PRODUCT MANUAL

MiniCore® RCM5700/RCM6700

C-Programmable Ethernet Core Module

User's Manual

90001191_G

MiniCore RCM5700/RCM6700 User's Manual

Part Number 90001191 • Printed in U.S.A.

©2016 Digi International® Inc. • All rights reserved.
Digi International reserves the right to make changes and improvements to its products without providing notice.

Trademarks

Rabbit®, MiniCore, Dynamic C®, Rabbit 5000® and Rabbit 6000® are registered trademarks of Digi International Inc.

Wi-Fi® is a registered trademark of the Wi-Fi Alliance.

The latest revision of this manual is available at www.digi.com.

Digi International Inc.

www.digi.com

TABLE OF CONTENTS

1. Introduction

1.1 RCM5700/RCM6700 Features 6
1.2 Advantages of the RCM5700 and RCM6700 8
1.3 Development and Evaluation Tools 9
1.3.1 Standard Development Kit 9
1.3.2 Deluxe Development Kit 9
1.3.3 Software 10
1.3.4 Online Documentation 10

2. Getting Started

2.1 Install Dynamic C 11
2.2 Hardware Connections 12
2.2.1 Step 1 — Prepare the Interface Board for Development 12
2.2.2 Step 2 — Install Module on Interface Board 13
2.2.3 Step 3 — Connect USB Cable 15
2.3 Starting Dynamic C 17
2.4 Run a Sample Program 17
2.4.1 Troubleshooting 17
2.5 Where Do I Go From Here? 18
2.5.1 Technical Support 18

3. Running Sample Programs

3.1 Introduction 19
3.2 Sample Programs 20
3.2.1 Use of Serial Flash (not supported for RCM5700/RCM5710) 24

4. Hardware Reference

4.1 RCM5700/RCM6700 Digital Inputs and Outputs 26
4.1.1 Memory I/O Interface 33
4.1.2 Other Inputs and Outputs 33
4.2 Serial Communication 34
4.2.1 Serial Ports 34
4.2.2 Ethernet PHY 36
4.2.3 Programming Port 36
4.3 Programming Modes 37
4.3.1 Standalone Operation of the RCM5700/RCM6700 38
4.4 Other Hardware 38
4.4.1 Clocks 38
4.4.2 Spectrum Spreader 38
4.5 Memory 39
4.5.1 RAM 39
4.5.2 Program Flash Memory 39
4.5.3 Mass Storage Serial Flash 39
4.5.4 Encryption RAM Memory 39

5. Software Reference

5.1 More About Dynamic C 40
5.2 Dynamic C Function Calls 42
5.2.1 Digital I/O 42
5.2.2 Serial Communication Drivers 42
5.2.3 User Block 42
5.2.4 RCM5700/RCM6700 Cloning 43
5.2.5 TCP/IP Drivers 43
5.2.6 Serial Flash Drivers (not supported on RCM5700/RCM5710) 43
5.3 Upgrading Dynamic C 43
5.3.1 Add-On Modules 43

Appendix A. RCM5700/RCM6700 Specifications

A.1 Electrical and Mechanical Characteristics .. 45
A.1.1 mini PCI Express Connector Design Recommendations 52
A.2 Rabbit 5000 and 6000 Microprocessor Characteristics 54
A.3 Jumper Configurations 55

Appendix B. Interface Board

B.1 Introduction 58
B.1.1 Interface Board Features 59
B.2 Mechanical Dimensions and Layout 60
B.2.1 Headers 61
B.3 Ethernet 62
B.3.1 RJ-45 62
B.3.2 RCM6700 LEDs 62
B.4 Power Supply 63
B.5 Using the Interface Board 64
B.5.1 Add Additional Boards 65
B.6 Interface Board Jumper Configurations 66

Appendix C. Prototyping Board

C.1 Introduction 68
C.1.1 Prototyping Board Features 68
C.2 Mechanical Dimensions and Layout 69
C.2.1 Headers 71
C.3 Using the Prototyping Board 72
C.3.1 Add Additional Boards 73

Appendix D. Digital I/O Accessory Board

D.1 Introduction 75
D.1.1 Digital I/O Accessory Board Features 75
D.2 Mechanical Dimensions and Layout 76
D.2.1 Headers 77
D.3 Using the Digital I/O Accessory Board..... 78
D.3.1 Configuration 79
D.3.2 Add Additional Boards 81

Appendix E. Serial Communication Accessory Board

E.1 Introduction..... 83
E.1.1 Serial Communication Accessory Board Features 83
E.2 Mechanical Dimensions and Layout 84
E.2.1 Headers 85
E.3 Using the Serial Communication Accessory Board..... 86
E.3.1 Configuration 87
E.3.2 Add Additional Boards 89

Appendix F. Using the TCP/IP Features

F.1 TCP/IP Connections..... 90
F.2 TCP/IP Primer on IP Addresses..... 92
F.2.1 IP Addresses Explained 94
F.2.2 How IP Addresses are Used 95
F.2.3 Dynamically Assigned Internet Addresses 96
F.3 Placing Your Device on the Network 97
F.4 Running TCP/IP Sample Programs 98
F.4.1 How to Set IP Addresses in the Sample Programs 99
F.4.2 How to Set Up your Computer for Direct Connect 100
F.5 Run the PINGME.C Sample Program 101
F.6 Running Additional Sample Programs With Direct Connect 102
F.7 Where Do I Go From Here?..... 103

Appendix G. Power Supply

G.1 Power Supplies..... 104
G.1.1 Battery Backup for the RCM5700/RCM5710 and RCM6700 Family . 105
G.1.2 Battery-Backup Circuit 106
G.1.3 Reset Generator 106

1. INTRODUCTION

The RCM5700 and RCM6700 are compact modules in a mini PCI Express form factor, and incorporate the powerful Rabbit® 5000 and 6000 microprocessors with integrated 10/100Base-T Ethernet functionality and onchip SRAM. The Rabbit® 5000 and 6000 microprocessor features include hardware DMA, I/O lines shared with up to six serial ports and four levels of alternate pin functions that include variable-phase PWM, an external I/O bus, quadrature decoder, and input capture. This equates to a MiniCore module that is fast, efficient, and the ideal solution for a wide range of embedded applications.

Development Kits are available with the essentials that you need to design your own microprocessor-based system, and includes a complete Dynamic C software development system. The Development Kit also contains an Interface Board with USB and Ethernet connections that will allow you to evaluate the RCM5700 or RCM6700, and a Prototyping Board to help you to develop your own applications. You will also be able to write and test software for the RCM5700/RCM6700 modules, including Ethernet applications.

The RCM5700 has a Rabbit 5000 microprocessor operating at up to 50.0 MHz, flash memory, two clocks (main oscillator and real-time clock), and the circuitry necessary to reset and manage the Rabbit 5000. The RCM6700 has a Rabbit 6000 processor operating at up to 187.5 MHz and 1MB of internal SRAM. An edge connector brings out the MiniCore module user interface to a 52-pin mini PCI Express socket on the motherboard the core module is mounted on.

The RCM5700/RCM6700 receives its +3.3 V power from the motherboard on which it is mounted. The MiniCore module can interface with other CMOS-compatible digital devices through the motherboard.

1.1 RCM5700/RCM6700 Features

- Small size: ranges from 1.20" × 2.00" × 0.12" (30 mm × 51 mm × 3 mm) for the RCM5700/RCM6700 model to 1.20" × 2.00" × 0.73" (30 mm × 51 mm × 19 mm) for the RCM5760/RCM6760 model
- Microprocessor: RCM5700: Rabbit 5000 running at 50.0 MHz, RCM6700: Rabbit 6000 running at 187.5 MHz
- Up to 35 general-purpose I/O lines each configurable with up to four alternate functions
- 3.3 V I/O lines
- Six CMOS-compatible serial ports — four ports are configurable as a clocked serial port (SPI), and two ports are configurable as SDLC/HDLC serial ports
- Ethernet PHY interface chooses Ethernet interface automatically based on whether a crossover cable or a straight-through cable is used in a particular setup
- External I/O bus can be configured for 8 data lines, 8 address lines (shared with parallel I/O lines), and I/O read/write
- **RCM5700:** 128KB SRAM (on Rabbit 5000 chip) and 1MB flash memory, with 2MB serial flash and 512KB SRAM on RCM5750/RCM5760 models.

  RCM6700: 1MB SRAM (on Rabbit 6000 chip) and 1MB serial flash, with 4MB serial flash and additional 1MB external SRAM on RCM6750/6760 models.
- Real-time clock (battery-backable on all models except the RCM5750/5760)
- Watchdog supervisor

There are four RCM5700 production models. Table 1-1 summarizes their main features.

Table 1-1. RCM5700 Features

Feature	RCM5700	RCM5710	RCM5750	RCM5760
Microprocessor	Rabbit 5000 at 50.0 MHz			
Ethernet Port	10/100Base-T PHY only	10/100Base-T, RJ-45, 2 LEDs	10/100Base-T PHY only	10/100Base-T, RJ-45, 2 LEDs
External SRAM	—		512KB (8-bit)	
SRAM	128KB (Rabbit 5000 onchip)			
Flash Memory (program)	1MB			
Flash Memory (mass data storage)	—		2MB (serial flash)	
Serial Ports	6 shared high-speed, CMOS-compatible ports: 6 are configurable as asynchronous serial ports; 4 are configurable as clocked serial ports (SPI); 2 are configurable as SDLC/HDLC serial ports; 1 asynchronous serial port is used during programming			

The RCM5700 and RCM5750 models already have an Ethernet PHY device, the Integrated Circuit Systems ICS1893BK. The RCM5710 and RCM5760 are identical to the RCM5700 and RCM5750 respectively, except that they have an integrated 10/100 Base-T magnetic RJ-45 jack with two LEDs on the MiniCore printed circuit board.

There are four RCM6700 production models. Table 1-2 summarizes their main features.

Table 1-2. RCM6700 Features

Feature	RCM6700	RCM6710	RCM6750	RCM6760
Microprocessor	Rabbit 6000 at 162.5 MHz			
Ethernet Port	10/100Base-T PHY only	10/100Base-T, RJ-45, 2 LEDs	10/100Base-T PHY only	10/100Base-T, RJ-45, 2 LEDs
External SRAM	—		1MB (16-bit)	
RAM	**NOTE:** 1MB (Rabbit 6000 onchip)			
SRAM	32KB (Rabbit 6000 onchip, battery-backable)			
Flash Memory (mass data storage)	1MB (serial flash)		4 MB (serial flash)	
Serial Ports	6 shared high-speed, CMOS-compatible ports: 6 are configurable as asynchronous serial ports; 4 are configurable as clocked serial ports (SPI); 2 are configurable as SDLC/HDLC serial ports; 1 asynchronous serial port is used during programming			

The Rabbit 6000 has an Ethernet PHY built into the processor. The RCM6710 and RCM6760 are identical to the RCM6700 and RCM6750 respectively, except that they have an integrated 10/100 Base-T magnetic RJ-45 jack with two LEDs on the MiniCore printed circuit board.

Placing the RJ-45 Ethernet jack on the motherboard, which is what must be done with the RCM5700/RCM6700 and RCM5750/RCM6750 models, provides significant design advantages because the core module may then be placed anywhere on the motherboard. Rabbit's Technical Note TN266, *PCB Layout for the Ethernet PHY Interface*, provides complete design recommendations. If the RJ-45 Ethernet jack is on the MiniCore printed circuit board, as it is on the RCM5710/RCM5760/RCM6710/RCM6760 modules, the edge of the MiniCore module with the Ethernet jack must be positioned along the edge of the motherboard to allow access to the jack since the entire assembly will likely be inside a customer-designed enclosure.

All RCM5700/RCM6700 models are programmed through a USB connector on the motherboard using a USB cable supplied with the Development Kit. The RCM5750/RCM5760 may also be programmed remotely over an Ethernet link using the Remote Program Update library with Dynamic C v. 10.56 or later. The RCM67xx series may also be used with the Remote Program Update with Dynamic C v. 10.64 or later. See Application Note AN421, *Remote Program Update*, for more information.

NOTE: The RabbitLink cannot be used to program the RCM5700 or RCM6700.

Appendix A provides detailed specifications for the RCM5700 and RCM6700.

1.2 Advantages of the RCM5700 and RCM6700

- Fast time to market using a fully engineered, “ready-to-run/ready-to-program” microprocessor core.
- Competitive pricing when compared with the alternative of purchasing and assembling individual components.
- Easy C-language program development and debugging
- Rabbit Field Utility to download compiled Dynamic C .bin files.
- Generous memory size allows large programs with tens of thousands of lines of code, and substantial data storage.
- Reference design allows integrated Ethernet port for network connectivity, with royalty-free TCP/IP software.

1.3 Development and Evaluation Tools

1.3.1 Standard Development Kit

The RCM5700/RCM6700 Standard Development Kit contains the hardware essentials you will need to use your RCM5700 or RCM6700 module. These items are supplied in the standard version of the Development Kit.

- RCM5700 or RCM6700 module.
- Interface Board with standoffs/connectors.
- Prototyping Board with standoffs/connectors.
- USB cable to program MiniCore via Interface Board.
- *Dynamic C* CD-ROM, including product documentation on disk.
- ***Getting Started*** instructions.
- Registration card.

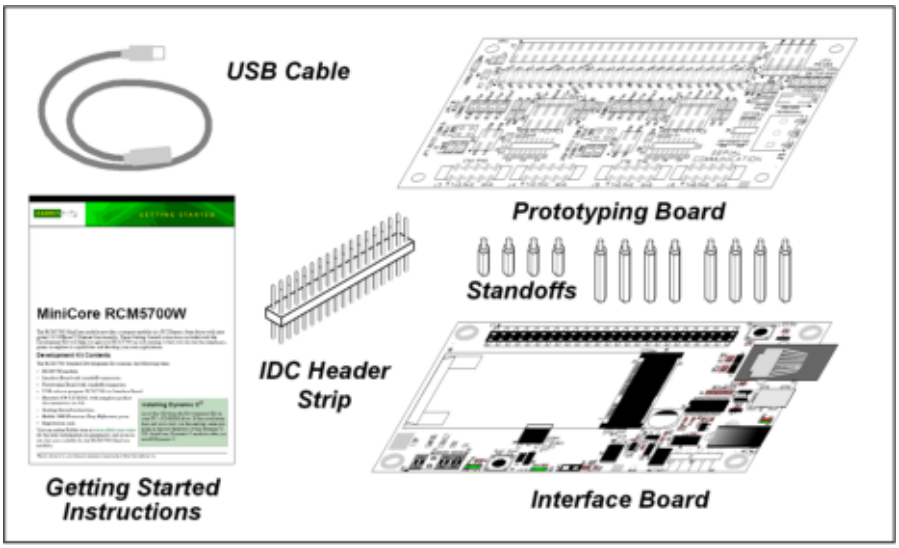


Figure 1-1. Standard Development Kit

1.3.2 Deluxe Development Kit

In addition to the items included in the standard Development Kit, the Deluxe Development Kit contains the following items.

- Universal AC adapter, 5 V DC, 2 A (includes Canada/Japan/U.S., Australia/N.Z., U.K., and European style plugs). Development Kits sold in North America may contain an AC adapter with only a North American style plug.
- Digital I/O and Serial Communication accessory boards for use with certain sample programs.
- CAT 5/6 Ethernet cable and DB9 to 10-pin header serial cable.
- *Rabbit 5000 or 6000 Processor Easy Reference* poster.

1.3.3 Software

The RCM5700/RCM5710 is programmed using version 10.44 or later of Dynamic C; the RCM5750/RCM5760 requires version 10.56 or later of Dynamic C; and the RCM6700 family requires version 10.64 or later. A compatible version is included on the Development Kit CD-ROM.

In addition to the Web-based technical support included at no extra charge, a one-year telephone-based technical support subscription is also available for purchase. For further information and complete documentation, contact your Rabbit sales representative or authorized distributor, or visit our Web site at www.digi.com.

1.3.4 Online Documentation

The online documentation is installed along with Dynamic C, and an icon for the documentation menu can be placed on the workstation's desktop. Double-click this icon to reach the menu. If the icon is missing, use your browser to find and load **default.htm** in the **docs** folder, found in the Dynamic C installation folder.

The latest versions of all documents are always available for download from our web site.

2. GETTING STARTED

This chapter describes the RCM5700/RCM6700 hardware in more detail, and explains how to set up and use the accompanying Interface Board.

NOTE: This chapter (and this manual) assume that you have the RCM5700/RCM6700 Development Kit. If you purchased an MiniCore module by itself, you will have to adapt the information in this chapter and elsewhere to your test and development setup.

2.1 Install Dynamic C

Dynamic C must be installed to develop and debug programs for the RCM5700/RCM6700 series of modules (and for all other Rabbit hardware). The latest version of Dynamic C is included on CD in the Development Kit and is also available for download on our website at www.digi.com.

If you have not yet installed Dynamic C, do so now by inserting the Dynamic C CD from the Development Kit in your PC's CD-ROM drive. If autorun is enabled, the CD installation will begin automatically.

If autorun is disabled or the installation 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. Most steps of the process are self-explanatory.

Once your installation is complete, you will have up to three new icons on your PC desktop. One icon is for Dynamic C, another opens the documentation menu, and the third is for the Rabbit Field Utility, a tool used to download precompiled software to a target system.

If you have purchased any of the optional Dynamic C modules, install them after installing Dynamic C. The modules may be installed in any order. You must install the modules in the same folder where Dynamic C was installed.

2.2 Hardware Connections

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

1. Insert standoffs/connectors on the Interface Board.
2. Install the MiniCore module on the Interface Board.
3. Connect the USB cable between the Interface Board and the workstation PC.

2.2.1 Step 1 — Prepare the Interface Board for Development

Insert a short plastic standoff supplied from the Development Kit in one of the corner holes from the bottom of the Interface Board, then secure it with a long plastic standoff from above as shown in Figure 2-2. Repeat this step so that plastic standoffs/connectors are in place at all four positions.

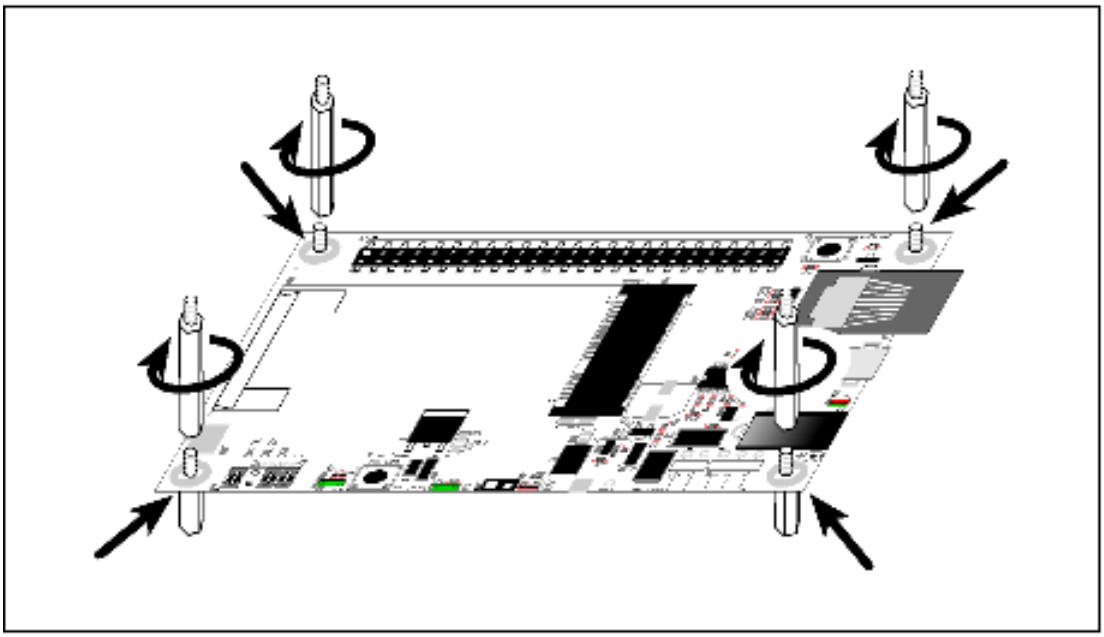


Figure 2-2. Insert Standoffs/Connectors

2.2.2 Step 2 — Install Module on Interface Board

Position the MiniCore module with the edge connectors facing the mini PCI Express socket J1A at an angle as shown in Figure 2-3 below. Insert the edge connectors into the mini PCI Express socket J1A, then press down on the opposite edge of the MiniCore module to snap it into place in holder J1B.

