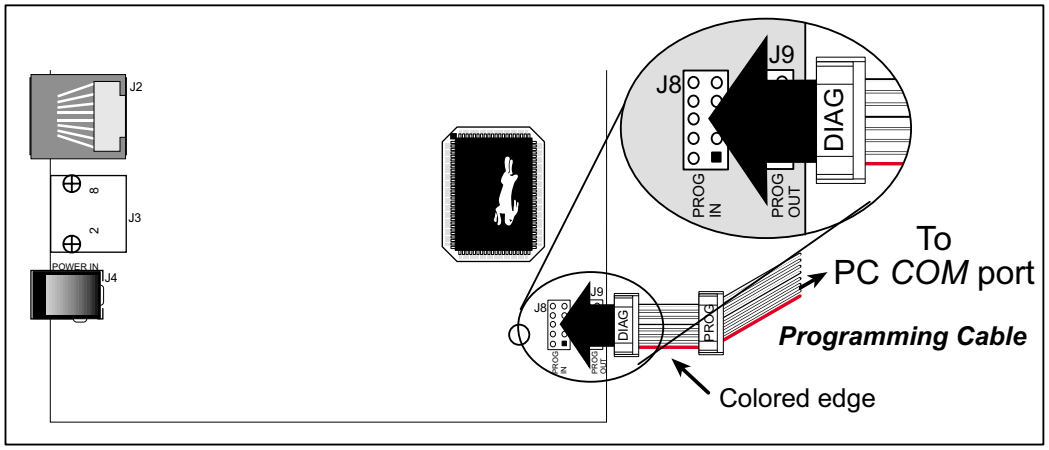


Figure 11. RabbitLink Connections for Transferring Files Directly to RabbitLink Flash from PC

Since there is no flow control supported on the **PROG IN** port of the RabbitLink, it is advisable to reduce the baud rate from 57,600 bps to a lower value while transferring large binary files to ensure that no bytes are missed as the file is written to the flash file system. Alternatively, some terminal programs (such as TeraTerm) can add a delay to a binary file transfer.

3.4.3 Viewing Web Pages on the RabbitLink

The RabbitLink should be connected to the target system and the Ethernet as shown in Figure 10. In a Web browser connected to the same network that the RabbitLink is on, go to the address

```
http://<IP address>/<filename>
```

where **<IP address>** is the IP address of the RabbitLink, and **<filename>** is the name of the Web page you wish to view.

To continue the example, enter the address

```
http://<IP address>/status.html
```

into a Web browser with network access to the RabbitLink board, replacing **<IP address>** with the address of your RabbitLink board. The page should appear something like this.

Status:

Temperature: OK

Humidity: OK

See data.

Click on the link to bring up the data file.

```
TIME  TEMP  HUM
0000  21    67
0600  25    75
1200  27    78
```

The files in the RabbitLink file system cannot be appended to, so the file must first be deleted and then rewritten in order to add a new line of data.

Type

```
delete data.txt
```

The RabbitLink responds **OK**.

The files currently in the file system can be listed by the **list files** command.

```
list files
```

The RabbitLink responds

```
      86  status.html
OK
```

Rewrite the new data.

```
put data.txt
TIME  TEMP  HUM
0000  21    67
0600  25    75
1200  27    78
1800  28    75
<Ctrl-d>
```

The RabbitLink responds **OK**.

Reloading the page of data should now show

```
TIME  TEMP  HUM
0000  21    67
0600  25    75
1200  27    78
1800  28    75
```

3.4.4 Serial Console Commands for Variables

The RabbitLink serial console also has the ability to handle SSI (Server Side Includes) variables. These variables are stored in an xmem buffer. This means that the memory will lose the variables when power is cycled, although the references to the variables will still exist.

Any HTML file that includes SSI tags must have the file extension **.shtml**.

The RabbitLink boots up whenever the power cycles, and then displays the message

```
RabbitLink Serial Console Version …
```

This section lists the serial configuration commands associated with variables.

createv <varname> <vartype> <format> <value> [strlen]

This command creates a variable called **<varname>**. The variable can be referenced within Web files (for example, **<!--#echo var="var1"-->**).

<vartype> is the type of the variable (**int8**, **int16**, **int32**, **float32**, or **string**)

<format> is the **printf**-style format specifier (such as **%d**) for outputting the variable

<value> is the value to assign to the variable.

[strlen] is only used if the variable is of type **string**. **[strlen]** is used to give the maximum length of the string.

Examples

```
createv var1 float32 "%.2f" 3.14

createv var2 string "%s" "This is only a test." 50
```

The RabbitLink will respond **OK** after each command.

getv <varname>

This command gets the value of the variable called **<varname>**. The value is printed using the format specifier given in the **createv** command.

Example

```
getv var2
```

The RabbitLink responds.

```
This is only a test.
OK
```

putv <varname> <value>

This command assigns the given **<value>** to the given variable **<varname>**.

Example

```
putv var1 3.14

OK

getv var1

3.14
OK
```

list variables

This command lists all the variables.

Example

```
list variables
```

The RabbitLink responds.

```
var1 float32
var2 string50
OK
```

reset variables

This command deletes all the variables.

Example

```
reset variables
```

The RabbitLink responds **OK**.

Additional examples are provided in the sample program **EG2100Vars.C** found in the **Samples/RabbitLink** folder. Refer to Section 3.7, “Sample Programs,” for more information.

3.5 Send E-mail

The SMTP client built into the RabbitLink firmware is able to send simple E-mail messages to a SMTP server via the RabbitLink console.

3.5.1 RabbitLink Configuration Commands

The following RabbitLink parameters must be configured before the RabbitLink is used.

set ip <IP address>

set gateway <address>

set netmask <address>

set mail from <username@domain.ca>

set mail server <IP address>

The from address will be included in the **From** line of all E-mail messages sent from the RabbitLink. This address should be an actual E-mail address that you have access to since any error responses from the SMTP server will be sent to this address.

Tip Ask your network administrator for assistance with the mail server IP address.

The configuration commands and procedures are described in detail in Section 2.2, "Configuring RabbitLink Network Parameters from Your PC,"

3.5.1.1 Connections

Connect the RabbitLink as shown in Figure 12 with the **PROG** connector on the program download cable connected to the programming header of the Rabbit-based target system and the unmarked connector connected to the **PROG OUT** header of the RabbitLink.