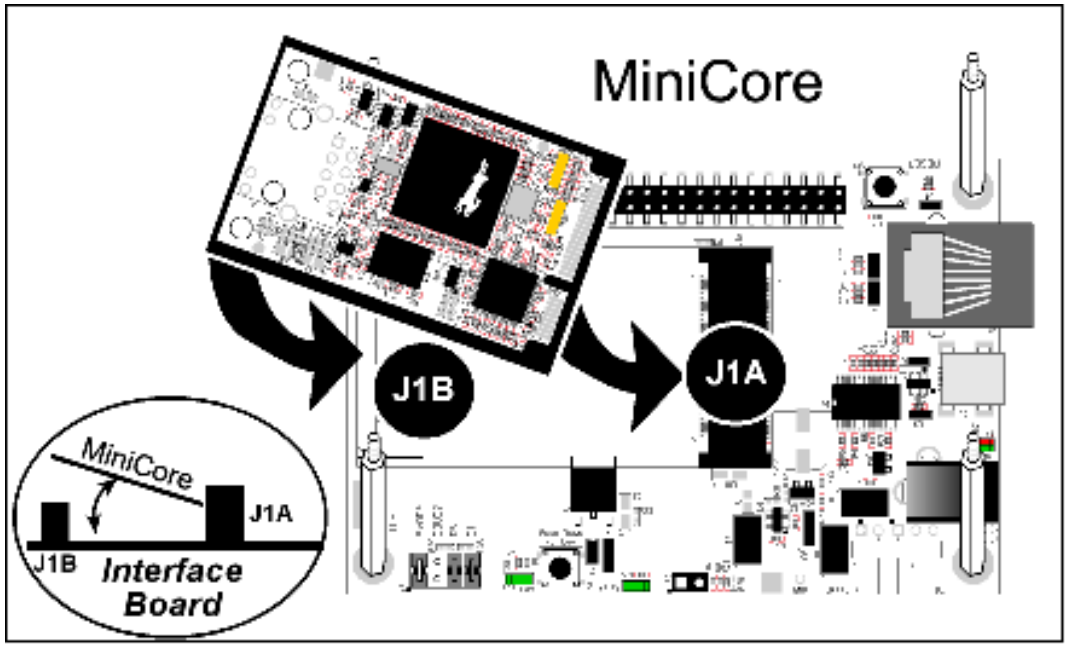
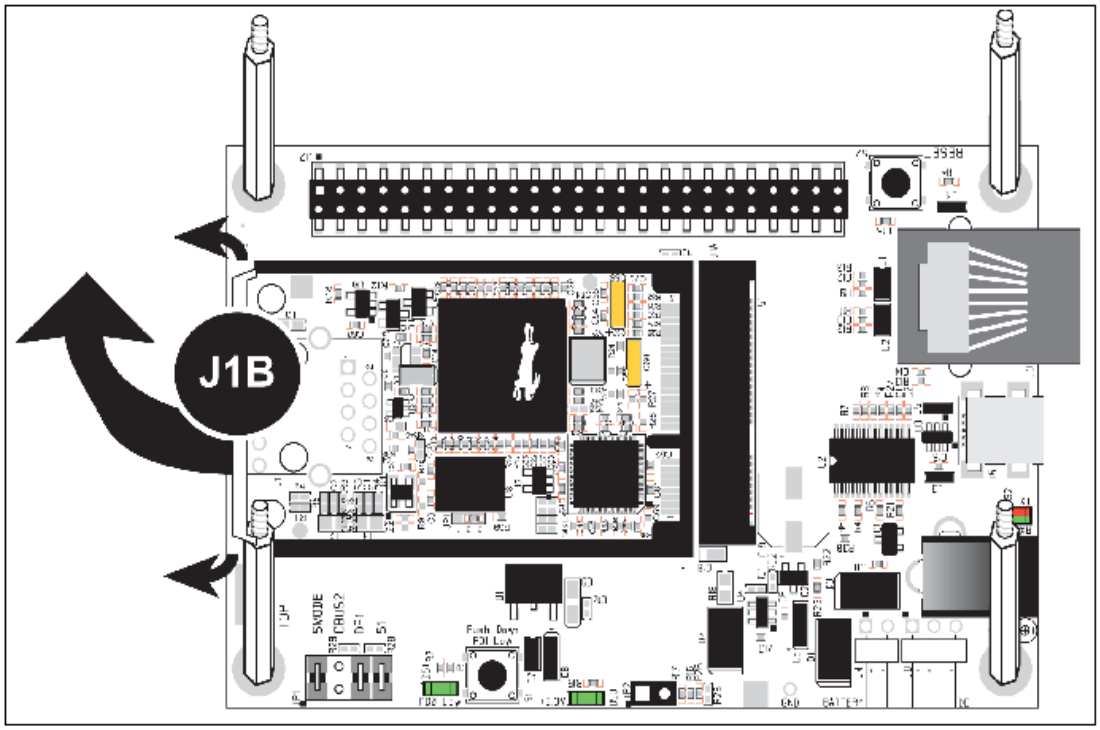


Figure 2-3. Install the MiniCore Module on the Interface Board

Should you need to remove the MiniCore module, use two fingernails to hold back the spring clip at J1B from the two MiniCore corners, lift up the edge of the MiniCore above J1B, then pull the MiniCore away to remove the edge connectors from the mini PCI Express socket.



CAUTION:
Remove power before attempting to insert or remove the MiniCore in the mini PCI Express socket.

2.2.3 Step 3 — Connect USB Cable

The USB cable connects the RCM5700/RCM6700 to the PC running Dynamic C to download programs and to monitor the MiniCore module during debugging. It also supplies power to the Interface Board and the MiniCore via the USB interface.

Connect the USB cable between USB connector J5 on the Interface Board and your PC as shown in Figure 2-4. Note that the USB cable connectors are different at either end, so there is only one way to connect them between the PC and the Interface Board.

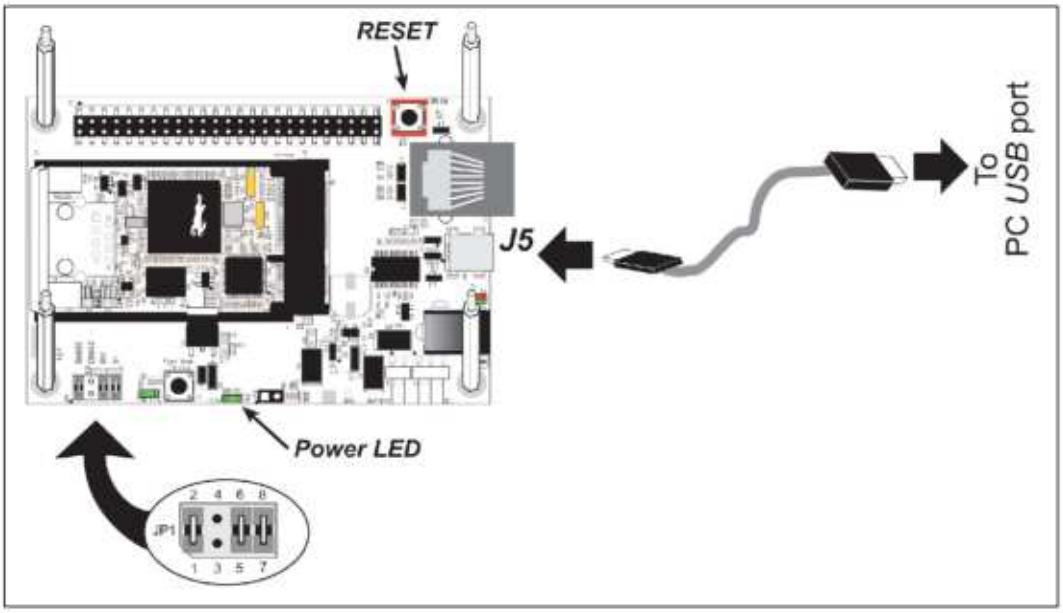


Figure 2-4. Connect USB Cable

Your PC should recognize the new USB hardware, and the LEDs next to the USB connector on the Interface Board will flash — if you get an error message, you will have to install USB drivers. Drivers for Windows XP are available in the Dynamic C `Drivers\Rabbit USB Programming Cable\WinXP_2K` folder — double-click `DPInst.exe` to install the USB drivers. Drivers for other operating systems are available online at www.ftdichip.com/Drivers/VCP.htm.

The green power LED on the Interface Board should light up when you connect the USB cable. The MiniCore and the Interface Board are now ready to be used.

NOTE: A **RESET** button is provided on the Interface Board above the Ethernet jack to allow a hardware reset without disconnecting power.

NOTE: Pins 1–2 on header JP1 on the Interface Board must be jumpered to download and debug applications and sample programs with Dynamic C running. Pins 1–2 should be left unjumpered to *run* an program already loaded in flash memory.

CAUTION: Do not jumper pins 1–3 on header JP1 on the Interface Board.

2.2.3.1 Alternate Power Supply Connections — Deluxe Development Kit

The deluxe Development Kit contains a separate AC adapter that may be used to supply power to the Interface Board and the RCM5700/RCM6700 when the USB cable is not connected or when more power is needed than the 500 mA the USB cable is able to supply. The AC adapter may also be used to supply power when the USB cable is connected, in which case the power supply through the USB cable will be disconnected automatically.

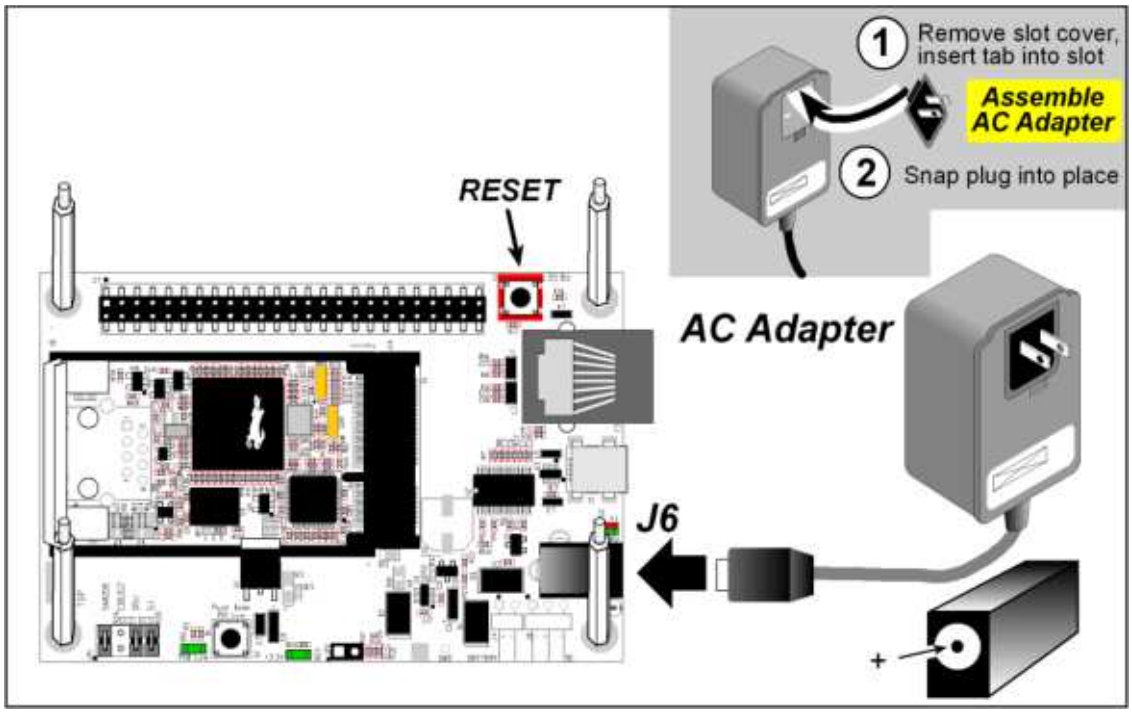


Figure 2-5. Alternate Power Supply Connections—Deluxe Development Kit

First, prepare the AC adapter for the country where it will be used by selecting the plug. The deluxe Development Kit presently includes Canada/Japan/U.S., Australia/N.Z., U.K., and European style plugs. Snap in the top of the plug assembly into the slot at the top of the AC adapter as shown in Figure 2-5, then press down on the spring-loaded clip below the plug assembly to allow the plug assembly to click into place. Release the clip to secure the plug assembly in the AC adapter.

Connect the AC adapter to DC input jack J6 on the Interface Board as shown in Figure 2-5. Plug in the AC adapter. The green power LED on the Interface Board should light up. The MiniCore and the Interface Board are now ready to be used.

Note that the center pin of J6 is positive.

2.3 Starting Dynamic C

If you already have Dynamic C installed, you are now ready to test your programming connections by running a sample program. Start Dynamic C by double-clicking on the Dynamic C icon on your desktop or in your **Start** menu. Select **Store Program in Flash** on the "Compiler" tab in the Dynamic C **Options > Project Options** menu. Then click on the "Communications" tab and verify that **Use USB to Serial Converter** is selected to support the USB cable. Click **OK**.

You may have to select the COM port assigned to the USB cable on your PC. In Dynamic C, select **Options > Project Options**, then select this COM port on the "Communications" tab, then click **OK**.

2.4 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, compile it using the **Compile** menu, and then run it by selecting **Run** in the **Run** menu. The **STDIO** window will open on your PC and will display a small square bouncing around in a box.

This program shows that the CPU is working.

2.4.1 Troubleshooting

If you receive the message `Could Not Open Serial Port`, check that the COM port assigned to the USB cable was identified and set up in Dynamic C as described above. This same error occurs when Windows has already allocated the COM port to another process.

If you receive the message `No Rabbit Processor Detected`, the USB cable may be connected to the wrong COM port, or the connection may be faulty. First, check both ends of the USB cable to ensure that it is firmly plugged into the PC and the USB connector in the Interface Board. Ensure that the module is firmly and correctly installed in its connector on the Interface Board.

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

- Locate the **Serial Options** dialog on the "Communications" tab in the Dynamic C **Options > Project Options** menu. Select a slower Max download baud rate. Click **OK** to save.

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 on the "Communications" tab in the Dynamic C **Options > Project Options** menu. Choose a lower debug baud rate. Click **OK** to save.

Press **<Ctrl-Y>** to force Dynamic C to recompile the BIOS. You should receive a `Bios compiled successfully` message once this step is completed successfully.

2.5 Where Do I Go From Here?

If the sample program ran fine, you are now ready to go on to other sample programs and to develop your own applications. The source code for the sample programs is provided to allow you to modify them for your own use. The ***RCM5700/RCM6700 User's Manual*** also provides complete hardware reference information for the RCM5700/RCM6700, the Interface Board, the Prototyping Board, and the accessory boards in the Deluxe Development Kit.

For advanced development topics, refer to the ***Dynamic C User's Manual***.

2.5.1 Technical Support

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

If there are any problems at this point:

- Use the Dynamic C **Help** menu to get further assistance with Dynamic C.
- Check the Rabbit Technical Bulletin Board and forums at www.digi.com/support//forum/.
- Use the Technical Support e-mail form at www.digi.com/support/.

3. RUNNING SAMPLE PROGRAMS

To develop and debug programs for the RCM5700/RCM6700 (and for all other Rabbit hardware), you must install and use Dynamic C. This chapter provides a tour of its major features with respect to the RCM5700/RCM6700.

3.1 Introduction

To help familiarize you with the RCM5700/RCM6700 modules, Dynamic C includes several sample programs. Loading, executing and studying these programs will give you a solid hands-on overview of the RCM5700/RCM6700's capabilities, as well as a quick start with Dynamic C as an application development tool.

> **NOTE:** The sample programs assume 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.

In order to run the sample programs discussed in this chapter and elsewhere in this manual,

1. Your RCM5700/RCM6700 must be installed on the Interface Board as described in Chapter 2, "Getting Started."
2. Dynamic C must be installed and running on your PC.
3. The USB cable must connect the Interface Board to your PC.
4. Power must be applied to the RCM5700/RCM6700 through the Interface Board.

Refer to Chapter 2, "Getting Started," if you need further information on these steps.

To run a sample program, open it with the **File** menu (if it is not still open), then compile and run it by selecting **Run** in the Run menu (or press **F9**). The RCM5700/RCM6700 must be in Program Mode (see Figure 4-12) and must be connected to a PC using the USB cable.

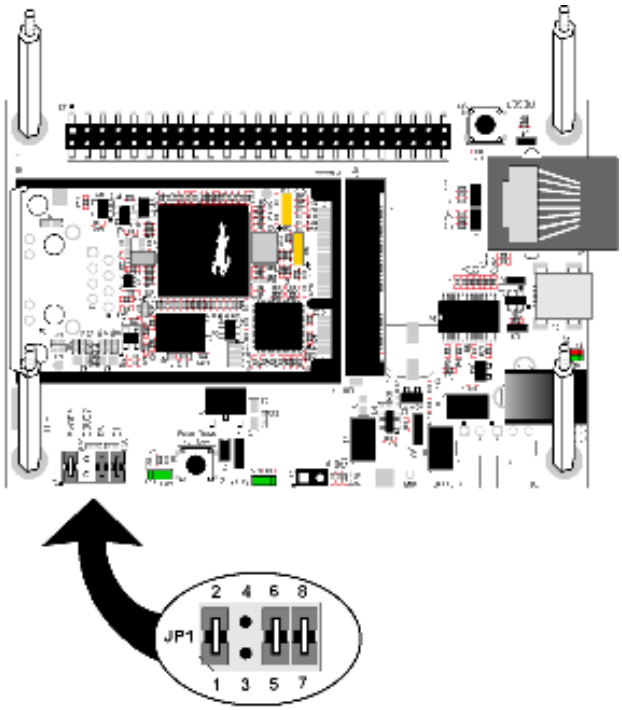
Complete information on Dynamic C is provided in the ***Dynamic C User's Manual***.

3.2 Sample Programs

Of the many sample programs included with Dynamic C, several are specific to the RCM5700/RCM6700. These programs will be found in the `SAMPLES\RCM5700 or SAMPLES\RCM6700 folder`, depending on your MiniCore model. Sample programs in the `SAMPLES` folder one level up are generally generic samples that can be run on any Rabbit-based product.

Before you compile and run the following sample programs, make sure that pins 1–2, 5–6, and 7–8 on header JP1 of the Interface Board are jumpered. Each sample program has comments that describe the purpose and function of the program. Follow the instructions at the beginning of the sample program.

CAUTION: Do not jumper pins 1–3 on header JP1 on the Interface Board.



- `FLASHLED01.C`—demonstrates the use of costatements to flash LED DS1 on the Interface Board. PD0 on the MiniCore is used to drive the LED.
- `FLASHLED01A.C`—demonstrates the use of a state machine with standard C constructs to flash LED DS1on the Interface Board. PD0 on the MiniCore is used to drive the LED.
- `FLASHLED02.c`—demonstrates the use of costatements to detect and debounce switch S1 being pressed on the Interface Board. LED DS1 flashes on the Interface Board, and the rate at which it flashes changes each time you press switch S1. I/O control for the LED is provided by PD0 on the MiniCore, and PD1 detects the activity on switch S1.
- `FLASHLED02A.c`—detects and debounces switch S1 being pressed on the Interface Board. LED DS1 flashes on the Interface Board, and the rate at which it flashes changes each time you press switch S1. I/O control for the LED is provided by PD0 on the MiniCore, and PD1 detects the activity on switch S1.

The Digital I/O accessory board needs to be installed to run the `SWITCHLEDS.C` and the `SERIALTOSERIAL.C` sample programs. This accessory board is included only with the Deluxe Development Kit.

To install the Digital I/O accessory board, insert the strip of header pins included with the accessory board into the socket at J12 on the bottom side of the Digital I/O accessory board. Then line up the Digital I/O accessory board with the Interface Board standoffs/connectors and install the Digital I/O accessory board pins into socket J2 on the Interface Board. Secure the Digital I/O accessory board with the long plastic standoffs/connectors from above as shown in Figure 3-6.

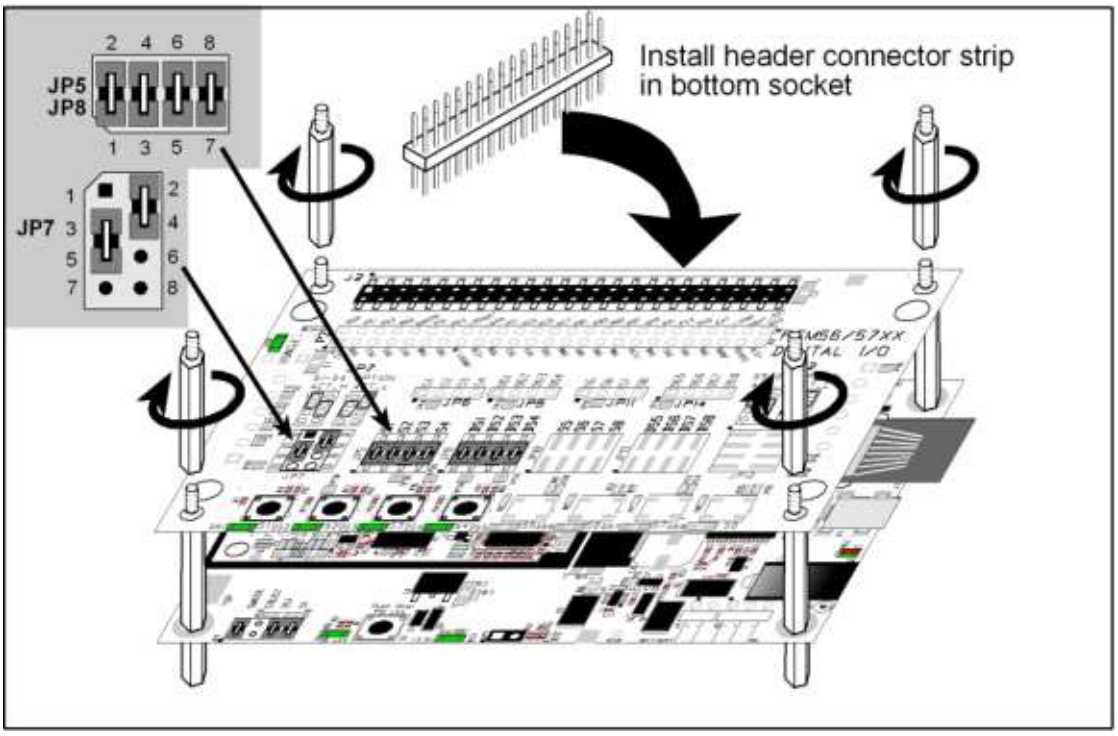


Figure 3-6. Install Digital I/O Accessory Board

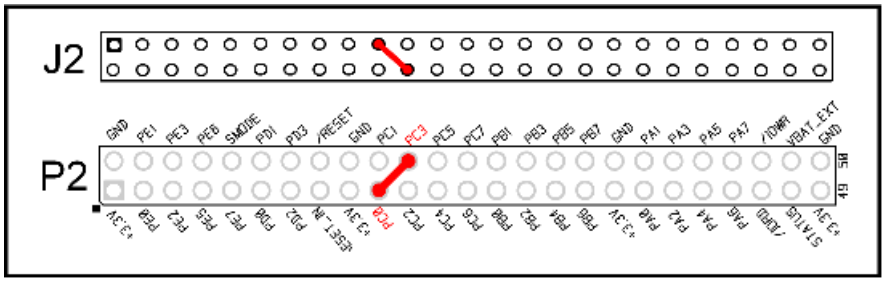
Pins 1–2, 3–4, 5–6, and 7–8 on headers JP5 and JP8 on the Digital I/O accessory board must be jumpered. Pins 2–4 and 3–5 on header JP7 on the Digital I/O accessory board must also be jumpered.

- `SWITCHLEDS.C`—monitors switches S1, S2, S3, and S4 on the Digital I/O accessory board and lights LEDs DS1–DS4 when the corresponding pushbutton switch is pressed. LEDs DS1–DS2 on the Digital I/O accessory board are controlled by PA4–PA7, and switches S1–S4 are controlled by PB4–PB7 respectively.

The `SERIALTOSERIAL.C` sample program is in the `SAMPLES\RCM5700\SERIAL` or `SAMPLES\RCM6700\SERIAL` folder, depending on your MiniCore model.

- `SERIALTOSERIAL.C`—monitors switches S1, S2, S3, and S4 on the Digital I/O accessory board and lights LEDs DS1–DS4 when the corresponding pushbutton switch is pressed. LEDs DS1–DS2 on the Digital I/O accessory board are controlled by PA4–PA7, and switches S1–S4 are controlled by PB4–PB7 respectively. The sample program sends messages from Serial Port D to Serial Port C to indicate that a switch was pressed. Messages received by Serial Port C are displayed in Dynamic C's **STDIO** window.

Before you compile and run this sample program, you will need to connect J2 pin 19 (PC0/TxD) to J2 pin 22 (PC3/RxC) *or* the corresponding holes on P2.



If you are using the Serial Communication Accessory Board, you should connect pin 3 (TXD) on header J3 to pin 5 (RXC) on header J4 instead.

The Serial Communication accessory board needs to be installed to run the following serial sample program in the `SAMPLES\RCM5700\SERIAL` or `SAMPLES\RCM6700\SERIAL` folder, depending on your MiniCore model. This accessory board is included only with the Deluxe Development Kit.

To install the Serial Communication accessory board, insert the strip of header pins included with the accessory board into the socket at J12 on the bottom side of the Serial Communication accessory board. Then line up the Serial Communication accessory board with the Interface Board or Digital I/O accessory board standoffs/connectors and install the Serial Communication accessory board pins into socket J2 on the Interface Board or the Digital I/O accessory board. Secure the Serial Communication accessory board with the long plastic standoffs/connectors from above as shown in Figure 3-6.

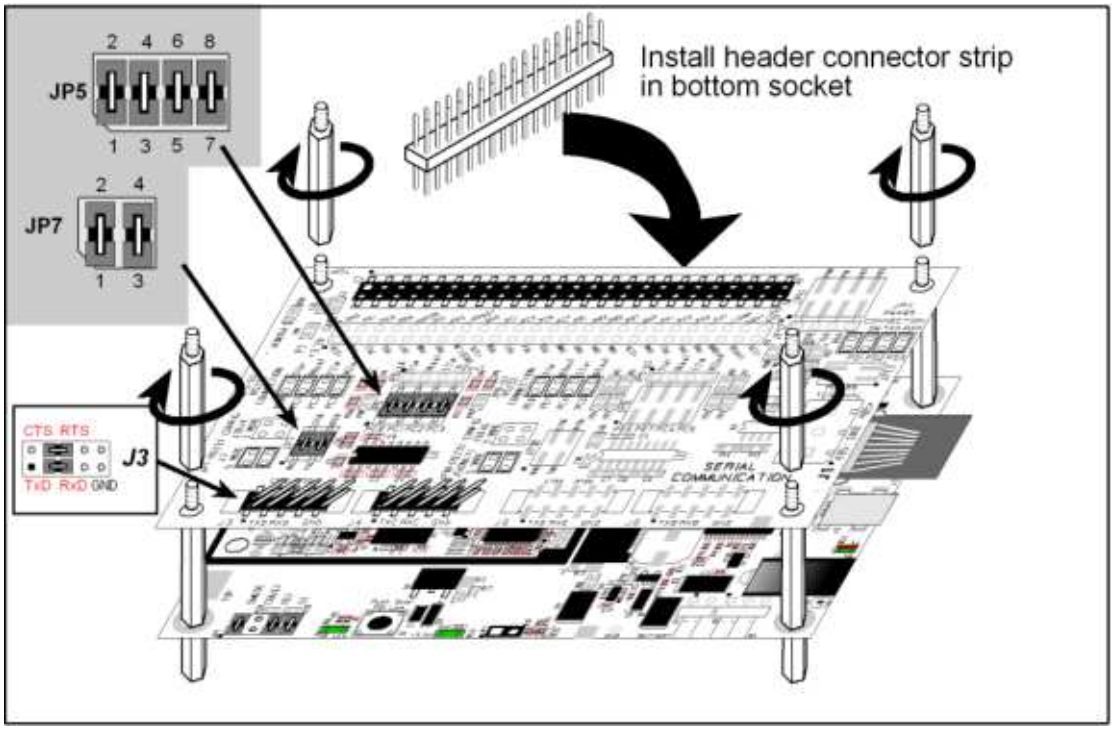


Figure 3-7. Install Serial Communication Accessory Board

Pins 1–2, 3–4, 5–6, and 7–8 on header JP5 on the Serial Communication accessory board must be jumpered. Pins 1–2 and 3–4 on header JP7 on the Serial Communication accessory board must also be jumpered.

- `SIMPLE5WIRE.C`—This program demonstrates 5-wire RS-232 serial communication with flow control on Serial Port C and data flow on Serial Port D.

  To set up the Serial Communication accessory board, you will need to tie TxD and RxD on the RS-232 header at J3, then tie CTS and RTS, also on J3, using jumpers as shown in Figure 3-7.

  Once you have compiled and run this program, you can test flow control by disconnecting the CTS jumper from RTS while the program is running. Characters will no longer appear in the **STDIO** window, and will display again once CTS is connected back to RTS.

Once you have loaded and executed these sample programs and have an understanding of how Dynamic C and the RCM5700/RCM6700 modules interact, you can move on and try the other sample programs, or begin developing your own application.

3.2.1 Use of Serial Flash (not supported for RCM5700/RCM5710)

The following sample programs from the `SAMPLES\RCM5700\Serial_Flash` folder may be used with the RCM5750/RCM5760 models.

- `SERIAL_FLASHLOG.C`—This program runs a simple Web server and stores a log of hits on the home page of the serial flash "server." This log can be viewed and cleared from a browser at http://10.10.6.100/. You will likely have to first "configure" your network interface card for a "10Base-T Half-Duplex," "100Base-T Half-Duplex," or an "Auto-Negotiation" connection on the "Advanced" tab, which is accessed from the control panel (**Start > Settings > Control Panel**) by choosing **Network Connections**.
- `SFLASH_INSPECT.C`—This program is a handy utility for inspecting the contents of a serial flash chip. When the sample program starts running, it attempts to initialize a serial flash chip on Serial Port B. Once a serial flash chip is found, the user can perform five different commands to print out the contents of a specified page, print out the contents of a range of pages, clear (set to zero) all the bytes in a specified page, set all bytes on the specified page to a given value, or save user-specified text to a selected page.

4. HARDWARE REFERENCE

Chapter 4 describes the hardware components and principal hardware subsystems of the RCM5700/RCM6700. Appendix A, "RCM5700/RCM6700 Specifications," provides complete physical and electrical specifications.

Figure 4-8 shows the Rabbit-based subsystems designed into the RCM5700/RCM6700.

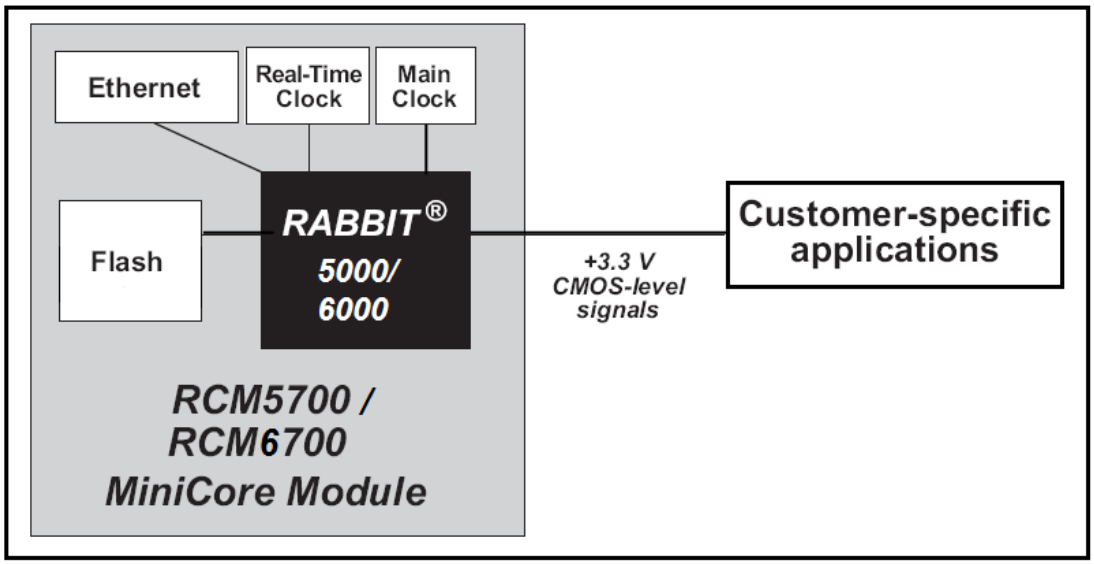


Figure 4-8. RCM5700/RCM6700 Subsystems

4.1 RCM5700/RCM6700 Digital Inputs and Outputs

Figure 4-9 shows the RCM5700/RCM6700 pinouts for the edge connector.

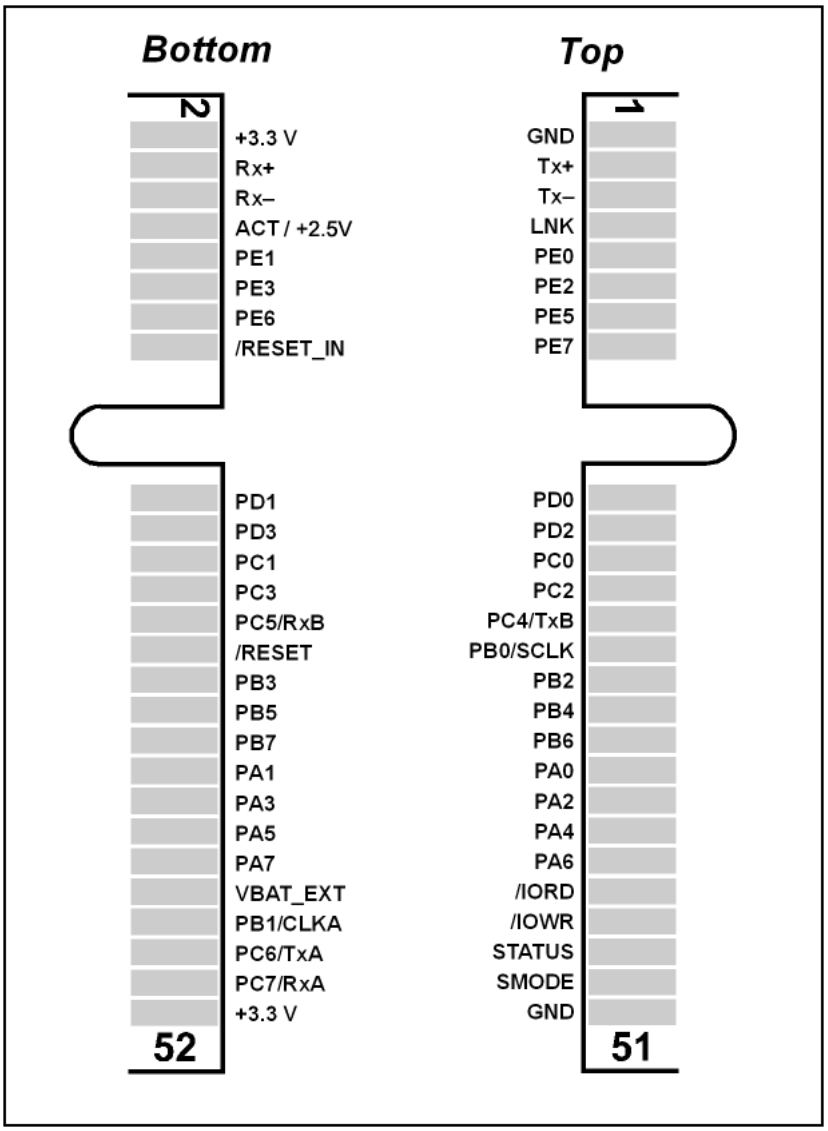


Figure 4-9. RCM5700/RCM6700 Pinouts

The edge connectors are designed to interface with a 52-pin mini PCI Express socket.

Pin 8 has different functionality between the RCM5700 and RCM6700, due to differences in the Ethernet PHY designs. On the RCM5700, it is the activity LED signal, while on the RCM6700 it is +2.5V for the Ethernet interface.

Figure 4-10 shows the use of the Rabbit 5000/6000 microprocessor ports in the RCM5700/ RCM6700 modules.

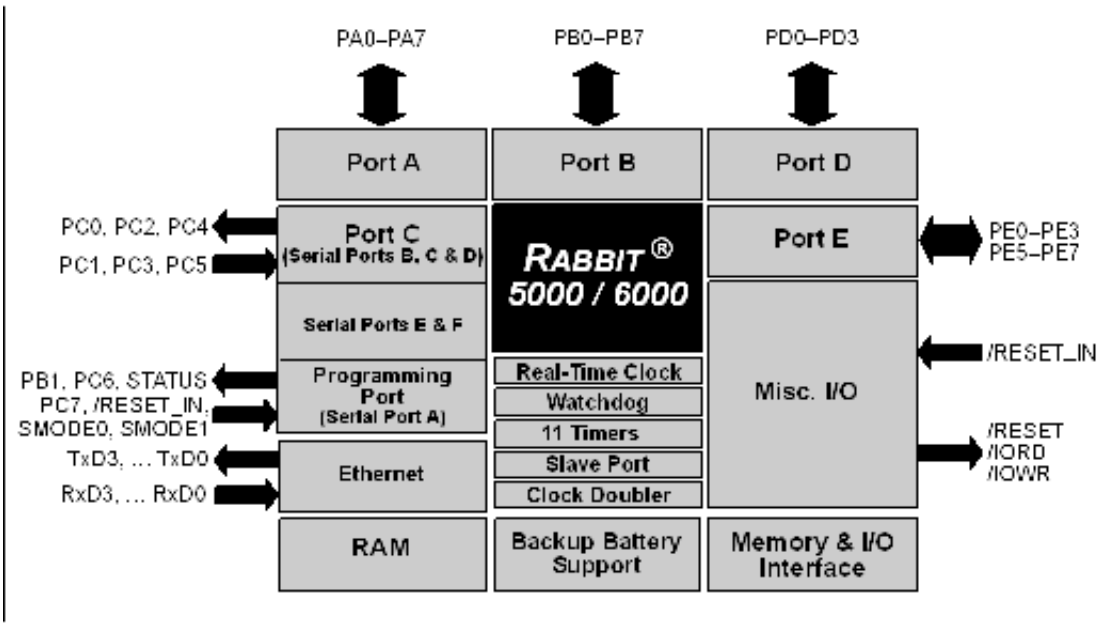


Figure 4-10. Use of Rabbit 5000/6000 Ports

The ports on the Rabbit microprocessor used in the RCM5700/RCM6700 are configurable, and so the defaults can be reconfigured. Table 4-3 lists the Rabbit factory defaults and the alternate configurations.