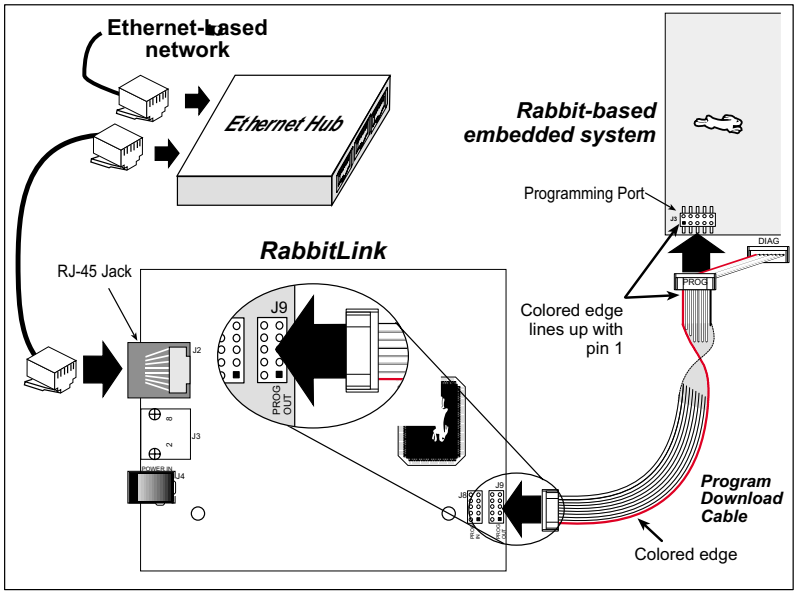


Figure 12. RabbitLink Connections for E-mail (Rabbit-based embedded system)

3.5.2 Sending a Message

To send an E-mail message, use the console **mail** command. The following example shows how an E-mail message is sent.

First, set the RabbitLink parameters. These parameters will be stored in flash so they only need to be set once. The RabbitLink responses are shown in a grey box.

```
set mail from status@rabbitcompany.com
OK
set mail server 10.10.4.212
OK
```

Note that you must specify the IP address of the mail server.

Next, send an E-mail message with the **mail** command. The first line typed after the **mail** command will be the subject line, and the rest will form the body of the message.

```
mail engineer@rabbitcompany.com
Error -- South wall
The system on the south wall has reported an error.
Please check it.
<Ctrl-d>
OK
```

Use a (04h) or a (1Ah) to send the message since **Enter(↵)** will be interpreted by the RabbitLink serial console as a carrier return (**\r**) or as a carrier return + line feed (**\r \n**).

The E-mail is sent immediately.

The E-mail account will receive the following E-mail message.

```
Return-Path: <status@rabbitcompany.com>
From: <status@rabbitcompany.com>
To: <engineer@rabbitcompany.com>
Subject: Error -- South wall

The system on the south wall has reported an error.
Please check it.
```

There may be additional headers as well, depending on your SMTP server and E-mail cli-

3.6 Web Serving and E-mail With Non-Rabbit Boards

Connect the RabbitLink board as shown in Figure 13 with the DE9 connector on the programming cable connected to a serial connector on the non-Rabbit target system and the unmarked connector connected to the **PROG IN** header of the RabbitLink board. The level converter on the programming cable converts the RS-232 levels from the target system to the CMOS-level signals used by the RabbitLink.

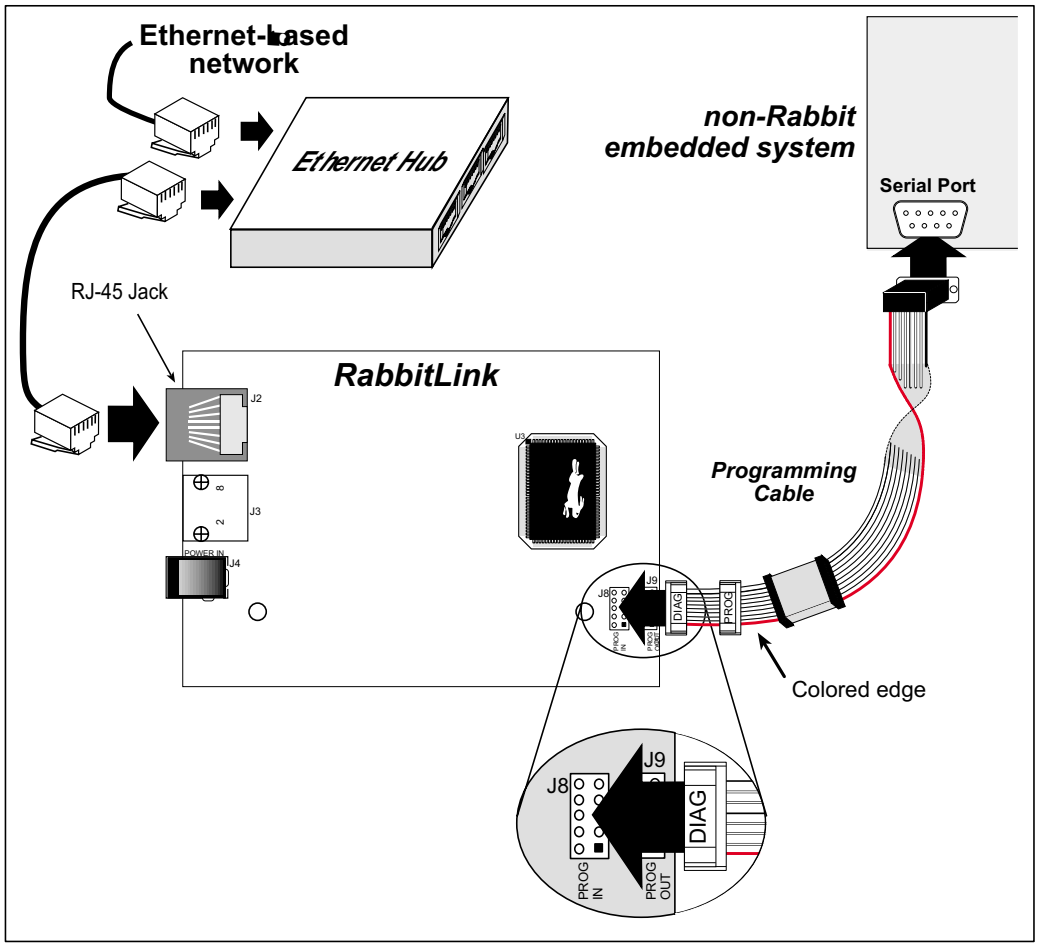


Figure 13. RabbitLink Connections for Web Serving (non-Rabbit embedded system)