Table 4-3. RCM5700/RCM6700 Pinout Configurations

Pin	Pin Name	Default Use	Alternate Use	Notes
1	GND			
2	+3.3 V			
3	Tx+	Ethernet		Not connected on the RCM5710/5760/6710/6760
4	Rx+			
5	Tx–			
6	Rx–			
7	LNK			Requires external pulldown on the RCM5700/5750/6700/6750
8	ACT (RCM5700) +2.5V (RCM6700)			Requires external pullup on the RCM5700/5750
9	PE0	Input/Output	I/O Strobe I0 A20 Timer C0 TCLKF INT0 QRD1B	
10	PE1	Input/Output	I/O Strobe I1 A21 Timer C1 RXD/RCLKF INT1 QRD1A Input Capture	
11	PE2	Input/Output	I/O Strobe I2 A22 Timer C2 TXF DREQ0 QRD2B	
12	PE3	Input/Output	I/O Strobe I3 A23 Timer C3 RXC/RXF/SCLKD DREQ1 QRD2A Input Capture	

Table 4-3. RCM5700/RCM6700 Pinout Configurations

Pin	Pin Name	Default Use	Alternate Use	Notes
13	PE5	Input/Output	I/O Strobe I5 INT1 PWM1 RXB/RCLKE Input Capture	
14	PE6	Input/Output	I/O Strobe I6 PWM2 TXE DREQ0	Serial Port E
15	PE7	Input/Output	I/O Strobe I7 PWM3 RXA/RXE/SCLKC DREQ1 Input Capture	
16	/RESET_IN	Input		Input to Reset Generator
17	PD0	Input/Output	I/O Strobe I0 Timer C0 D8 INT0 SCLKD/TCLKF QRD1B	
18	PD1	Input/Output	IA6 I/O Strobe I1 Timer C1 D9 INT1 RXD/RCLKF QRD1A Input Capture	

Table 4-3. RCM5700/RCM6700 Pinout Configurations

Pin	Pin Name	Default Use	Alternate Use	Notes
19	PD2	Input/Output	I/O Strobe I2 Timer C2 D10 DREQ0 TXF/SCLKC QRD2B	Serial Port F
20	PD3	Input/Output	IA7 I/O Strobe I3 Timer C3 D11 DREQ1 RXC/RXF QRD2A Input Capture	
21	PC0	Input/Output	TXD I/O Strobe I0 Timer C0 TCLKF	Serial Port D
22	PC1	Input/Output	RXD/TXD I/O Strobe I1 Timer C1 RCLKF Input Capture	
23	PC2	Input/Output	TXC/TXF I/O Strobe I2 Timer C2	Serial Port C
24	PC3	Input/Output	RXC/TXC/RXF I/O Strobe I3 Timer C3	
25	PC4	Input/Output	TXB I/O Strobe I4 PWM0	Serial Port B (The RCM5750/RCM5760 and RCM6700 family serial flash uses Alternate Serial Port B (PD4/PD5).
26	PC5	Input/Output	RXB/TXB I/O Strobe I5 PWM1	

Table 4-3. RCM5700/RCM6700 Pinout Configurations

Pin	Pin Name	Default Use	Alternate Use	Notes
27	PB0	Input/Output	SCLKB External I/O Address IA6	SCLKB (used by serial flash on RCM5750/RCM5760 and RCM6700 family.) Due to serial boot flash requirements, the RCM6700 family's edge connector pin 27 (PB0 / SCLKB) must not be used for general purpose I/O. Although reserved for SCLKB only, this usage may be shared with an off-minicore SPI device.
28	/RESET	Reset output		Output from the Reset Generator
29	PB2	Input/Output	/SWR External I/O Address IA0	
30	PB3	Input/Output	/SRD External I/O Address IA1	
31	PB4	Input/Output	SA0 External I/O Address IA2	
32	PB5	Input/Output	SA1 External I/O Address IA3	
33	PB6	Input/Output	/SCS External I/O Address IA4	
34	PB7	Input/Output	/SLAVATN External I/O Address IA5	
35–42	PA[0:7]	Input/Output	Slave port data bus (SD0–SD7) External I/O data bus (ID0–ID7)	
43	/IORD	Output		External I/O read strobe
44	VBAT_EXT	Battery input		
45	/IOWR	Output		External I/O write strobe
46	PB1	Input/Output	SCLKA External I/O Address IA7	Programming port SCLKA
47	STATUS	Output		Programming port

Table 4-3. RCM5700/RCM6700 Pinout Configurations

Pin	Pin Name	Default Use	Alternate Use	Notes
48	PC6	Input/Output	TXA/TXE I/O Strobe I6 PWM2	Programming port
49	SMODE	Input		
50	PC7	Input/Output	RXA/TXA/RXE I/O Strobe I7 PWM3 SCLKC Input Capture	
51	GND			
52	+3.3 V			

4.1.1 Memory I/O Interface

The Rabbit 5000 address lines (A0–A19) and data lines (D0–D7) are routed to the onboard flash memory chip. I/O write (/IOWR) and I/O read (/IORD) are available for interfacing to external devices.

Parallel Port A can also be used as an external I/O data bus to isolate external I/O from the main data bus. Parallel Port B pins PB2–PB7 can also be used as an external address bus.

When using the external I/O bus for either Ethernet or the LCD/keypad module on the Prototyping Board or for any other reason, you must add the following line at the beginning of your program.

```
#define PORTA_AUX_IO    // required to enable external I/O bus
```

Selected pins on Parallel Ports D and E as specified in Table 4-3 may be used for input capture, quadrature decoder, DMA, and pulse-width modulator purposes.

4.1.2 Other Inputs and Outputs

The status, /RESET_IN, and SMODE I/O are normally associated with the programming port. Since the status pin is not used by the system once a program has been downloaded and is running, the status pin can then be used as a general-purpose CMOS output. The programming port is described in more detail in Section 4.2.3.

/RESET_IN is an external input used to reset the Rabbit 5000/6000 microprocessor and the MiniCore memory. /RESET is an output from the reset circuitry that can be used to reset other peripheral devices.

The two SMODE pins, SMODE0 and SMODE1, are tied together, and may be used as a special input when the MiniCore is operating in the Run Mode. The logic state of these two pins determines the startup procedure after a reset.

4.2 Serial Communication

The RCM5700/RCM6700 board does not have any serial level converters directly on the board. However, an Ethernet or other serial interface may be incorporated on the board the MiniCore is mounted on. For example, the Serial Communication accessory board in the Deluxe Development Kit has an RS-232 transceiver, and the Interface Board has Ethernet and USB connections.

4.2.1 Serial Ports

There are six serial ports designated as Serial Ports A, B, C, D, E, and F. All six serial ports can operate in an asynchronous mode up to the baud rate of the system clock divided by 8. An asynchronous port can handle 7 or 8 data bits. A 9th bit address scheme, where an additional bit is sent to mark the first byte of a message, is also supported.

Serial Port A is normally used as a programming port, but may be used either as an asynchronous or as a clocked serial port once application development has been completed and the MiniCore is operating in the Run Mode.

Serial Ports B, C, and D can also be operated in the clocked serial mode. In this mode, a clock line synchronously clocks the data in or out. Either of the two communicating devices can supply the clock.

Note that Alternate Serial Port B, which uses PD4 and PD5, is used together with the clock on PB0 for the serial flash on the RCM5750/RCM5760 and RCM6700 family. If you wish to use the regular Serial Port B I/O pins (PC4 and PC5) as serial I/O pins with the RCM5750/RCM5760 or RCM6700 family, your application must manage its sharing of Serial Port B so as to avoid any conflicts with using the onboard serial flash.

Serial Ports E and F can also be configured as SDLC/HDLC serial ports. The IrDA protocol is also supported in SDLC format by these two ports. Serial Ports E and F must be configured before they can be used. The following macros show one way to do this.

```
#define SERE_TXPORT PEDR
#define SERE_RXPORT PEDR

#define SERF_TXPORT PFDR
#define SERF_RXPORT PFDR
```

Table 4-4 summarizes the possible parallel port pins for the serial ports and their clocks.

Table 4-4. Rabbit 5000 and 6000 Serial Port and Clock Pins

Serial Port A	TXA	PC6, PC7	Serial Port E	TXE	PE6, PC6
	RXA	PC7,PE7		RXE	PE7, PC7
	SCLKA	PB1		RCLKE	PE5, PC5
Serial Port B	TXB	PC4, PC5		TCLKE	PE4, PC4
	RXB	PC5, PE5	Serial Port F	TXF	PD2, PE2, PC2
	SCLKB	PB0		RXF	PD3, PE3, PC3
Serial Port C	TXC	PC2, PC3		RCLKF	PD1, PE1, PC1
	RXC	PC3, PD3, PE3		TCLKF	PD0, PE0, PC0
	SCLKC	PD2, PE2, PE7, PC7	RCLKE/TCLKE and RCLKF/TCLKF must be selected to be on the same parallel port as RXE/TXE and RXF/TXF respectively.		
Serial Port D	TXD	PC0, PC1			
	RXD	PC1, PD1, PE1			
	SCLKD	PD0, PD3, PE0, PE3, PC3			

4.2.2 Ethernet PHY

All RCM5700/RCM6700 models have an Ethernet PHY, which can either be accessed through the Interface Board or directly on the RCM5710/5760/6710/6760. The PHY connections or integrated 10/100Base-T connections on MiniCores with an on-board RJ-45 jack are via 0 Ω jumpers on headers JP2–JP5 (see Table A-6 in Appendix A.3). Rabbit's Technical Note TN266, *PCB Layout for the Ethernet PHY Interface*, provides further details about designing your own PHY interface.

The pinout for the RJ-45 Ethernet jack (J1) is shown in Figure 4-11. Note that some Ethernet connectors are numbered in reverse to the order used here.

The RJ-45 Ethernet jack incorporates green and yellow LEDs, one to indicate Ethernet link (**LINK**), and one to indicate Ethernet activity (**ACT**).

The RJ-45 connector is shielded to minimize EMI effects to/from the Ethernet signals.

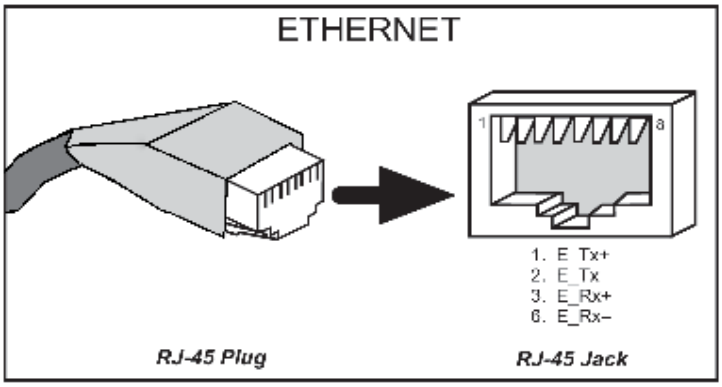


Figure 4-11. RJ-45 Ethernet Port Pinout

4.2.3 Programming Port

The RCM5700/RCM6700 programming port is accessed via the USB connector (J5) on the Interface Board. The programming port uses the Rabbit CPU's Serial Port A for communication. Dynamic C uses the programming port to download and debug programs.

The programming port is also used to cold-boot the Rabbit processor on the RCM5700/RCM6700 after a reset.

4.3 Programming Modes

The USB cable is used to connect the programming port of the RCM5700/RCM6700 to a PC USB port via the Interface Board.

Whenever the MiniCore is reset, the operating mode is determined by the state of the SMODE pins. The MiniCore is automatically in Program Mode when the SMODE pins, which are tied together, are pulled up to +3.3 V. This happens when the MiniCore is installed on the Interface Board, and pins 1–2 on header JP1 on the Interface Board are jumpered. When the SMODE pins are pulled low by removing the jumpers from pins 1–2 on header JP1 on the Interface Board, the Rabbit 5000/6000 will operate in the Run Mode once the MiniCore is reset.

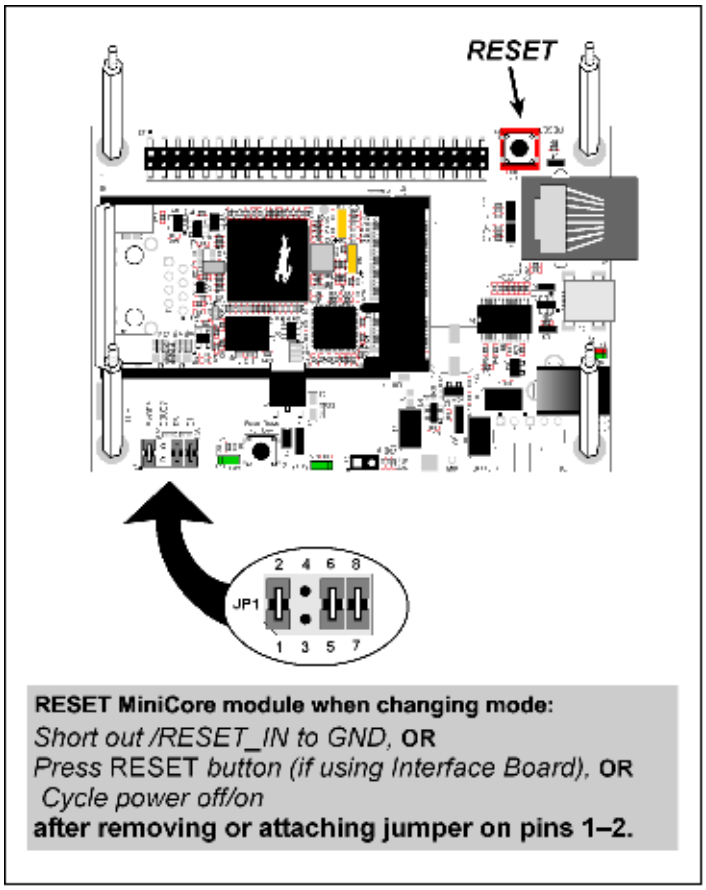


Figure 4-12. Switching Between Program Mode and Run Mode

When you use the USB cable supplied with the Standard Development Kit while the MiniCore is operating in the Run Mode, the MiniCore will reboot once approximately 2 seconds after its initial startup. To avoid this reboot, you will have to use the AC adapter supplied with the Deluxe Development Kit, your own 5 V power supply, or you will need a special USB cable that has a separate "power only" connection option to supply power via the Interface Board to operate the MiniCore in the Run Mode.

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

Refer to the *Rabbit 5000 Microprocessor User's Manual* or *Rabbit 6000 Microprocessor User's Manual* for more information on the programming port.

4.3.1 Standalone Operation of the RCM5700/RCM6700

The RCM5700/RCM6700 must be programmed via the Interface Board or via a similar arrangement on a customer-supplied board. Once the MiniCore has been programmed successfully, reset the MiniCore. The MiniCore may be reset by cycling power off/on or by pressing the **RESET** button on the Interface Board. The jumper across pins 1–2 on header JP1 on the Interface Board must be removed in order for the MiniCore to operate in the Run Mode after it is reset. The MiniCore module may now be removed from the Interface Board for end-use installation.

CAUTION: Power to the Interface Board or other boards should be disconnected when removing or installing your RCM5700/RCM6700 module to protect against inadvertent shorts across the pins or damage to the MiniCore if the pins are not plugged in correctly. Do not reapply power until you have verified that the MiniCore module is plugged in correctly.

4.4 Other Hardware

4.4.1 Clocks

The RCM5700 takes advantage of the Rabbit 5000 microprocessor's internal clock doubler. A built-in clock doubler allows half-frequency crystals to be used to reduce radiated emissions. The 50.0 MHz frequency specified for the RCM5700 model is generated using a 25.0 MHz crystal.

The RCM5700 requires that the clock doubler remain enabled for proper Ethernet operation.

The RCM6700 requires that the clock doubler remain disabled, but takes advantage of the Rabbit 6000 microprocessor's internal PLL, providing clock speeds up to 162.5 MHz.

4.4.2 Spectrum Spreader

The Rabbit 5000/6000 features a spectrum spreader, which helps to mitigate EMI problems. The spectrum spreader is on by default on the RCM5700, but is off on the RCM6700 because of its 115 MHz maximum clock limitation. It may also be turned off or set to a stronger setting. The means for doing so is through a simple configuration macro as shown below.

1. Select the "Defines" tab from the Dynamic C **Options > Project Options** menu.
2. Normal spreading is the default, and usually no entry is needed. If you need to specify normal spreading, add the line: **`ENABLE_SPREADER=1`**

   For strong spreading, add the line: **`ENABLE_SPREADER=2`**

   To disable the spectrum spreader, add the line: **`ENABLE_SPREADER=0`**

   NOTE: The strong spectrum-spreading setting is not recommended since it may limit the maximum clock speed or the maximum baud rate. It is unlikely that the strong setting will be used in a real application.

3. Click **OK** to save the macro. The spectrum spreader will be set according to the macro value whenever a program is compiled using this project file.

   NOTE: Refer to the *Rabbit 5000 Microprocessor User's Manual* or the *Rabbit 6000 Microprocessor User's Manual* for more information on the spectrum-spreading setting and the maximum clock speed.

4.5 Memory

4.5.1 RAM

RCM5700 boards have 128KB of onchip SRAM on the Rabbit 5000 microprocessor. The RCM5750/RCM5760 models also have 1 MB of external SRAM.

RCM6700 boards have 1MB of onchip RAM and 32KB of onchip battery-backable SRAM on the Rabbit 6000 microprocessor. The RCM6750/6760 models also have 1MB of external SRAM.

4.5.2 Program Flash Memory

All RCM5700 models have 1MB of parallel flash memory installed at U3. RCM6700/RCM6710 have a 1MB serial flash and RCM6750/RCM6760 have a 4MB serial flash installed.

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

A "user block" area is defined to store persistent data. The function calls `writeUserBlock()` and `readUserBlock()` are provided for this. Refer to the ***Dynamic C Function Reference Manual*** for additional information.

4.5.3 Mass Storage Serial Flash

The RCM5750/RCM5760 have 2MB of serial flash available to store data and web pages. Sample programs in the `SAMPLES\RCM5700\Serial_Flash` folder illustrate the use of serial flash memory.

The RCM6700/RCM6710 have 1MB of serial flash memory, and RCM6750/RCM6760 have 4MB of serial flash memory. The serial flash is used to store the program code, but can also be used to store data and web pages. Sample programs in the `SAMPLES\RCM6700\Serial_Flash` folder illustrate the use of the serial flash memory.