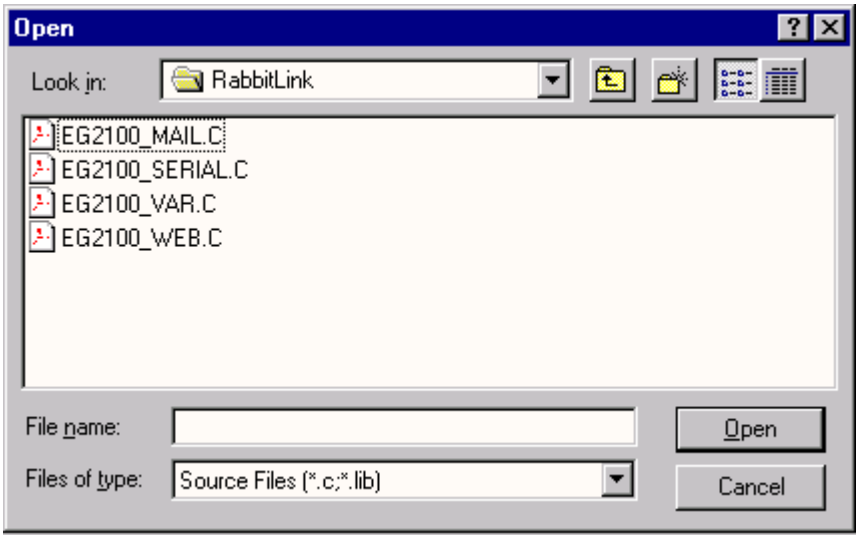
The **put**, **get**, and **mail** commands (and the associated commands for use with variables) are used in the same way as in a Rabbit-based embedded system, but use the serial port on the non-Rabbit target system.

See the **`Samples/RabbitLink`** folder for examples of sending console commands over the serial port. Refer to Section 3.7, "Sample Programs," for more information.

3.7 Sample Programs

Dynamic C Premier contains sample programs for the target systems connected to the Ethernet via the RabbitLink. The sample programs are located in **Samples/RabbitLink**.

Let's take a look at the **Samples/RabbitLink** folder.



- **EG2100_MAIL.C**—Sample program for Rabbit-based embedded system to generate an E-mail message to be sent by the RabbitLink.
- **EG2100_SERIAL.C**—Sample program to illustrate serial communication with any embedded system connected to the RabbitLink.
- **EG2100_VAR.C**—Sample program for Rabbit-based embedded system using variables on Web pages to be sent by the RabbitLink.
- **EG2100_WED.C**—Sample program for Rabbit-based embedded system to generate Web pages to be served by the RabbitLink.

APPENDIX A. SPECIFICATIONS

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

A.1 Electrical and Mechanical Specifications

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

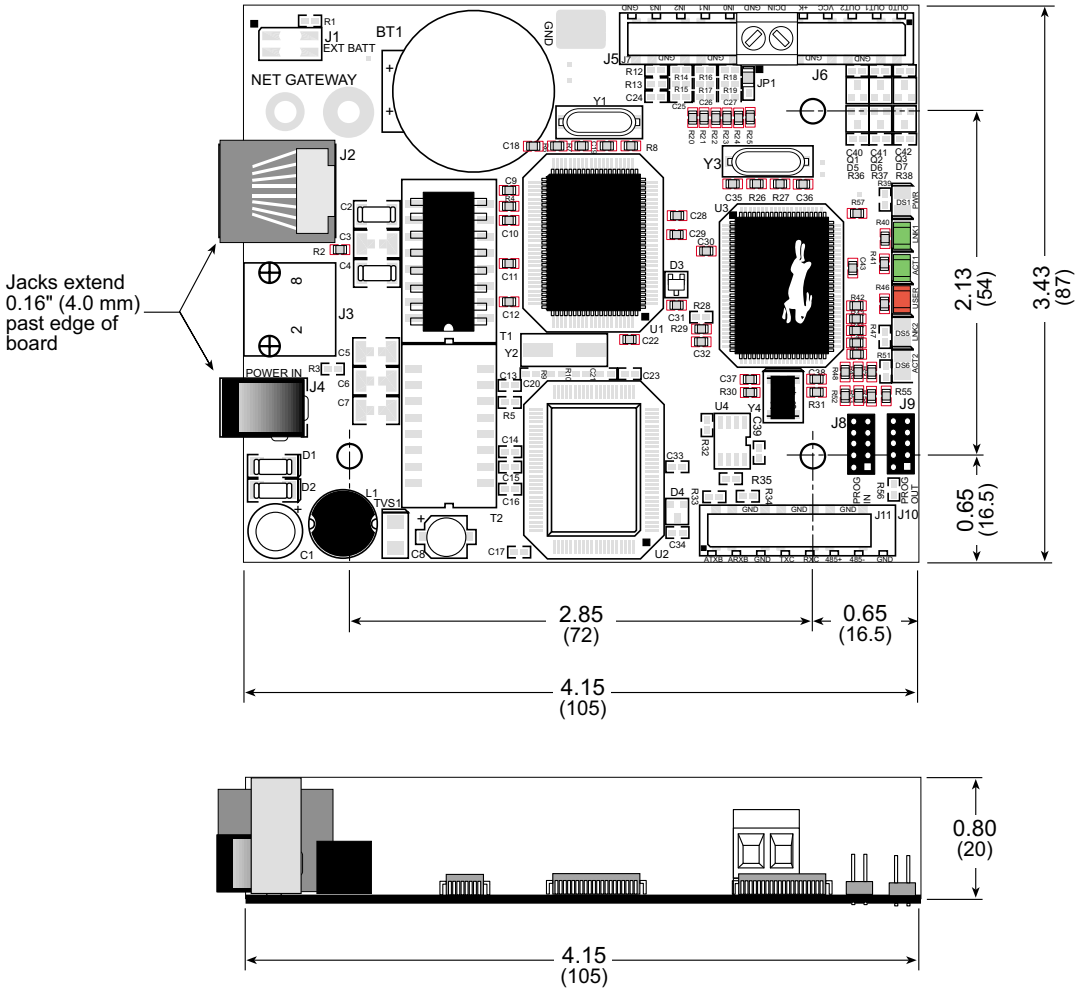


Figure A-1. RabbitLink Dimensions

All measurements are in inches followed by millimeters enclosed in parentheses.

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

Table A-1. RabbitLink Specifications

Parameter	Specification
Board Size	3.43" × 4.15" × 0.80" (87 mm × 105 mm × 20 mm)
Connectors	one RJ-45 (Ethernet) two 2 × 5, 2 mm pitch (serial programming) one contact power jack for AC adpater one 2-terminal screw connector (18 to 26 AWG wire) for wired-in power supply
Ethernet Interface	Direct connection to 10BaseT 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
Current	44 mA at 24 V, 84 mA at 12 V (typical)
Onboard Voltage Regulator	Surface-mount switching regulator sources 5 V at 1 A
Microprocessor	Rabbit 2000™
Clock	22.1 MHz
SRAM	128K, surface mount
Flash EPROM	256K for program plus 256K for data
Serial Ports	2 CMOS-compatible serial programming ports
Serial Rate	Maximum asynchronous 345,600 bps Maximum synchronous 142,700 bps
Watchdog/Supervisor	Yes
Time/Date Clock	Yes
Backup Battery	No

APPENDIX B. PLASTIC ENCLOSURE

The plastic enclosure provides a secure way to enclose your RabbitLink board. The enclosure itself may be mounted on any flat surface.

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

B.1 Assembly

1. Attach the RabbitLink board to the plastic enclosure base.

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

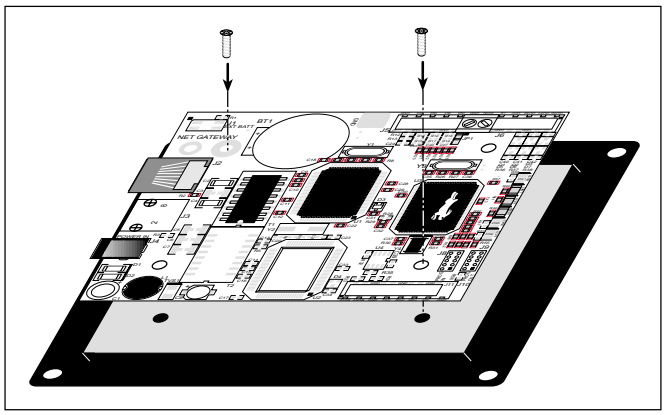


Figure B-1. Attach RabbitLink Board to Plastic Enclosure Base

2. Install light pipes (optional).

Light pipes are included in the tool kit to facilitate seeing the LEDs on the RabbitLink board once the enclosure is assembled.

With the enclosure top positioned as shown in Figure B-2, insert three 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 RabbitLink board. 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.

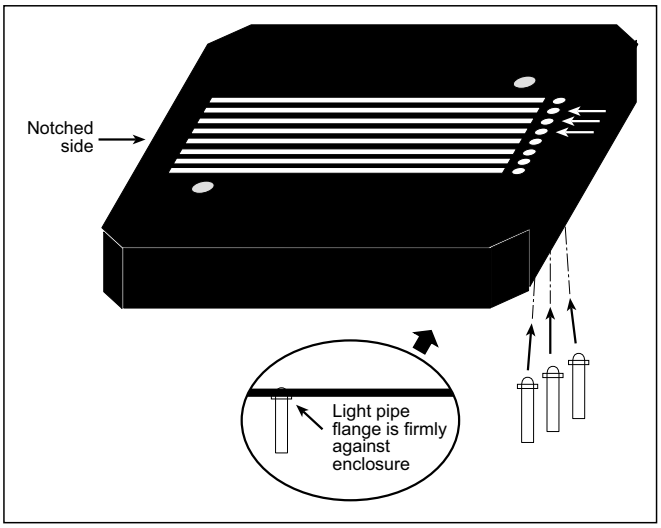


Figure B-2. Install Light Pipes in Enclosure Top

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

3. Attach the enclosure top to the base.

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

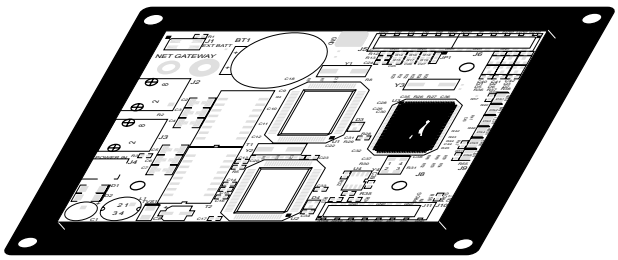


Figure B-3. Attach Enclosure Top

4. Mount plastic enclosure (optional).

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

B.2 Dimensions

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

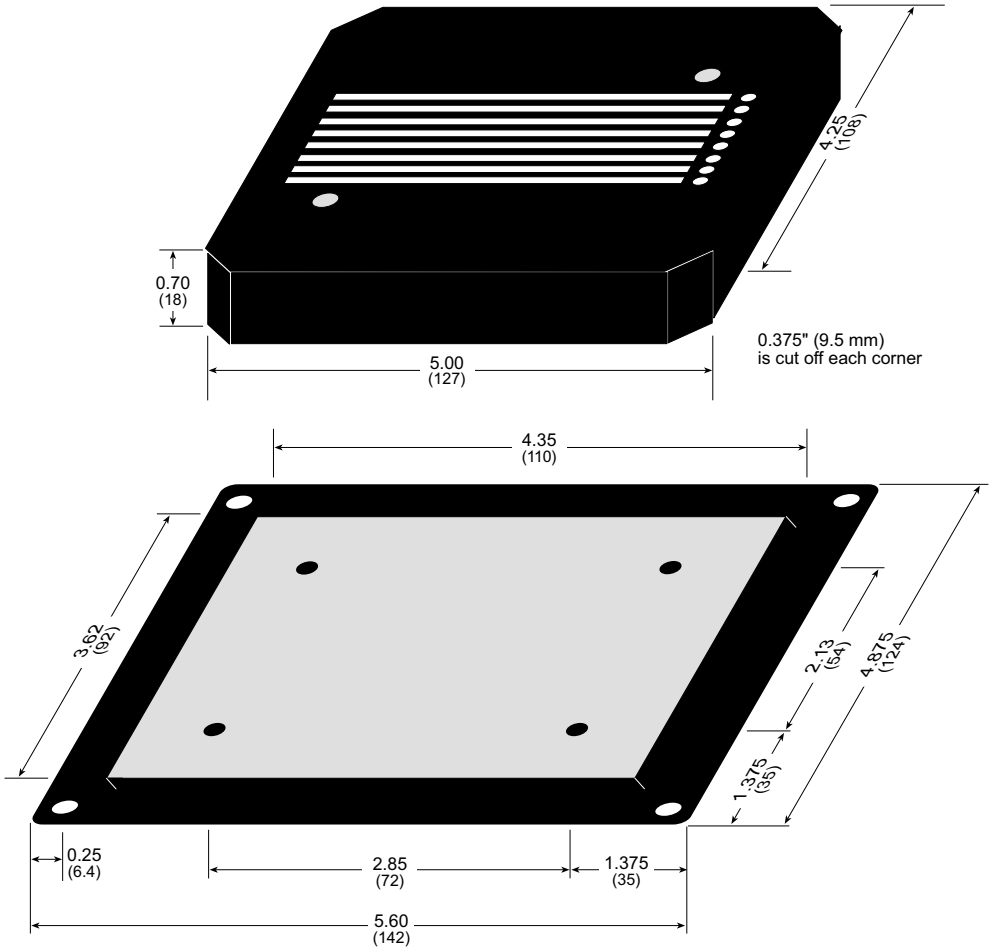


Figure B-4. Plastic Enclosure Dimensions

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

APPENDIX C. INSTALL FIRMWARE AND UPGRADES

Appendix C provides information on how to connect the RabbitLink to reload the firmware and firmware upgrades.