4.5.4 Encryption RAM Memory

The tamper detection feature of the Rabbit 5000/6000 microprocessor can be used to detect any attempt to enter the bootstrap mode. When such an attempt is detected, the VBAT RAM memory in the Rabbit 5000/6000 chip is erased.

5. SOFTWARE REFERENCE

Dynamic C is an integrated development system for writing embedded software. It runs on a Windows-based PC and is designed for use with single-board computers and other devices based on the Rabbit microprocessor. Chapter 5 describes the libraries and function calls related to the RCM5700/RCM6700.

5.1 More About Dynamic C

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

You should do your software development in the flash memory on the RCM5700/RCM6700. The flash memory and options are selected with the **Options > Program Options > Compiler** menu.

> **NOTE:** Do not depend on the flash memory sector size or type in your program logic. The RCM5700/RCM6700 and Dynamic C were designed to accommodate flash devices with various sector sizes in response to the volatility of the flash-memory market.

Developing software with Dynamic C is simple. Users can write, compile, and test C and assembly code without leaving the Dynamic C development environment. Debugging occurs while the application runs on the target. Alternatively, users can compile a program to an image file for later loading. Dynamic C runs on PCs under Windows NT and later—see Rabbit's Technical Note TN257, *Running Dynamic C® With Windows Vista®*, for additional information if you are using Dynamic C under Windows Vista. Programs can be downloaded at baud rates of up to 460,800 bps after the program compiles.

Dynamic C has a number of standard features.

- Full-feature source and/or assembly-level debugger, no in-circuit emulator required.
- Royalty-free TCP/IP stack with source code and most common protocols.
- Hundreds of functions in source-code libraries and sample programs:
  - Exceptionally fast support for floating-point arithmetic and transcendental functions.
  - RS-232 and RS-485 serial communication.
  - Analog and digital I/O drivers.
  - I^2C, SPI, GPS, file system.
  - LCD display and keypad drivers.
- Powerful language extensions for cooperative or preemptive multitasking
- Loader utility program to load binary images into Rabbit targets in the absence of Dynamic C.
- Provision for customers to create their own source code libraries and augment on-line help by creating "function description" block comments using a special format for library functions.
- Standard debugging features:
  - Breakpoints—Set breakpoints that can disable interrupts.
  - Single-stepping—Step into or over functions at a source or machine code level, µC/OS-II aware.
  - Code disassembly—The disassembly window displays addresses, opcodes, mnemonics, and machine cycle times. Switch between debugging at machine-code level and source-code level by simply opening or closing the disassembly window.
  - Watch expressions—Watch expressions are compiled when defined, so complex expressions including function calls may be placed into watch expressions. Watch expressions can be updated with or without stopping program execution.
  - Register window—All processor registers and flags are displayed. The contents of general registers may be modified in the window by the user.
  - Stack window—shows the contents of the top of the stack.
  - Hex memory dump—displays the contents of memory at any address.
  - **STDIO** window—`printf` outputs to this window and keyboard input on the host PC can be detected for debugging purposes. `printf` output may also be sent to a serial port or file.

5.2 Dynamic C Function Calls

5.2.1 Digital I/O

The RCM5700/RCM6700 was designed to interface with other systems, and so there are no drivers written specifically for the Rabbit 5000/6000 I/O. The general Dynamic C read and write functions allow you to customize the parallel I/O to meet your specific needs. For example, use

```
WrPortI(PEDDR, &PEDDRShadow, 0x00);
```

to set all the Port E bits as inputs, or use

```
WrPortI(PEDDR, &PEDDRShadow, 0xFF);
```

to set all the Port E bits as outputs.

When using the external I/O bus on the Rabbit 5000/6000 chip, add the line

```
#define PORTA_AUX_IO    // required to enable external I/O bus
```

to the beginning of any programs using the auxiliary I/O bus.

The sample programs in the Dynamic C `SAMPLES\RCM5700 and SAMPLES\RCM6700` folders provide further examples.

5.2.2 Serial Communication Drivers

Library files included with Dynamic C provide a full range of serial communications support. The `RS232.LIB` library provides a set of circular-buffer-based serial functions. The `PACKET.LIB` library provides packet-based serial functions where packets can be delimited by the 9th bit, by transmission gaps, or with user-defined special characters. Both libraries provide blocking functions, which do not return until they are finished transmitting or receiving, and nonblocking functions, which must be called repeatedly until they are finished, allowing other functions to be performed between calls. For more information, see the *Dynamic C Function Reference Manual* and Rabbit's Technical Note TN213, *Rabbit Serial Port Software*, both included with the online documentation.

5.2.3 User Block

None of the user block memory area is reserved. This leaves the address range 0–1FFA in the user block available for your application.

These address ranges may change in the future in response to the volatility in the flash memory market, in particular sector size. The sample program `USERBLOCK_INFO.C` in the Dynamic C `SAMPLES\USERBLOCK` folder can be used to determine the version of the ID block, the size of the ID and user blocks, whether or not the ID/user blocks are mirrored, the total amount of flash memory used by the ID and user blocks, and the area of the user block available for your application.

The `USERBLOCK_CLEAR.C` sample program shows you how to clear and write the contents of the user block that you are using in your application (the calibration constants in the reserved area and the ID block are protected).

5.2.4 RCM5700/RCM6700 Cloning

The RCM5700/RCM6700 does not have a programming header, and is programmed through the USB connection on the Interface Board. Rabbit's Cloning Board does not support cloning through a USB connection. If there is a need to copy programs, the Rabbit Field Utility can be used to download compiled Dynamic C .bin files.

5.2.5 TCP/IP Drivers

The TCP/IP drivers are located in the `LIB\Rabbit4000\TCPIP` folder.

Complete information on these libraries and the TCP/IP functions is provided in the *Dynamic C TCP/IP User's Manual.*

5.2.6 Serial Flash Drivers (not supported on RCM5700/RCM5710)

The Dynamic C `LIB\SerialFlash\SFLASH.LIB` library provides the function calls needed to use the serial flash. The FAT file system function calls from in the Dynamic C `LIB\FileSystem\FAT_CONFIG.LIB` library are not supported.

5.3 Upgrading Dynamic C

Dynamic C patches that focus on bug fixes are available from time to time. For the latest patches, workarounds, and bug fixes, check the Web site www.digi.com/support/.

5.3.1 Add-On Modules

Starting with Dynamic C version 10.40, Dynamic C includes the popular µC/OS-II real-time operating system, point-to-point protocol (PPP), FAT file system, RabbitWeb, and other select libraries. Starting with Dynamic C version 10.56, Dynamic C includes the Rabbit Embedded Security Pack featuring the Secure Sockets Layer (SSL) and a specific Advanced Encryption Standard (AES) library.

In addition to the Web-based technical support included at no extra charge, a one-year telephone-based technical support subscription is also available for purchase.

For further information and complete documentation, visit our Web site at www.digi.com.

APPENDIX A. RCM5700/RCM6700 SPECIFICATIONS

Appendix A provides the specifications for the RCM5700 and RCM6700.

A.1 Electrical and Mechanical Characteristics

Figures A-1(a) and A-1(b) show the mechanical dimensions for the RCM5700/RCM6700 and RCM5760/RCM6760. The related dimensions for the RCM5710/6710 and RCM5750/RCM6750 are listed in Table A-1. (All measurements are in inches followed by millimeters enclosed in parentheses.)

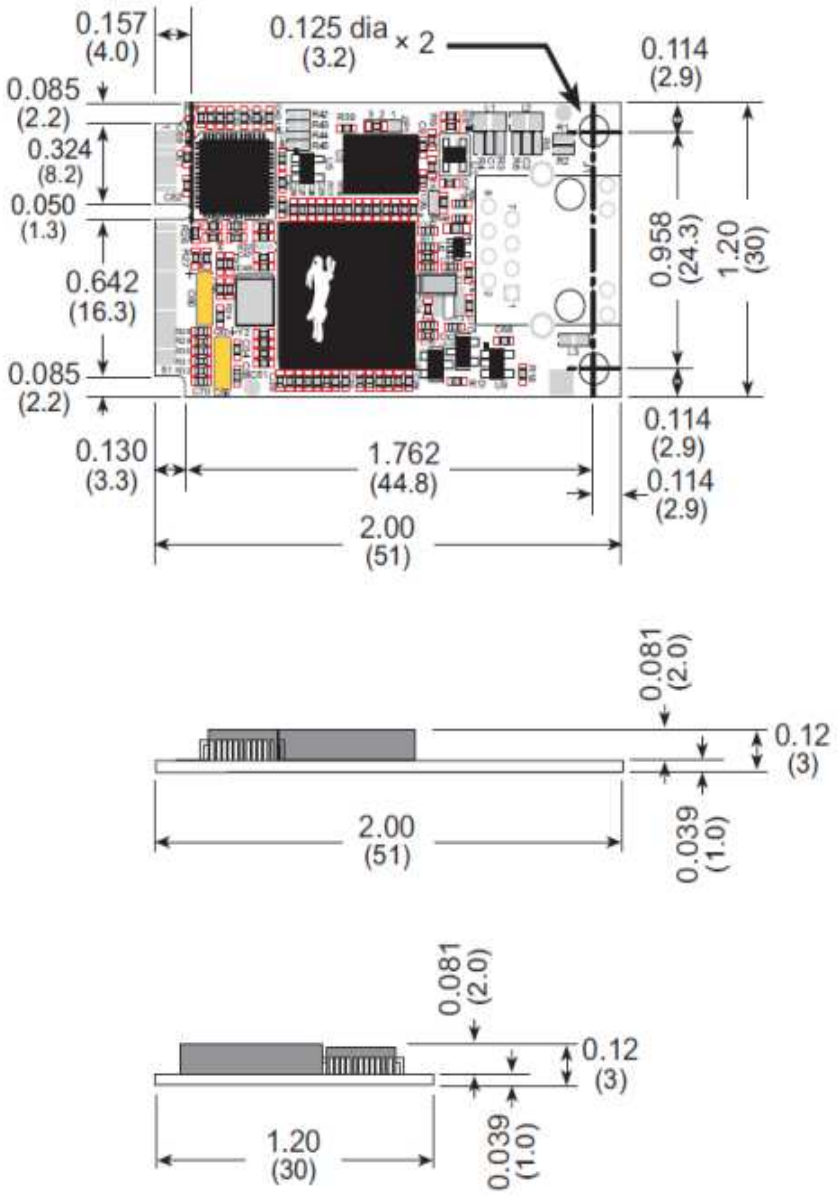


Figure A-1(a). RCM5700/RCM6700 Dimensions

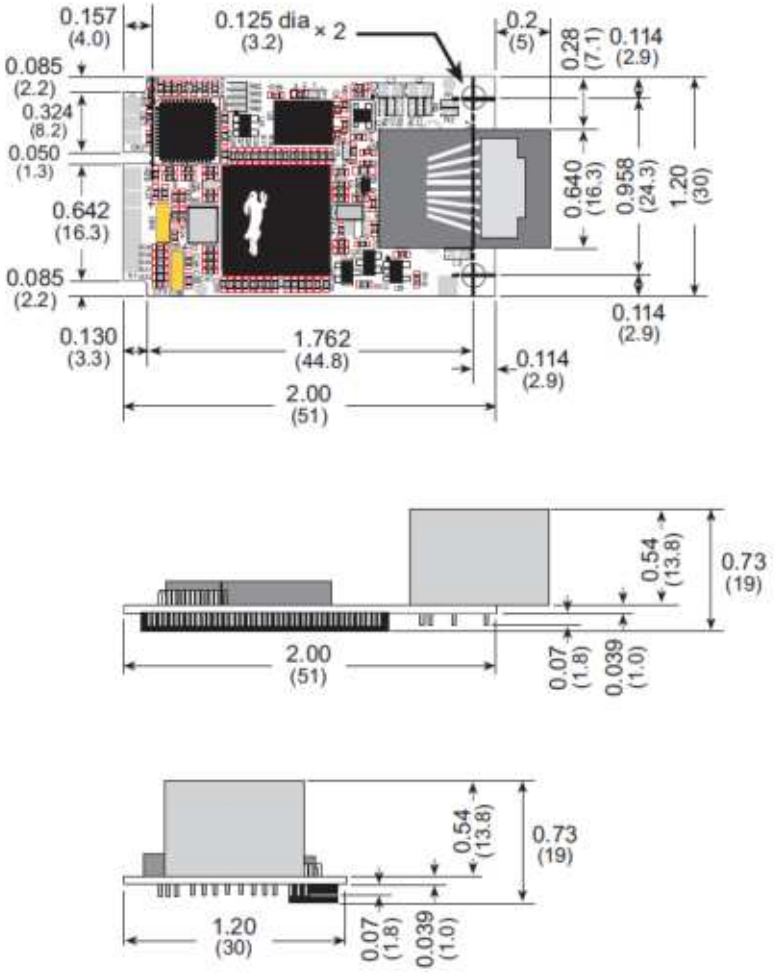


Figure A-1(b). RCM5760/RCM6760 Dimensions

It is recommended that you allow for an “exclusion zone” of 0.08" (2 mm) around the RCM5700/RCM6700 top and bottom and 0.04" (1 mm) around the three non-connector edges when the RCM5700/RCM6700 is incorporated into an assembly that includes other printed circuit boards. This “exclusion zone” that you keep free of other components and boards will allow for sufficient air flow, and will help to minimize any electrical or electromagnetic interference between adjacent boards. Figure A-2 shows this “exclusion zone” for the RCM5700/RCM6700 model. In the event that you are using a device with additional components installed (such as the RCM5710, RCM6710, RCM5750, or RCM6750) the exclusion zone in Figure A-2 should be extended 0.08" (2 mm) beyond the edge of the added components.

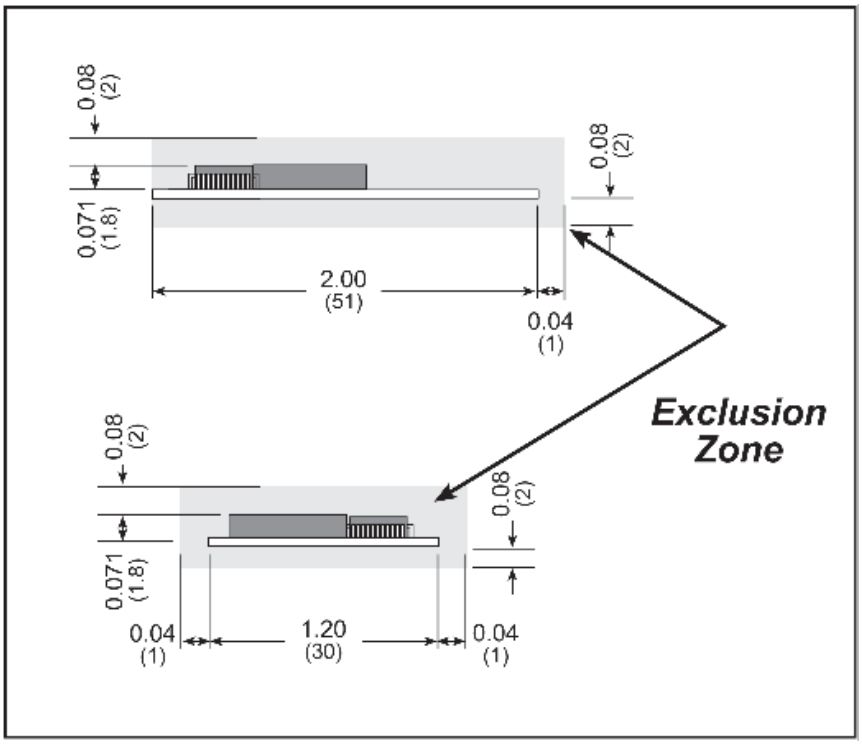


Figure A-2. RCM5700/RCM6700 “Exclusion Zone”

Table A-1 lists the electrical, mechanical, and environmental specifications for the RCM5700.

Table A-1. RCM5700 Specifications

Parameter	RCM5700	RCM5710	RCM5750	RCM5760
Microprocessor	Rabbit® 5000 at 50.0 MHz			
EMI Reduction	Spectrum spreader for reduced EMI (radiated emissions)			
Ethernet Port	10/100Base-T PHY only	10/100Base-T, RJ-45, 2 LEDs	10/100Base-T PHY only	10/100Base-T, RJ-45, 2 LEDs
Flash Memory (program)	1MB			
Flash Memory (mass data storage)	—		2MB (serial flash)	
External SRAM	—		512KB (8-bit)	
SRAM	128KB (Rabbit® 5000 onchip)			
Backup Battery	Connection for user-supplied backup battery (to support RTC)		None	
General-Purpose I/O	up to 35 parallel digital I/0 lines configurable with four layers of alternate functions			
Additional Inputs	Reset in			
Additional Outputs	Status, reset out			
External I/O Bus	Can be configured for 8 data lines and 8 address lines (shared with parallel I/O lines), plus I/O read/write			
Serial Ports	6 high-speed, CMOS-compatible ports: • all 6 configurable as asynchronous (with IrDA), 4 as clocked serial (SPI), and 2 as SDLC/HDLC • 1 clocked serial port shared with programming port			
Serial Rate	Maximum asynchronous baud rate = CLK/8			
Slave Interface	Slave port allows the RCM5700 to be used as an intelligent peripheral device slaved to a master processor			
Real-Time Clock	Yes			
Timers	Ten 8-bit timers (6 cascadable from the first), one 10-bit timer with 2 match registers, and one 16-bit timer with 4 outputs and 8 set/reset registers			
Watchdog/Supervisor	Yes			
Pulse-Width Modulators	4 channels synchronized PWM with 10-bit counter or 4 channels variable-phase or synchronized PWM with 16-bit counter			

Table A-1. RCM5700 Specifications

Parameter	RCM5700	RCM5710	RCM5750	RCM5760
Input Capture	2-channel input capture can be used to time input signals from various port pins			
Quadrature Decoder	2-channel quadrature decoder accepts inputs from external incremental encoder modules			
Power	3.15 V DC (min.) – 3.45 V DC (max.) 70 mA @ 3.3 V (typical — without Ethernet) 200 mA @ 3.3 V (typical — with Ethernet)			
Operating Temperature	–40°C to +85°C			
Humidity	5% to 95%, noncondensing			
Connectors	Edge connectors for interface with 52-pin mini PCI Express socket			
Board Size	1.20" × 2.00" × 0.12" (30 mm × 51 mm × 3 mm)	1.20" × 2.00" × 0.70" (30 mm × 51 mm × 18 mm)	1.20" × 2.00" × 0.27" (30 mm × 51 mm × 7 mm)	1.20" × 2.00" × 0.73" (30 mm × 51 mm × 19 mm)

Table A-2 lists the electrical, mechanical, and environmental specifications for the RCM6700.