C.1 Download Firmware

Two binary files are included in the Dynamic C **RabbitLink** folder.

- **DOWNLOAD.BIN**—RabbitLink firmware binary image.
- **CLEAR_PARAM.BIN**—Binary image to reset the network configuration parameters of the RabbitLink board (stored on the second flash) to default values. This binary file is used to reset the serial port if its operation has become undefined.

The RabbitLink board supplied from the factory already has **DOWNLOAD.BIN** firmware loaded. Follow these steps to reload the firmware or to reset the RabbitLink serial port.

1. Connect the RabbitLink board to your PC as shown in Figure C-1 with the **PROG** connector on the programming cable connected to the **PROG IN** header of the RabbitLink board.

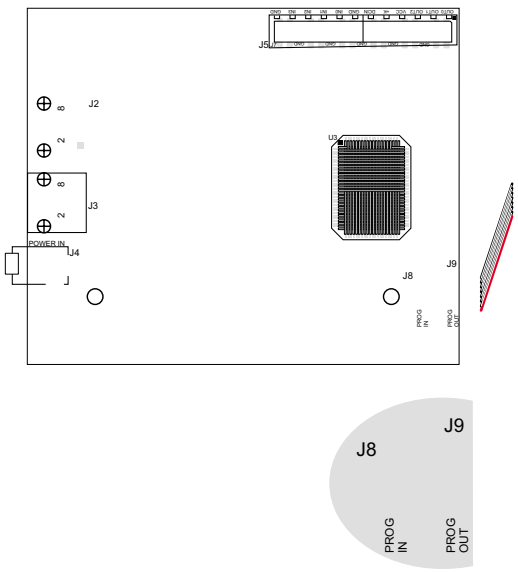


Figure C-1. RabbitLink Connections for Downloading Firmware

2. Use the Rabbit Field Utility to load the **DOWNLOAD.BIN** firmware or the **CLEAR_PARAM.BIN** binary image onto the RabbitLink board.
3. Disconnect the programming cable.
4. Unplug the power supply, then plug the power supply back in. This resets the RabbitLink from Program Mode to Run Mode.

C.1.1 Firmware Upgrades

Follow the above steps to install a firmware upgrade. Just subsitute the name of the firmware upgrade for the **DOWNLOAD.BIN** firmware binary image file. Web ugrades will be available at www.rabbitsemiconductor.com or at www.zworld.com.

APPENDIX D. SUBSYSTEMS

Appendix D describes the principal subsystems for the RabbitLink.

- RabbitLink Subsystems
- Serial Communication
- Memory
- Power Supplies

D.1 RabbitLink Hardware Subsystems

Figure D-1 shows the Rabbit-based subsystems designed into the RabbitLink and shows the parallel ports and signal lines they use on the Rabbit 2000 microprocessor.

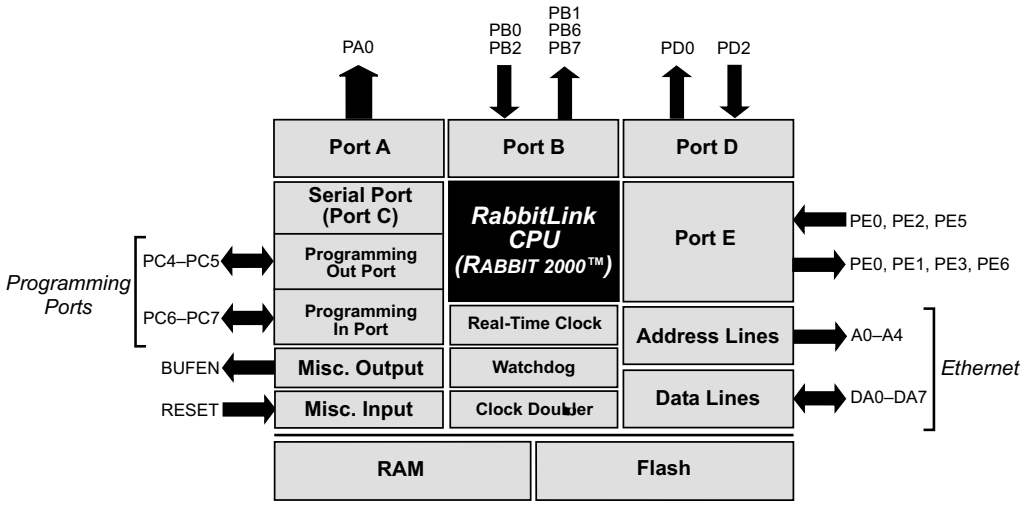


Figure D-1. RabbitLink Subsystems

D.1.1 Pinouts

Figure D-2 shows the pinouts for the RJ-45 Ethernet jack and the two programming headers on the RabbitLink board.

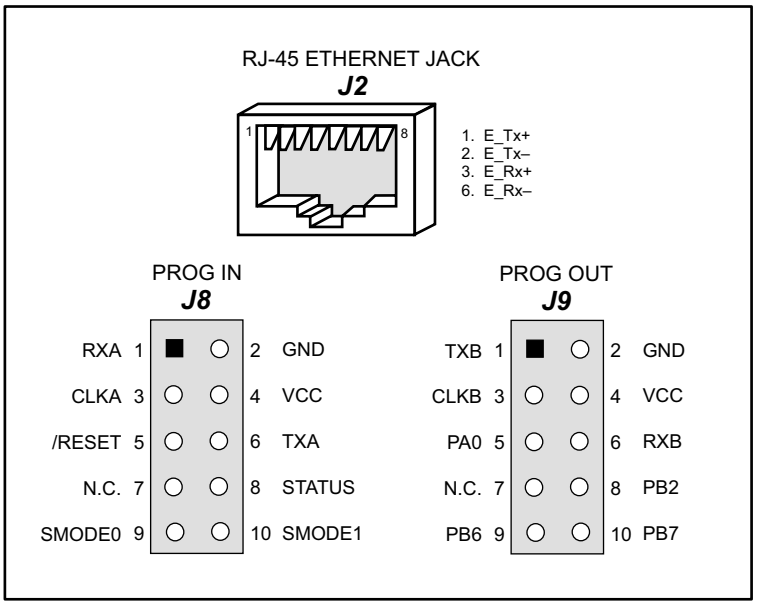


Figure D-2. Pinouts for Ethernet Jack and Programming Headers

D.2 Serial Communication

D.2.1 Serial Programming Ports

The RabbitLink board has two 10-pin programming headers labeled J8 and J9. The **PROG IN** port uses the Rabbit 2000's serial port A for communication, and the **PROG OUT** port uses serial port B. The Rabbit 2000 startup-mode pins (SMODE0, SMODE1) are presented to the **PROG IN** port so that an externally connected device can force a start-up in an external bootstrap mode when the **PROG** connector on either the programming cable or the program download cable is used.