Table A-2. RCM6700 Specifications

Parameter	RCM6700	RCM6710	RCM6750	RCM6760
Microprocessor	Rabbit® 6000 at 187.5 MHz			
EMI Reduction	Spectrum spreader for reduced EMI (radiated emissions) (when running below 115 MHz)			
Ethernet Port	10/100Base-T PHY only	10/100Base-T, RJ-45, 2 LEDs	10/100Base-T PHY only	10/100Base-T, RJ-45, 2 LEDs
Flash Memory (mass data storage)	1 MB (serial flash)		4 MB (serial flash)	
External SRAM	—		1MB (16-bit)	
RAM	1MB (Rabbit® 6000 onchip)			
SRAM	32KB (Rabbit 6000 onchip, battery-backable)			
Backup Battery	Connection for user-supplied backup battery (to support RTC)			
General-Purpose I/O	up to 35 parallel digital I/0 lines configurable with four layers of alternate functions			
Additional Inputs	Reset in			
Additional Outputs	Status, reset out			
External I/O Bus	Can be configured for 8 data lines and 8 address lines (shared with parallel I/O lines), plus I/O read/write			
Serial Ports	6 high-speed, CMOS-compatible ports: • all 6 configurable as asynchronous (with IrDA), 4 as clocked serial (SPI), and 2 as SDLC/HDLC • 1 clocked serial port shared with programming port			
Serial Rate	Maximum asynchronous baud rate = CLK/8			
Slave Interface	Slave port allows the RCM6700 to be used as an intelligent peripheral device slaved to a master processor			
Real-Time Clock	Yes			
Timers	Ten 8-bit timers (6 cascadable from the first), one 10-bit timer with 2 match registers, and one 16-bit timer with 4 outputs and 8 set/reset registers			
Watchdog/Supervisor	Yes			
Pulse-Width Modulators	4 channels synchronized PWM with 10-bit counter or 4 channels variable-phase or synchronized PWM with 16-bit counter			
Input Capture	2-channel input capture can be used to time input signals from various port pins			

Table A-2. RCM6700 Specifications

Parameter	RCM6700	RCM6710	RCM6750	RCM6760
Quadrature Decoder	2-channel quadrature decoder accepts inputs from external incremental encoder modules			
Power*	3.15 V DC (min.) – 3.45 V DC (max.)			
	210 mA @ 3.3V (typical -- with Ethernet)	250 mA @ 3.3V (typical -- with Ethernet) 320 mA @ 3.3V (MAX -- with Ethernet)	220 mA @ 3.3V (typical -- with Ethernet)	260 mA @ 3.3V (typical -- with Ethernet) 330 mA @ 3.3V (MAX -- with Ethernet)
	120 mA @ 3.3V (typical -- without Ethernet	130 mA @ 3.3V (typical -- without Ethernet	130 mA @ 3.3V (typical -- without Ethernet)	140 mA @ 3.3V (typical -- without Ethernet
Operating Temperature	–40°C to +85°C			
Humidity	5% to 95%, noncondensing			
Connectors	Edge connectors for interface with 52-pin mini PCI Express socket			
Board Size	1.20" × 2.00" × 0.12" (30 mm × 51 mm × 3 mm)	1.20" × 2.00" × 0.70" (30 mm × 51 mm × 18 mm)	1.20" × 2.00" × 0.27" (30 mm × 51 mm × 7 mm)	1.20" × 2.00" × 0.73" (30 mm × 51 mm × 19 mm)

* Test data taken with TEST_NET_IF = 1 and flood ping (for "w/Ethernet" values) and TEST_NET_IF = 0 (for "wo/Ethernet" values)

A.1.1 mini PCI Express Connector Design Recommendations

The MiniCore is mounted on the Interface Board via a mini PCI Express connector and a corresponding locking latch connector. These are offered by manufacturers as a matched set, although in some cases different manufacturer's parts may be interchangeable. Table A-3 lists the recommended parts for the mini PCI Express connector and the locking latch connector used for the Interface Board. (All measurements are in inches followed by millimeters enclosed in parentheses.)

Table A-3. Interface Board Connector-Related Parts

Connector Part	Manufacturer Part Number	Rabbit Store Part Number
mini PCI Express Connector	Pinrex 984-63-052202	498-0090
Latch Connector	FOXCONN AS0B226S68K-7F	498-0091

Figure A-3 shows a footprint for the SMT connectors in Table A-3.

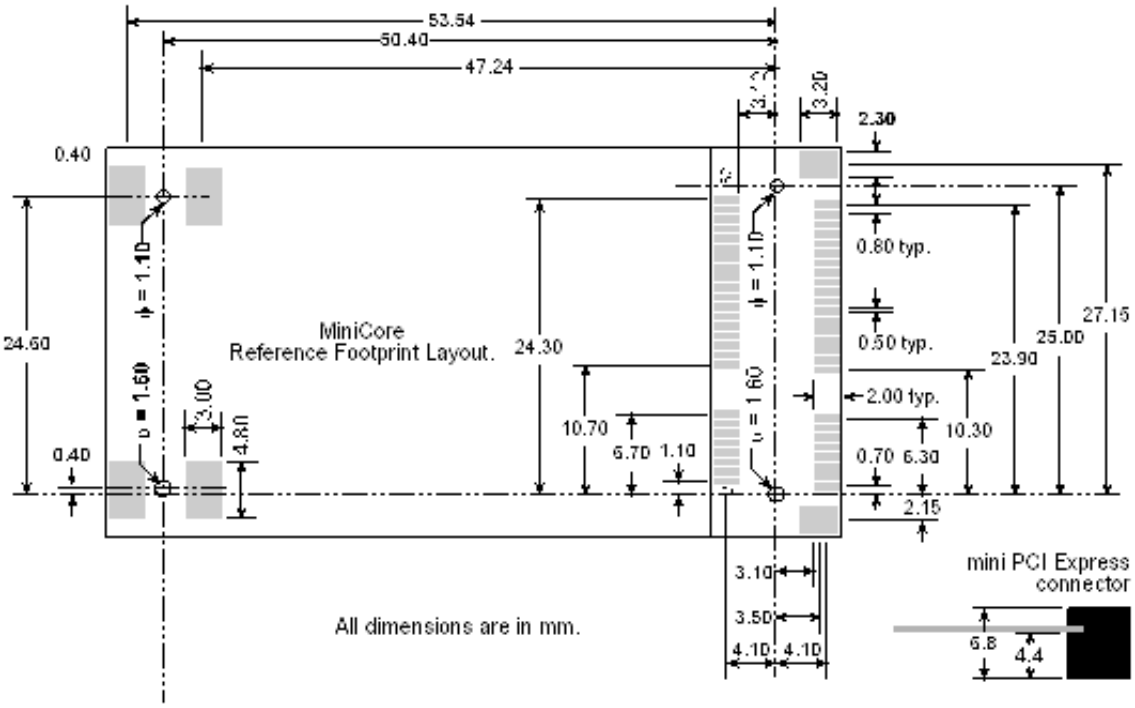


Figure A-3. PCB SMT Footprint for Pinrex and FOXCONN Connectors

Other manufacturers such as Molex offer similar connectors and latches, but these can have different mechanical structures and PCB footprints to what we use on the Interface Board. Table A-4 lists a pair of matched Molex parts that might be used.

Table A-4. Molex Connector-Related Parts

Connector Part	Molex Part Number
mini PCI Express Connector	67910-0002
Latch Connector	48099-4000

Follow the PCB design and layout recommendations and considerations provided by the manufacturer for the connector and latch that you select.

The MiniCore may also be mounted with standoffs instead of a latch connector — this is the type of mounting recommended for the MiniCore to maximize both the vibration resistance and the ground connections via the standoffs between the MiniCore module and the board it is mounted on. The height of the standoffs will depend on the mini PCI Express connector selected — Table A-5 provides some examples. Figure A-4 shows a footprint for this mounting option based on the mini PCI Express connector in Table A-3.

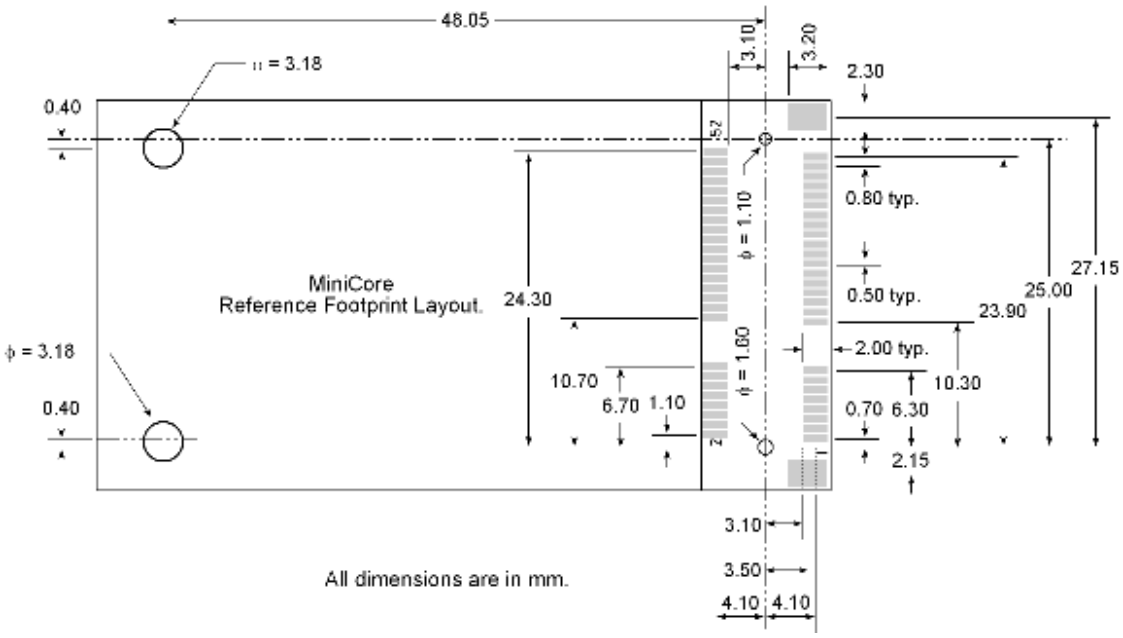


Figure A-4. PCB Standoff Footprint with mini PCI Express Connectors

Table A-5. Standoff Heights Based on mini PCI Express Connector

mini PCI Express Connector Height	Standoff Height	Remarks
6.8 mm	4.4 mm	Used with Interface Board
9.2 mm	6.2 mm	Used with Accessory Kit

The SMT connectors are ideal in a development environment, where the latch connector facilitates swapping MiniCore modules as development progresses. The absence of holes also maximizes trace routing flexibility on the printed circuit board. The standoff option offers better vibration resistance and grounding once you are ready to deploy your application based on the RCM5700/RCM6700.

The Rabbit store sells an accessory kit (Part No. 101-1306) with the standoffs, screws, and mini PCI Express connector needed to mount an MiniCore module using the footprint shown in Figure A-4. The heights of the mini PCI Express connector and the associated standoffs in the accessory kit are shown in millimeters at right.

A.2 Rabbit 5000 and 6000 Microprocessor Characteristics

The *Rabbit 5000 Microprocessor User's Manual* and *Rabbit 6000 Microprocessor User's Manual*, which are included with the online documentation, provide complete specifications and timing diagrams for the Rabbit 5000 and 6000 microprocessors.

Rabbit's Technical Note TN227, *Interfacing External I/O with Rabbit Microprocessor Designs*, which is also included with the online documentation, contains suggestions for interfacing I/O devices to the Rabbit 5000/6000 microprocessors.

A.3 Jumper Configurations

Figure A-5 shows the header locations used to configure the various RCM5700/RCM6700 options via jumpers. Note that some early versions of the RCM5700 model sold in 2008 and 2009 do not have jumper locations JP2–JP5 — this does not affect the functionality of the RCM5700 model.

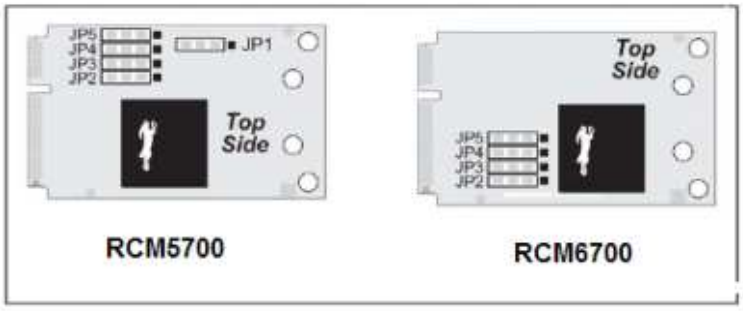


Figure A-5. Location of RCM5700/RCM6700 Configurable Position

Table A-6 lists the configuration options for the RCM5700, and Table A-7 lists the configuration options for the RCM6700.

Table A-6. RCM5700 Jumper Configurations

Header	Description	Pins Connected		Factory Default
JP1	Flash Memory Size	1–2	More than 1MB	
		2–3	Up to 1MB	×
JP2	Tx+	1–2	Tx+ to RJ-45 jack (J1)	RCM5710/ RCM5760
		2–3	Tx+ to mini PCI Express pins	RCM5700/ RCM5750
JP3	Tx–	1–2	Tx– to RJ-45 jack (J1)	RCM5710/ RCM5760
		2–3	Tx– to mini PCI Express pins	RCM5700/ RCM5750
JP4	Rx–	1–2	Rx– to RJ-45 jack (J1)	RCM5710/ RCM5760
		2–3	Rx– to mini PCI Express pins	RCM5700/ RCM5750
JP5	Rx+	1–2	Rx+ to RJ-45 jack (J1)	RCM5710/ RCM5760
		2–3	Rx+ to mini PCI Express pins	RCM5700/ RCM5750

NOTE: The jumper connections are made using 0 Ω surface-mounted resistors.

Table A-7. RCM6700 Jumper Configurations

Header	Description	Pin Connected		Factory Default
JP2	Tx+	1-2	Tx+ to RJ-45 jack (J1)	RCM6710/ RCM6760
		2-3	Tx+ to mini PCI Express pins	RCM6700/ RCM6750
JP3	Tx-	1-2	Tx- to RJ-45 jack (J1)	RCM6710/ RCM6760
		2-3	Tx- to mini PCI Express pins	RCM6700/ RCM6750
JP4	Rx-	1-2	Rx- to RJ-45 jack (J1)	RCM6710/ RCM6760
		2-3	Rx- to mini PCI Express pins	RCM6700/ RCM6750
JP5	Rx+	1-2	Rx+ to RJ-45 jack (J1)	RCM6710/ RCM6760
		2-3	Rx+ to mini PCI Express pins	RCM6700/ RCM6750

NOTE: The jumper connections are made using 0 Ω surface-mounted resistors.

APPENDIX B. INTERFACE BOARD

Appendix B describes the features and accessories of the Interface Board, and explains the use of the Interface Board to demonstrate the RCM5700 and RCM6700. The Interface Board has power-supply connections and a USB interface to program either MiniCore module.

B.1 Introduction

The Interface Board included in the Development Kit makes it easy to connect MiniCore module to a power supply and a PC workstation for development. It also provides an Ethernet port.

The Interface Board is shown below in Figure B-1, with its main features identified.

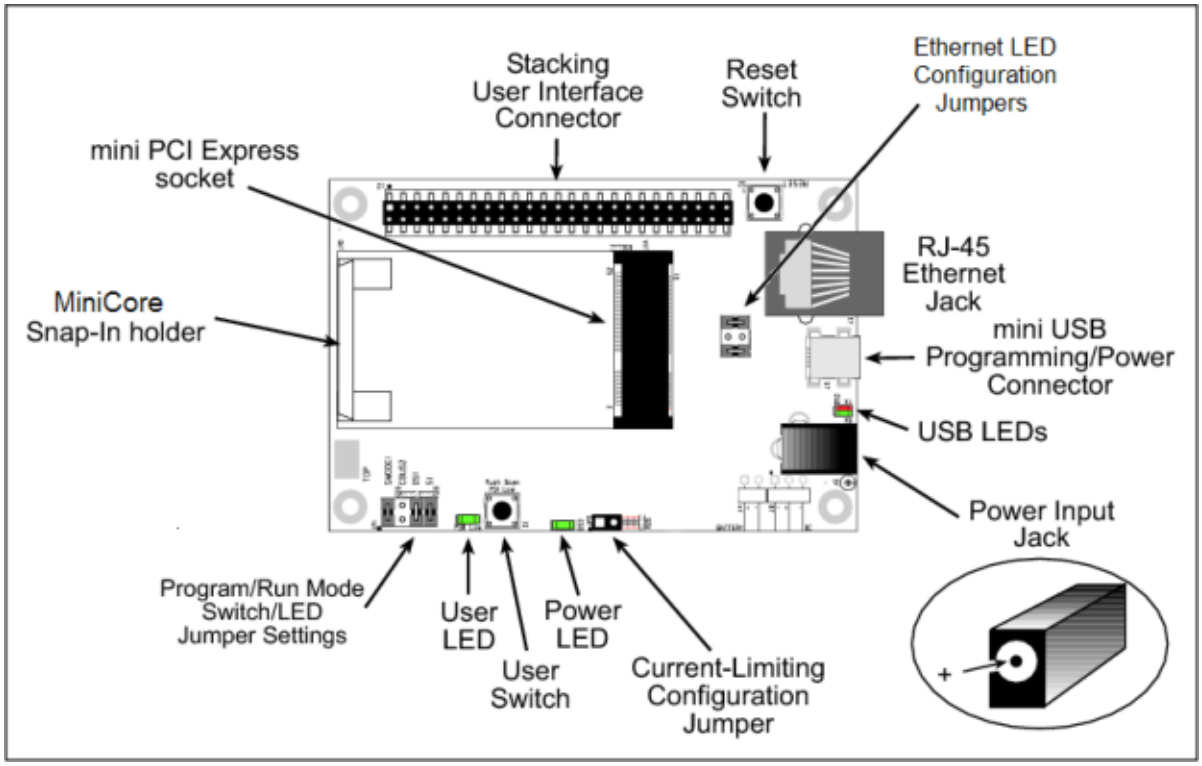


Figure B-1. Interface Board

B.1.1 Interface Board Features

- **Power Connection**—Power is supplied to the Interface Board either from the PC via the USB connection or through a power supply jack, J6. A chip at U4 disconnects the USB power supply from the rest of the Interface Board when power is supplied through jack J6

  Users providing their own power supply should ensure that it delivers +5 V DC at 10 W.

- **Regulated Power Supply**—The raw DC voltage is routed to a 3.3 V linear regulator. The regulator provides stable power to the MiniCore module and other boards connected to it.

- **Power LED**—The power LED lights whenever power is connected to the Interface Board.

- **Reset Switch**—A momentary-contact, normally open switch is connected directly to the MiniCore's **/RESET_IN** pin. Pressing the switch forces a hardware reset of the system.

- **Mini USB Connector**—A USB cable is used to connect the MiniCore via the Interface Board to your PC to enable you to program your MiniCore module.

- **Mini PCI Express socket**—The Interface Board provides a 52-pin mini PCI Express socket to enable you to install your MiniCore module. There is a snap-in holder to hold the opposite end of the MiniCore module firmly in place.

- **I/O Switch and LED**—One momentary-contact, normally open switch is connected to PD1 on the MiniCore module and may be read as an input by sample applications.

  One LED is connected to PD0 on the MiniCore module, and may be driven as an output indicator by sample applications.

- **Ethernet**—One 10/100Base-T Ethernet port is available on the Interface Board with an RJ-45 jack at J3. **LNK** and **ACT** LEDs indicate a live Ethernet link and Ethernet activity.

B.2 Mechanical Dimensions and Layout

Figure B-2 shows the mechanical dimensions and layout for the Interface Board. All measurements are in inches followed by millimeters enclosed in parentheses.