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

The **PROG IN** port is used with the **DIAG** connector on the programming cable to configure the RabbitLink. The **PROG IN** port transmits information to and from a PC running a terminal emulation program.

The RabbitLink network configuration can be reset through the **PROG IN** port.

The **PROG OUT** port is used (with the **PROG** connector on the program download cable connected to the programming port of the target) to download or to debug a program through an Ethernet-based network or even the Internet to a target Rabbit-based board. The **PROG OUT** port transmits information to and from a PC elsewhere on the Ethernet-based network running Dynamic C Premier (version 7.02 or later) or the Rabbit Field Utility.

See Appendix E, "Programming Cable," for more information.

D.2.2 Ethernet Port

The 10 Mbps twisted-pair Ethernet system allows segment lengths of approximately 100 m for "voice grade" twisted-pair telephone wiring. The maximum segment length may be shorter or longer than this, depending on the quality of the twisted-pair cabling in your system. While the 10Base-T system is designed to use voice-grade telephone cable that may already be installed, higher quality Category 5 cables, connectors, and wire terminating devices provide the best possible signal carrying system for 100 Mbps Ethernet media systems.

The 10Base-T media system uses two pairs of wires, which are terminated in an eight-pin (RJ-45 style) connector. This means that four pins of the eight-pin MDI connector are used as shown in Figure D-2. The transmit and receive data signals on each pair of a 10Base-T segment are polarized, with one wire of each signal pair carrying the positive (+) signal, and the other carrying the negative (-) signal.

D.4 Power Supplies

Power is supplied to the RabbitLink board from an external source either through jack J4 or through screw terminal connector J5/J6. The connection through jack J4 is protected against reverse polarity by a Sshottky diode at D1 as shown in Figure D-4, but the alternative connection through J5/J6 is ***not*** protected against reverse polarity.

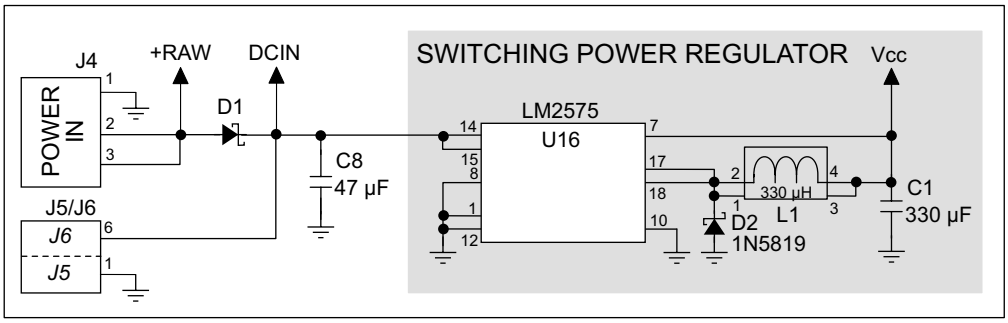


Figure D-4. RabbitLink Power Supply Schematic

The power supply connection through jack J4 is handy for the AC adapter included with the RabbitLink tool kit for desktop demonstration and development. The power supply connection through screw terminal connector J5/J6 enables you to connect the RabbitLink directly to a power supply in the production system.

Capacitor C8 provides noise and ripple stablization protection for the voltage regulator, and allows the external power supply to be located some distance away from the RabbitLink. A switching power regulator is used. The +RAW or DCIN input voltage may range from 9 V to 40 V.

D.5 Batteries and External Battery Connections

Although the RabbitLink has room for a backup battery on the circuit board, battery backup is not supported at this time.

D.6 Reset Generator

The RabbitLink uses a reset generator, U14, 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.

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 with the RabbitLink board. The **PROG** connector is used only when the programming cable is attached to the **PROG IN** connector (header J8) on the RabbitLink to download new firmware. Otherwise, the **DIAG** connector on the programming cable is used to configure the RabbitLink's network parameters, and also allows the programming cable to be used as an RS-232 to CMOS level converter for serial communication.

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.

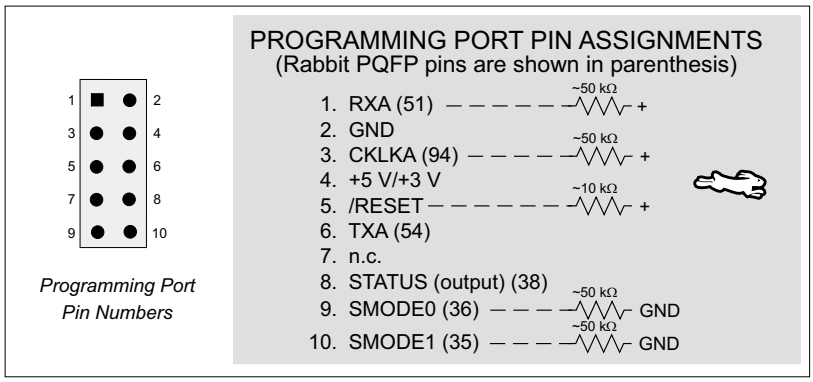


Figure E-1. Programming Port Pin Assignments

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.

APPENDIX F.
SERIAL CONSOLE COMMANDS

Appendix F lists all the serial console commands used to control the RabbitLink.

Table F-1. Configuration Commands

Task	Command	Parameters
set gateway	**set gateway x.x.x.x**	**x.x.x.x** is the gateway IP address
set name	**set hostname <name>**	**<name>** is a unique name (up to 40 characters) that you assign to the RabbitLink
set IP address	**set ip x.x.x.x**	**x.x.x.x** is the IP address
set sender's E-mail address	**set mail from <username@domain.ca>**	**<username@domain.ca>** is the RabbitLink's E-mail address
set mail server IP address	**set mail server x.x.x.x**	**x.x.x.x** is the IP address of the mail server
set netmask	**set netmask x.x.x.x**	**x.x.x.x** is the netmask
set password	**set passphrase**	should be as long as possible (up to 255 characters)
set port	**set port x**	x is the TCP port number, the default TCP port number is 4244
Ask your network administrator for assistance with these values.		