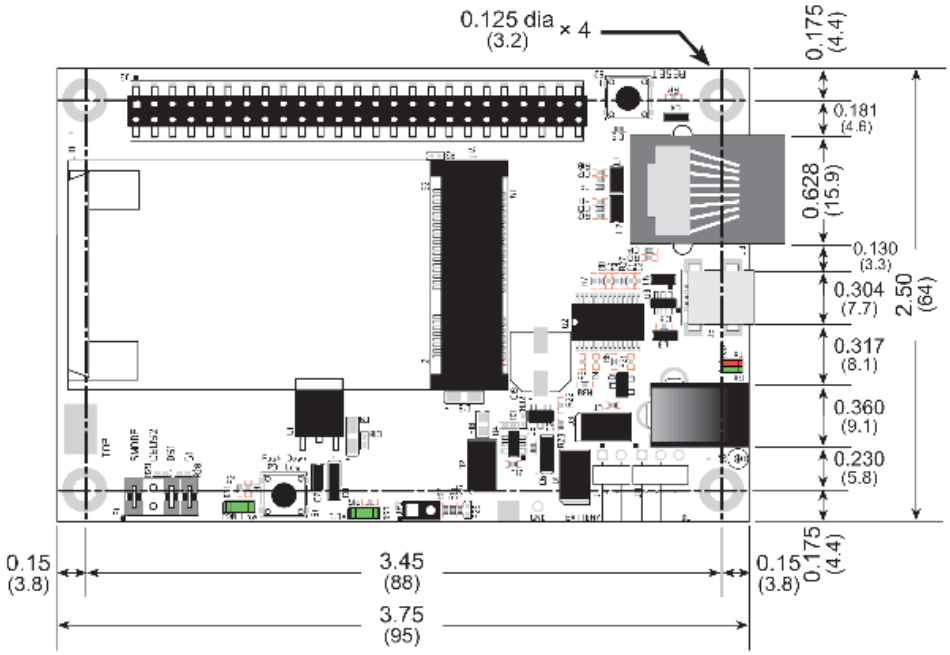


Figure B-2. Interface Board Dimensions

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

Table B-1. Interface Board Specifications

Parameter	Specification
Board Size	2.50" × 3.75" × 0.60" (64 mm × 95 mm × 15 mm)
Ethernet Port	10/100Base-T, RJ-45, 2 LEDs
Operating Temperature	–40°C to +85°C
Humidity	5% to 95%, noncondensing
Input Voltage	+5 V DC
Output Voltage	+3.3 V DC
Maximum Current Draw (including user-added circuits)	nominal 500 mA max. for USB supply, 1.5 A for separate power supply
Other Connectors	One 2 × 25 IDC header sockets, 0.1" pitch, One 52-pin mini PCI Express socket to accept RCM5700 One mini USB connector One 2 mm power supply jack
Standoffs/Spacers	4

B.2.1 Headers

The Interface Board has a header socket at J2 for physical connection to other boards. J2 is a 2 × 25 SMT header socket with a 0.1" pin spacing. Figure B-3 shows the layout of another board to be plugged into the Interface Board — this footprint is identical for the Prototyping Board and the two accessory boards. The values are relative to the mounting hole.

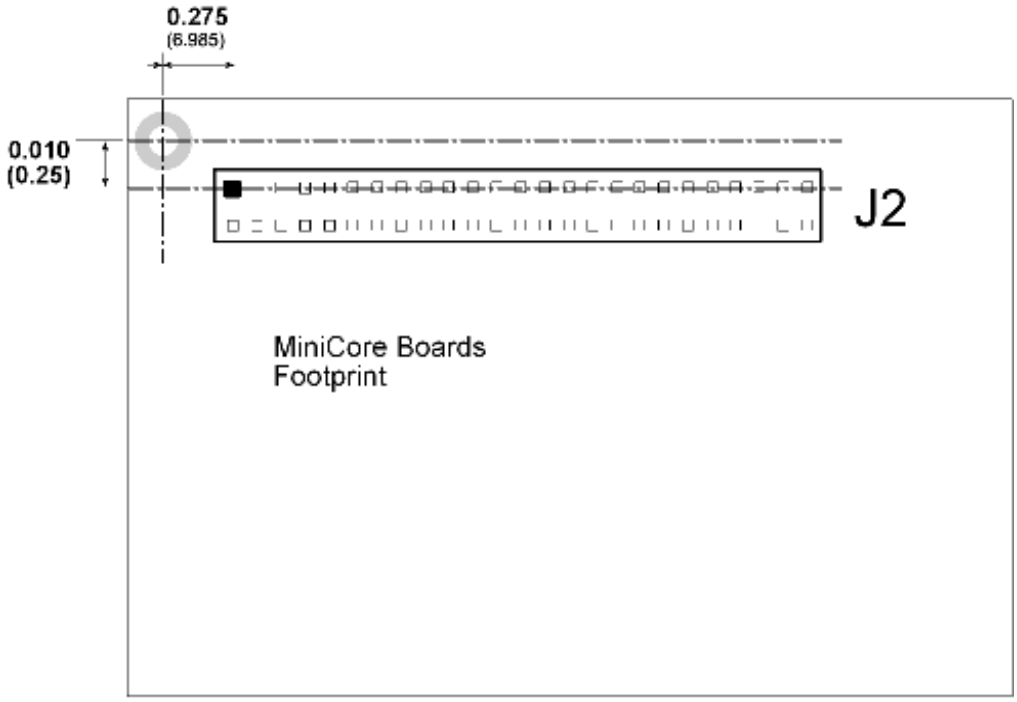


Figure B-3. Interface Board Footprint
(All measurements are in inches followed by millimeters enclosed in parentheses.)

B.3 Ethernet

B.3.1 RJ-45

The Ethernet filter circuit is different between the Rabbit 5000 and Rabbit 6000 Ethernet designs. If an RJ-45 jack is present on the MiniCore, then the appropriate circuitry is on the MiniCore. If the interface board Ethernet RJ-45 is to be used, then a given interface board will support Ethernet only for either the RCM5700 or RCM6700 family, but not both. The interface board supplied in the Development Kit is properly configured for the MiniCore in the kit.

B.3.2 RCM6700 LEDs

The RCM6700/RCM6750 only support a single off-board Ethernet status LED, unlike the RCM5700/RCM5750 which support two (LINK and ACT). The signal used as ACT on the RCM5700 is replaced with +2.5 V for the Ethernet circuitry on the RCM6700. To provide the same information, a combined LINK+ACT signal is used: off means no link, on means link, and blink means activity. The on-board Ethernet LEDs on the RJ-45 jack behave identically in both the RCM5700 and RCM6700.

If the RCM5700's two-LED configuration is desired for a RCM6700-based design, the behavior of the LINK+ACT signal must be changed to just LINK, and a GPIO pin must be assigned as the ACT signal. The Interface Board is designed to support either PE3 or PE5 as the ACT signal, and can be configured by defining macros and placing jumpers as shown in Table B-2.

Table B-2. Interface Board Ethernet LED Configuration for RCM6700

Mode	Behavior	JP3 Setting	Macros to define
Single LED	LINK off is no link LINK on is link, blink off for activity	4–6	None (default behavior)
Dual LEDs, ACT is PE3	LINK off is no link LINK on is link ACT off is no activity ACT on is activity (matches RCM5700)	1–3 4–6	`ENET_ACTIVITY_ON_PE3`
Dual LEDs, ACT is PE5		1–2 4–6	`ENET_ACTIVITY_ON_PE5`

The Ethernet LEDs are under software control on the RCM6700, so other configurations are possible as well. See the RCM6700-specific section of `LIB\Rabbit4000\TCPIP` folder for more details.

B.4 Power Supply

The MiniCore requires a regulated 3.15 V – 3.45 V DC power source to operate. Depending on the amount of current required by the application, different regulators can be used to supply this voltage.

The Interface Board has an onboard +3.3 V linear regulator. The Interface Board is protected against reverse polarity by a Shottky diode at D3 as shown in Figure B-4.

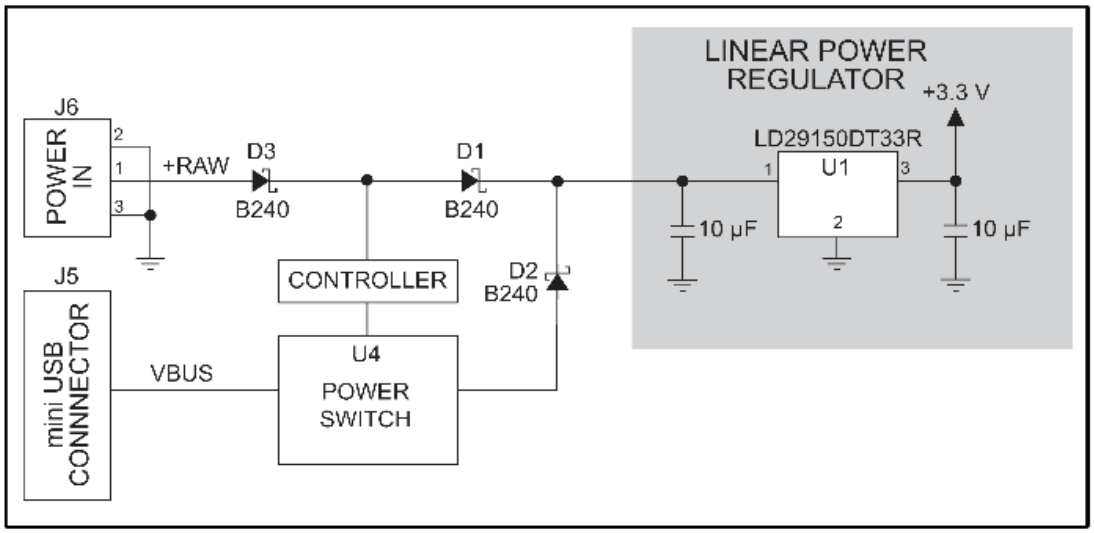


Figure B-4. Interface Board Power Supply

Power may be supplied to the Interface Board either via the mini USB connector at J5 or through the power supply jack J6. When a separate power supply is used, the chip at U4 disables power from the mini USB connector, which continues to supply power only to the USB interface chip. Diodes at D1 and D2 prevent power from going back to the other power supply that is not supplying power. A separate power supply is required whenever the Interface Board is not connected to the PC.

A jumper on header JP2 controls the current limiting applied to the power drawn via the mini USB connector — the current is nominally limited to 500 mA when no jumper is installed (default), and is nominally limited to 700 mA when a jumper is installed.

B.5 Using the Interface Board

The Interface Board is also a demonstration board. It can be used to demonstrate the functionality of the MiniCore right out of the box without any modifications to either board.

The Interface Board comes with the basic components necessary to demonstrate the operation of the MiniCore. One LEDs (DS1) is connected to PD0, and one switch (S1) is connected to PD1 to demonstrate the interface to the Rabbit microprocessor. Reset switch S2 is the hardware reset for the MiniCore.

The Interface Board provides the user with MiniCore connection points brought out conveniently to header socket J2. Other boards such as the Prototyping Board or the accessory boards from the Deluxe Development Kit can be plugged into header socket J2. The pinouts for header socket J2 are shown in Figure B-5.

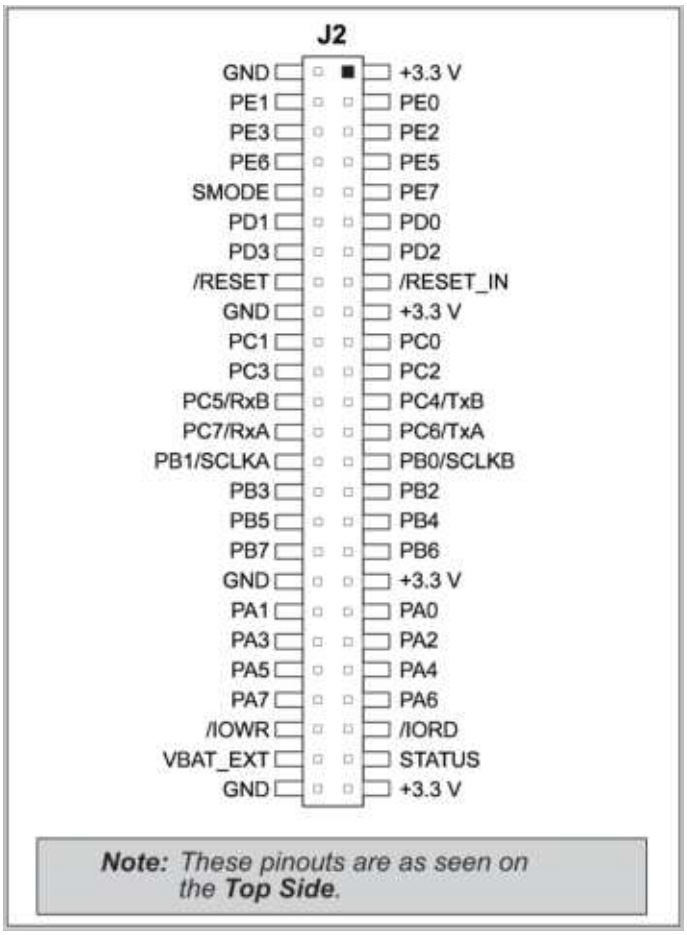


Figure B-5. Interface Board Pinout

B.5.1 Add Additional Boards

The Prototyping Board and the two accessory boards included with the Deluxe Development Kit may be installed on the Interface Board as shown in Figure B-6.

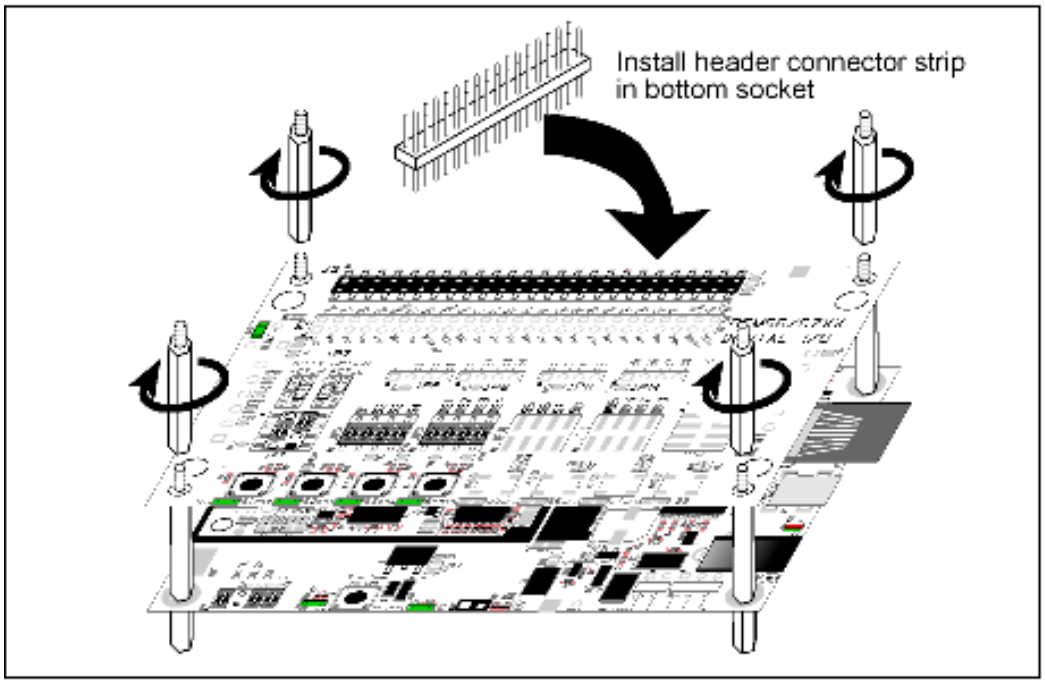


Figure B-6. Install Additional Boards on Interface Board

1. Insert the header strip into header socket J2 on the Interface Board or the board already installed above the Interface Board.
2. Line up the board being installed above the pins extending from the header socket and the stand-offs/connectors.
3. Press down to install the board.
4. Insert additional plastic standoffs/connectors as shown to hold the board firmly in place and to hold another board if desired.

When additional boards are installed, the board-to-board spacing is 0.7" (17.8 mm). Multiple boards should be installed in this sequence from bottom to top.

- Interface Board with MiniCore installed.
- Prototyping Board.
- Serial Communication accessory board.
- Digital I/O accessory board.

B.6 Interface Board Jumper Configurations

Figure B-7 shows the header locations used to configure the various Interface Board options via jumpers.

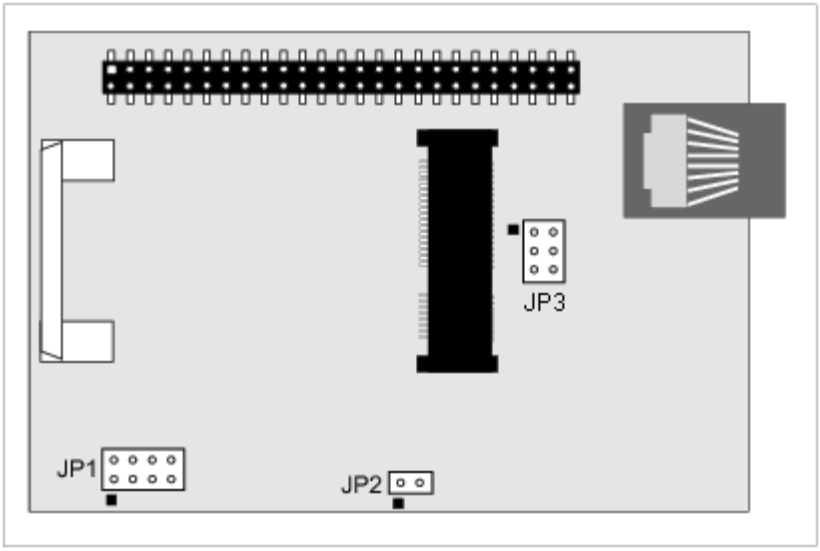


Figure B-7. Location of Configurable Jumpers on Interface Board

Table B-3 lists the configuration options using either jumpers or 0 Ω surface-mount resistors.

Table B-3. Interface Board Jumper Configurations

Header	Description	Pins Connected		Factory Default
JP1	Dynamic C Setup	1–2	SMODE pins pulled up (Programming Mode)*	×
		3–4	Reserved for future use	
		5–6	LED DS1 connected	×
		7–8	Switch S1 connected	×
JP2	mini USB Connector Power Supply Current Limiting	1–2	Nominal 700 mA	
		n.c.	Nominal 500 mA	×

* The MiniCore will operate in Run Mode when these pins are not jumpered.

APPENDIX C. PROTOTYPING BOARD

Appendix C describes the features and accessories of the Prototyping Board, and explains the use of the Prototyping Board to build prototypes of your own circuits. The Prototyping Board mounts on the Interface Board from which it receives its power and signals.

C.1 Introduction

The Prototyping Board included in the Development Kit provides a prototyping area for more advanced hardware development. The Prototyping Board is shown below in Figure C-1, with its main features identified.