Table F-2. Other Commands

Command		Description
createv	**<varname>** **<vartype>** **<format>** **<value>** **[strlen]**	• variable name • type of the variable (**`int8`**, **`int16`**, **`int32`**, **`float32`**, or **`string`**) • **<format>** is the **`printf`**-style format specifier (such as **`%d`**) for outputting the variable • **<value>** is the value to assign to the variable. • only used if the variable is of type **`string`** to give the maximum length of the string
delete	**<filename>**	deletes **<filename>**
echo	**<on>**	turn character echoing on
	<off>	turn character echoing off
get	**<filename>**	use to view **<filename>**
getv	**<varname>**	gets the value of the variable called **<varname>**
help		online help
	<filename>	online help for **<filename>** specified
list	**files**	show all files stored in flash
	variables	show all variables
put	**<filename>**	use to put **<filename>** on RabbitLink
putv	**<varname>** **<value>**	assigns the given **<value>** to the given variable **<varname>**
reset		reset values (except password) to factory defaults
	<filename>	reset **<filename>**to factory defaults
	variables	deletes all variables
show		get a list of all assignable variables (except the password) and their settings
	<filename>	

INDEX

R

S

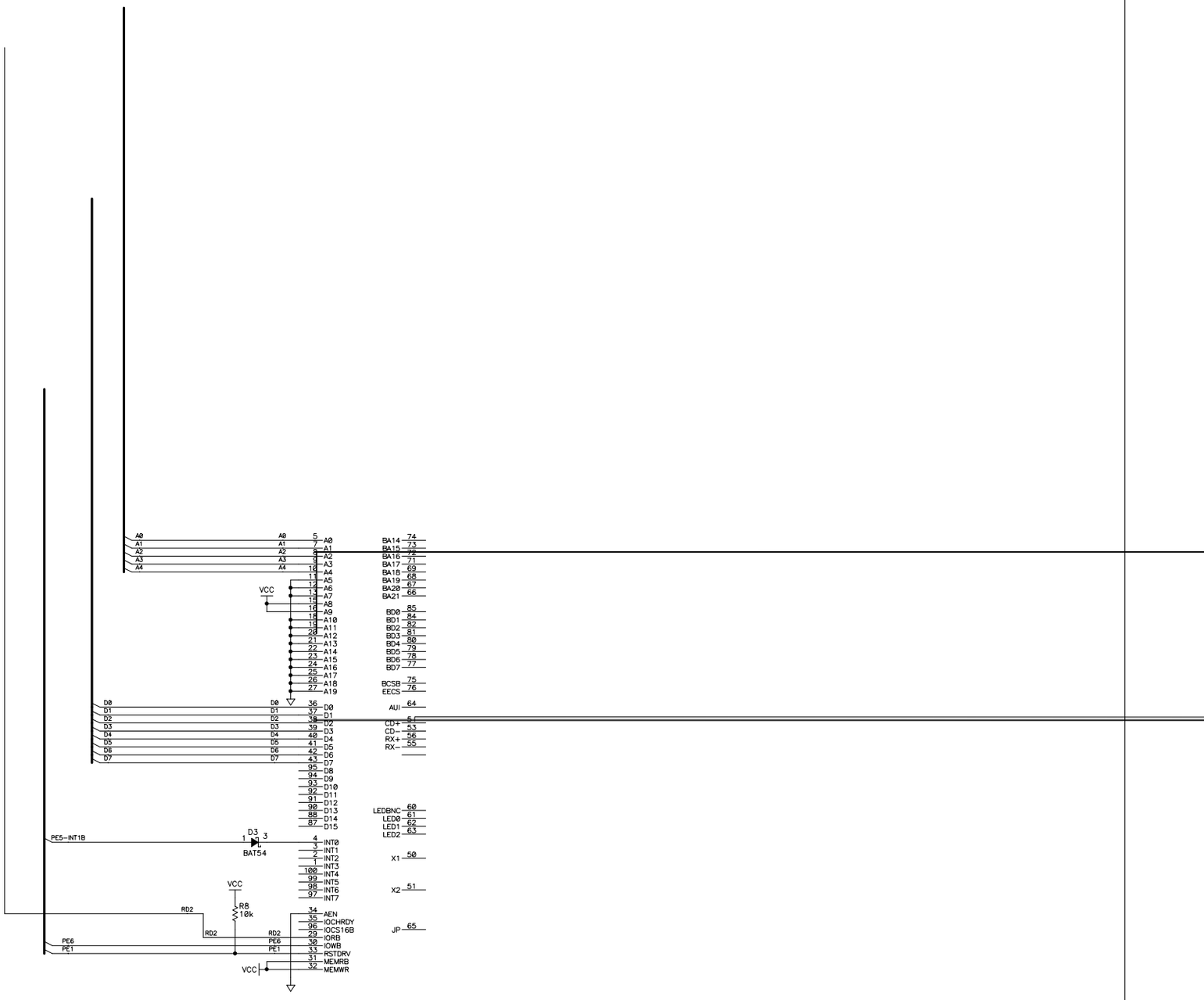
T

W

SCHEMATICS

090-0112 RabbitLink (EG2100) Schematic

090-0085 Programming Cable Schematic



SIZE	DWG NO.	
C		
SCALE NONE	REV LTR	SHEET OF

STUFFING TABLE

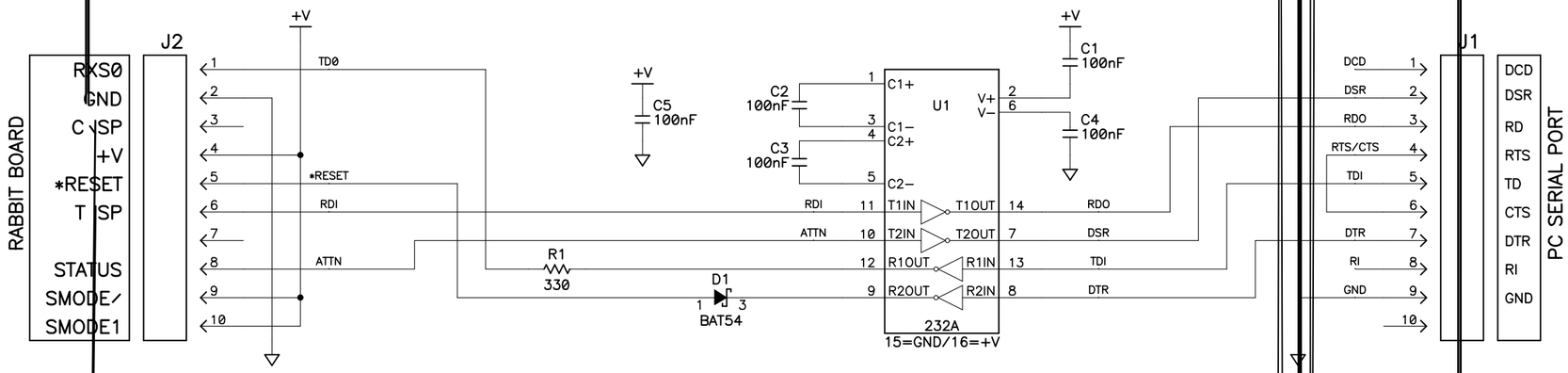
CIRCUIT		PART	MODEL EG2000	MODEL EG2020	MODEL EG2100
BACKUP BATTERY	MAIN	BT1	3V Li	3V Li	NOT INSTALLED
		J1	INSTALLED	INSTALLED	NOT INSTALLED
		R1	100 ohm	100 ohm	NOT INSTALLED
		R63	NOT INSTALLED	NOT INSTALLED	NOT INSTALLED
		R64	NOT INSTALLED	NOT INSTALLED	NOT INSTALLED
		R67	1.3k	1.3k	NOT INSTALLED
	REGULATOR	Q10	5088 npn	5088 npn	NOT INSTALLED
		R65	220k	220k	NOT INSTALLED
		R66	2M	2M	NOT INSTALLED
		C70	NOT INSTALLED	NOT INSTALLED	NOT INSTALLED
	REGULATOR TEMPERATURE COMPENSATION ADJUST	D10	NOT INSTALLED	NOT INSTALLED	NOT INSTALLED
		D11	NOT INSTALLED	NOT INSTALLED	NOT INSTALLED
		D12	NOT INSTALLED	NOT INSTALLED	NOT INSTALLED
		R60	ZERO ohm	ZERO ohm	NOT INSTALLED
		R61	ZERO ohm	ZERO ohm	NOT INSTALLED
		R62	ZERO ohm	ZERO ohm	NOT INSTALLED
CHIP SELECT CONTROL	W/BATTERY BACKUP	C71	NOT INSTALLED	NOT INSTALLED	NOT INSTALLED
		D13	NOT INSTALLED	NOT INSTALLED	NOT INSTALLED
		Q13	2N7002 n-ch	2N7002 n-ch	NOT INSTALLED
		Q12	FDV302P p-ch	FDV302P p-ch	NOT INSTALLED
		Q15	3904 npn	3904 npn	NOT INSTALLED
		R69	100k	100k	NOT INSTALLED
		R68	10k	10k	NOT INSTALLED
		R75	100K	100K	100K
		R73	47k	47k	NOT INSTALLED
		R76	0 ohm	0 ohm	0 ohm
	W/O BATTERY BACKUP	R71	NOT INSTALLED	NOT INSTALLED	ZERO ohm
POWER TO VRAM SWITCH	W/BATTERY BACKUP	Q11	FDV302P p-ch	FDV302P p-ch	NOT INSTALLED
		Q14	3904 npn	3904 npn	NOT INSTALLED
		R70	10k	10k	NOT INSTALLED
		R74	22k	22k	NOT INSTALLED
	W/O BATTERY BACKUP	R72	NOT INSTALLED	NOT INSTALLED	ZERO ohm

CIRCUIT		PART	MODEL EG2000	MODEL EG2020	MODEL EG2100
SRAM	MAIN	U15	128K SRAM	128K SRAM	
	SRAM SELECT	JP4	ZERO ohm ACROSS PINS 1-2	ZERO ohm ACROSS PINS 1-2	ZERO ohm
FLASH	FLASH 1	U10	256K FLASH	256K FLASH	256K FLASH
	FLASH 2	U12	256K FLASH	256K FLASH	256K FLASH
	FLASH SELECT	JP2	ZERO ohm ACROSS PINS 1-2	ZERO ohm ACROSS PINS 1-2	ZERO ohm ACROSS PINS 1-2
		JP3	ZERO ohm ACROSS PINS 1-2	ZERO ohm ACROSS PINS 1-2	ZERO ohm ACROSS PINS 1-2
RS485	MAIN	C39	.1uF	.1uF	NOT INSTALLED
		R34	681 ohm	681 ohm	NOT INSTALLED
		R35	220 ohm	220 ohm	NOT INSTALLED
		R33	681 ohm	681 ohm	NOT INSTALLED
		R32	47k	47k	NOT INSTALLED
		U4	SP483EN	SP483EN	NOT INSTALLED
OPEN COLLECTOR OUTPUTS	MAIN	Q1	4401 npn	4401 npn	NOT INSTALLED
		Q2	4401 npn	4401 npn	NOT INSTALLED
		Q3	4401 npn	4401 npn	NOT INSTALLED
		R36-R38	470 ohm	470 ohm	NOT INSTALLED
		D5-D7	914	914	NOT INSTALLED
		C40-C42	56nF	56nF	NOT INSTALLED

CIRCUIT		PART
	DIGITAL INPUTS	R12-R19
	RS232	U11
LEDS	POWER LED	D51
		R39
MISC	POWER SUPPLY	TVS1
	PROGRAMMING OUT PORT	R56
CONNECTORS		J5
		J6
		J10
		J7
		J11
		U2
		D4
		R9
		R10
		R5
		R47

SIZE	DWG NO.	
C		
SCALE NONE	REV LTR	SHEET OF

REVISION HISTORY			REVISION APPROVAL			
REV	ECO	DESCRIPTION	PROJECT ENGINEER	APPROVAL DATE	DOCUMENT CONTROL	APPROVAL DATE
X1	----	Engineering Prototype Release A/W Rev-A		----	----	----



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 ( ), AND ALL REFERENCES TO THAT VOLTAGE ARE REPRESENTED BY ( )

APPEND THE FOLLOWING DOCUMENTS WHEN CHANGING THIS DOCUMENT:		DRAWING CONTENT:		TITLE	
		DRAWN BY: (INTIAL RELEASE)			
		REVISED BY:			
		APPROVALS: INITIAL RELEASE			
		PROJECT ENGINEER:			
		ENGINEERING MANAGER:		SIZE B	DWG NO.
		SIGNATURES	DATE	SCALE NONE	RELEASE DATE / SHEET OF

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