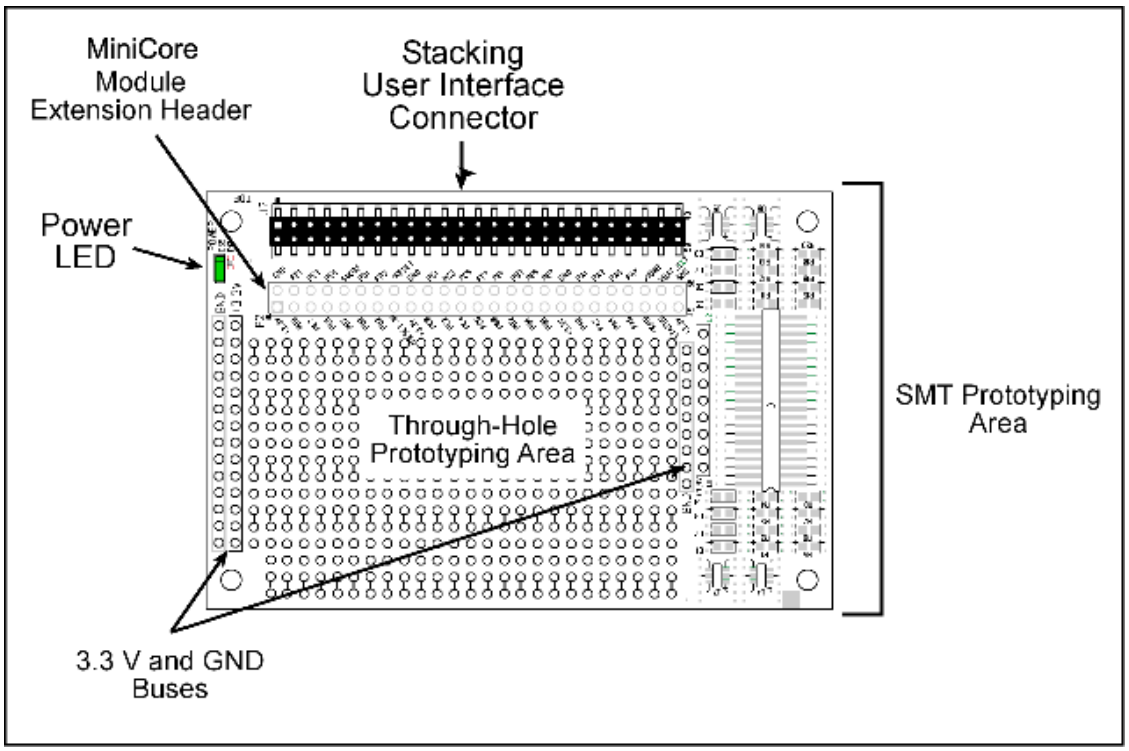


Figure C-1. Prototyping Board

C.1.1 Prototyping Board Features

- **Power Connection**—Power is supplied to the Prototyping Board via the MiniCore header socket connections.
- **Power LED**—The power LED lights whenever power is connected to the Prototyping Board.
- **Module Extension Headers**—The complete pin set of the MiniCore module is duplicated below header J2. Developers can solder wires directly into the appropriate holes, or, for more flexible development, a 2 × 25 header strip with a 0.1" pitch can be soldered into place. See Figure C-4 for the header pinouts.
- **Prototyping Area**—A generous prototyping area is provided for the installation of through-hole and surface-mount components. +3.3 V and ground buses run along the left and right edges of the through-hole prototyping area. The through-hole area is set up to accept components with a pitch of 0.1" or widths of 0.3" or 0.6". Several areas for surface-mount devices are also available. (Note that there are SMT pads on both the top and the bottom of the Prototyping Board.) Each SMT pad is connected to a hole designed to accept a 30 AWG solid wire.

C.2 Mechanical Dimensions and Layout

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

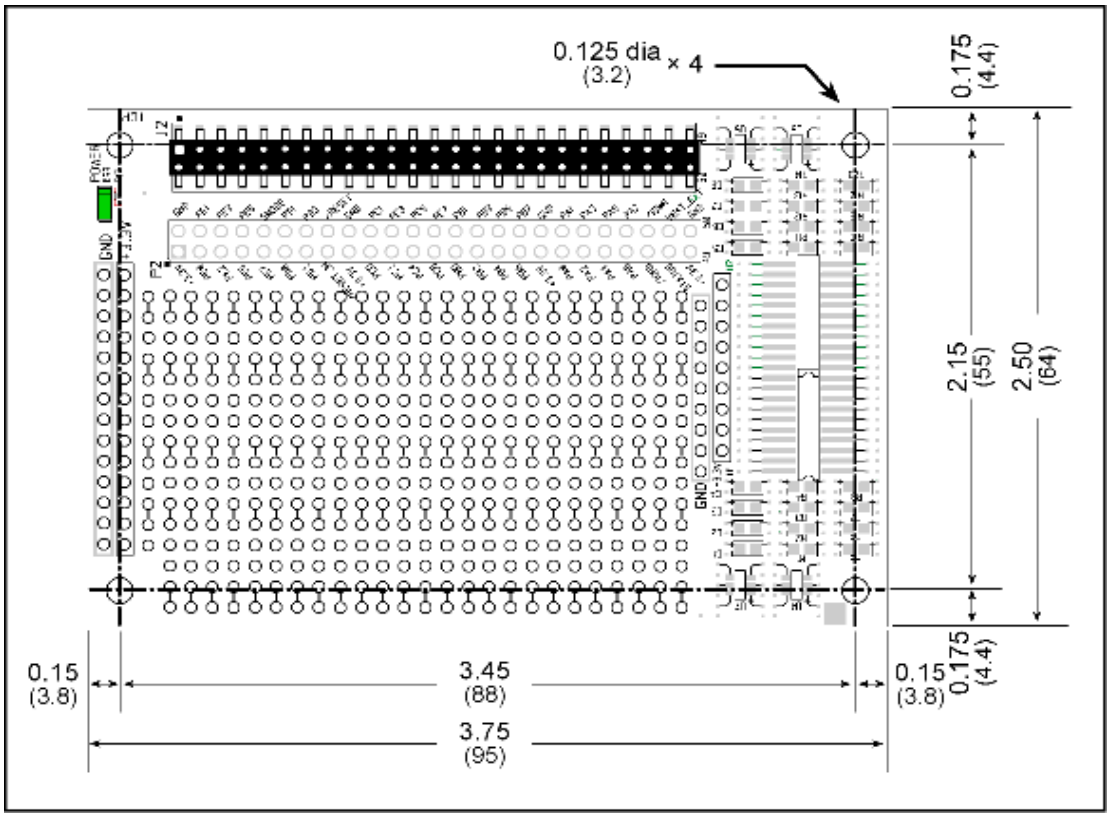


Figure C-2. Prototyping Board Dimensions
(All measurements are in inches followed by millimeters enclosed in parentheses.)

Table C-1 lists the electrical, mechanical, and environmental specifications for the Prototyping Board.

Table C-1. Prototyping Board Specifications

Parameter	Specification
Board Size	2.50" × 3.75" × 0.52" (64 mm × 95 mm × 13 mm)
Operating Temperature	–40°C to +85°C
Humidity	5% to 95%, noncondensing
Operating Voltage	+3.3 V DC
Current Draw from Interface Board (excluding user-added circuits)	2 mA
Prototyping Area	1.7" × 2.7" (40 mm × 70 mm) throughhole, 0.1" spacing, additional space for SMT components
Connectors	Two 2 × 25 IDC header sockets, 0.1" pitch (a 2 × 25 IDC header strip is included to connect the Prototyping Board to the Interface Board below it)
Standoffs/Spacers	4

C.2.1 Headers

The Prototyping Board has a header socket at J2 for physical connection to other boards above it, and a header socket at J12 on the bottom side to connect to boards below it. J2 and J12 are 2 × 25 SMT header sockets with a 0.1" pin spacing. Figure C-3 shows the layout of another board to be plugged into the Interface Board — this footprint is identical for the Prototyping Board and the two accessory boards. The values are relative to the mounting hole.

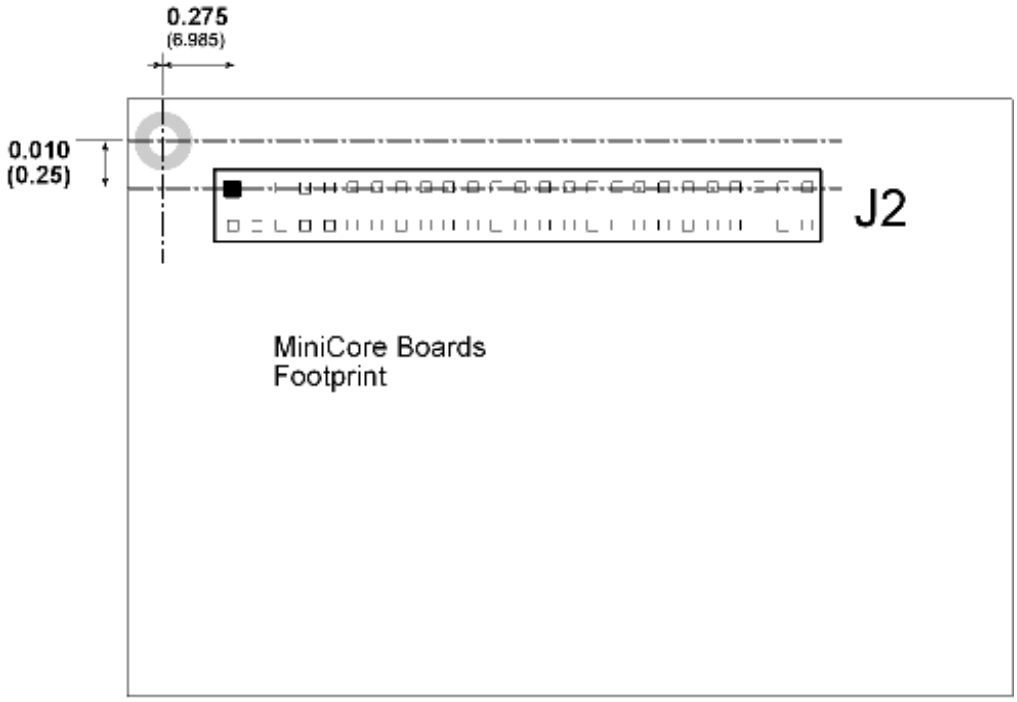


Figure C-3. MiniCore Boards Footprint
(All measurements are in inches followed by millimeters enclosed in parentheses.)

C.3 Using the Prototyping Board

The Prototyping Board provides the user with MiniCore connection points brought out conveniently to labeled points below header J2. The pinouts for header socket J2 are shown in Figure C-4.

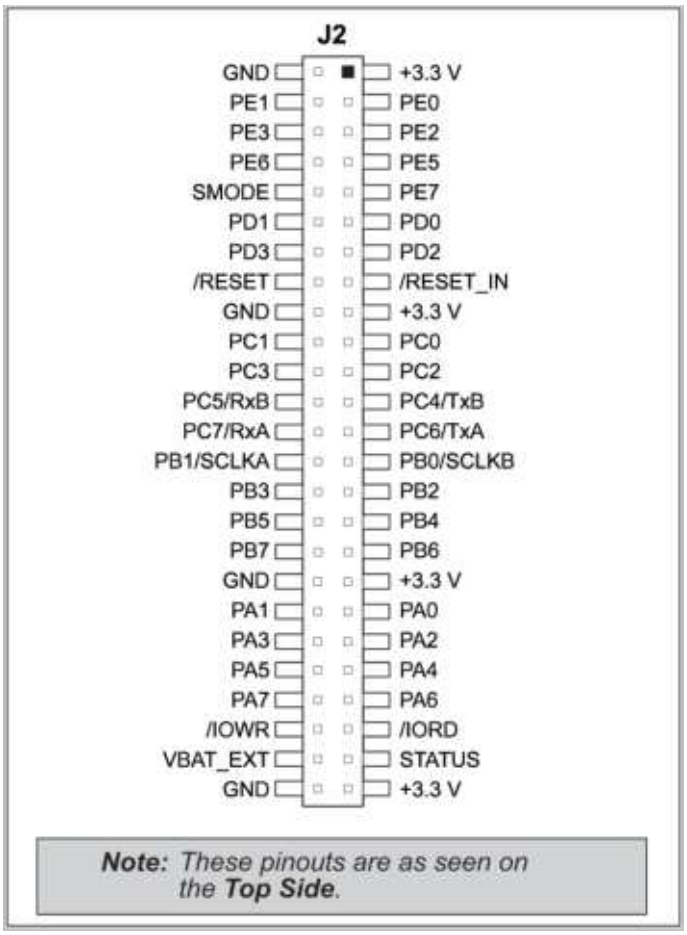


Figure C-4. MiniCore Boards Pinout

There is a 1.7" × 2.7" through-hole prototyping space available on the Prototyping Board. The holes in the prototyping area are spaced at 0.1" (2.5 mm). +3.3 V and GND traces run along the left edge of the Prototyping Board for easy access. Small to medium circuits can be prototyped using point-to-point wiring with 20 to 30 AWG wire between the prototyping area, the +3.3 V and GND traces, and the surrounding area where surface-mount components may be installed. Small holes are provided around the surface-mounted components that may be installed around the prototyping area.

C.3.1 Add Additional Boards

The Prototyping Board and the two accessory boards included with the Deluxe Development Kit may be installed on the Interface Board as shown in Figure C-5.

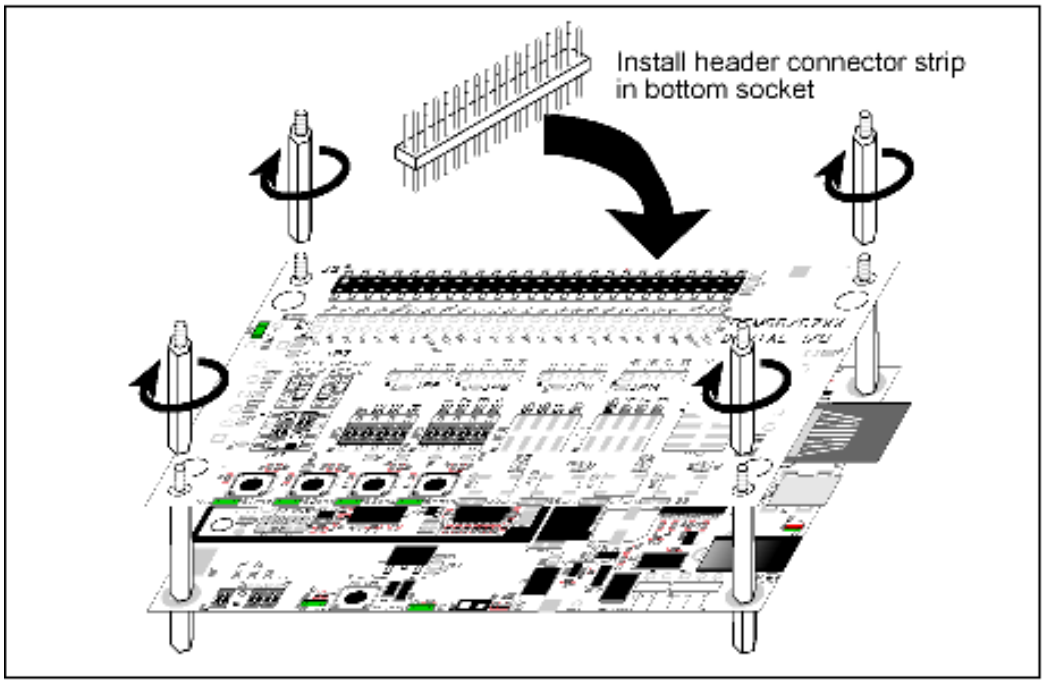


Figure C-5. Install Additional Boards

1. Insert the header strip into header socket J2 on the Interface Board or the board already installed above the Interface Board.
2. Line up the board being installed above the pins extending from the header socket and the stand-offs/connectors.
3. Press down to install the board.
4. Insert additional plastic standoffs/connectors as shown to hold the board firmly in place and to hold another board if desired.

When additional boards are installed, the board-to-board spacing is 0.7" (17.8 mm). Multiple boards should be installed in this sequence from bottom to top.

- Interface Board with MiniCore installed.
- Prototyping Board.
- Serial Communication accessory board.
- Digital I/O accessory board.

APPENDIX D. DIGITAL I/O ACCESSORY BOARD

Appendix D describes the features and accessories of the Digital I/O accessory board, and explains how to use the Digital I/O accessory board. The Digital I/O accessory board mounts on the Interface Board or other board already installed on the Interface Board from which it receives its power and signals.

D.1 Introduction

The Digital I/O accessory board included in the Deluxe Development Kit provides Pushbutton switches and LEDs to use in conjunction with selected sample programs. The Digital I/O accessory board is shown below in Figure D-1, with its main features identified.

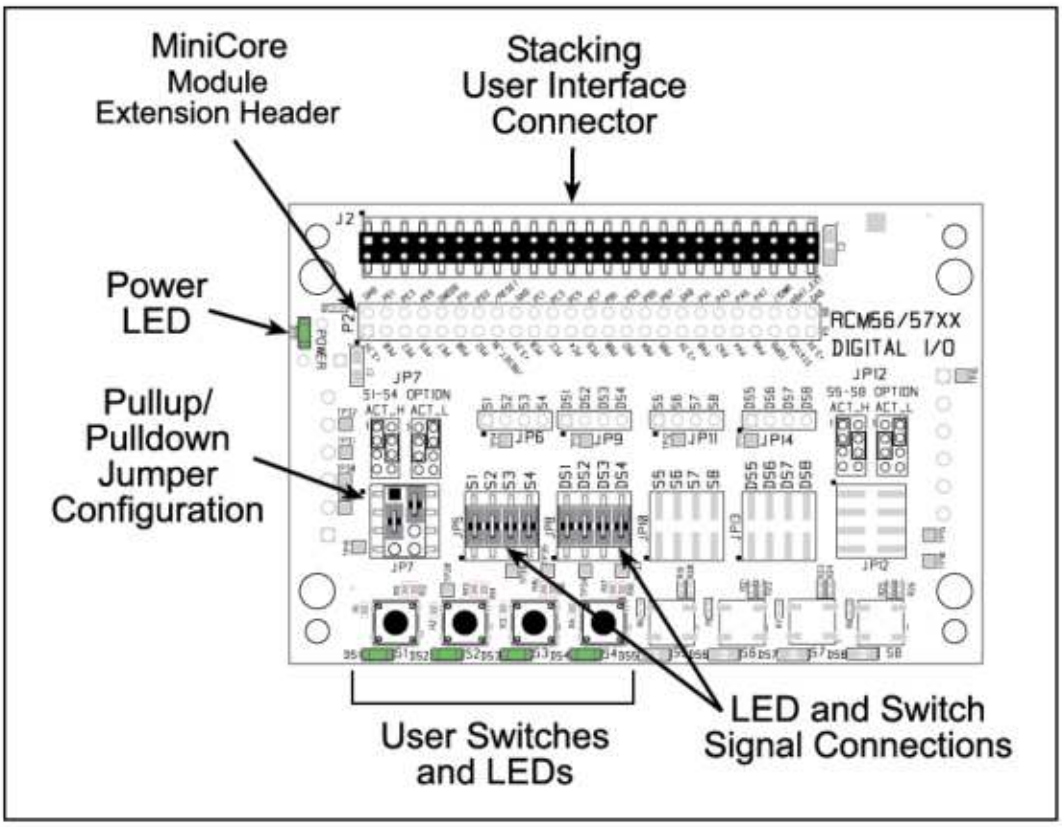


Figure D-1. Digital I/O Accessory Board

D.1.1 Digital I/O Accessory Board Features

- **Power Connection**—Power is supplied to the Digital I/O accessory board via the MiniCore header socket connections.
- **Power LED**—The power LED lights whenever power is connected to the Digital I/O accessory board.
- **Module Extension Headers**—The complete pin set of the MiniCore module is duplicated below header J2. Developers can solder wires directly into the appropriate holes, or, for more flexible development, a 2 × 25 header strip with a 0.1" pitch can be soldered into place. See Figure D-4 for the header pinouts.
- **I/O Switches and LEDs**—Four momentary-contact, normally open switches are connected to PB4–PB7 on the MiniCore module and may be read as an input by sample applications.

  Four LEDs are connected to PA4–PA7 on the MiniCore module, and may be driven as an output indicator by sample applications.

D.2 Mechanical Dimensions and Layout

Figure D-2 shows the mechanical dimensions and layout for the Digital I/O accessory board.

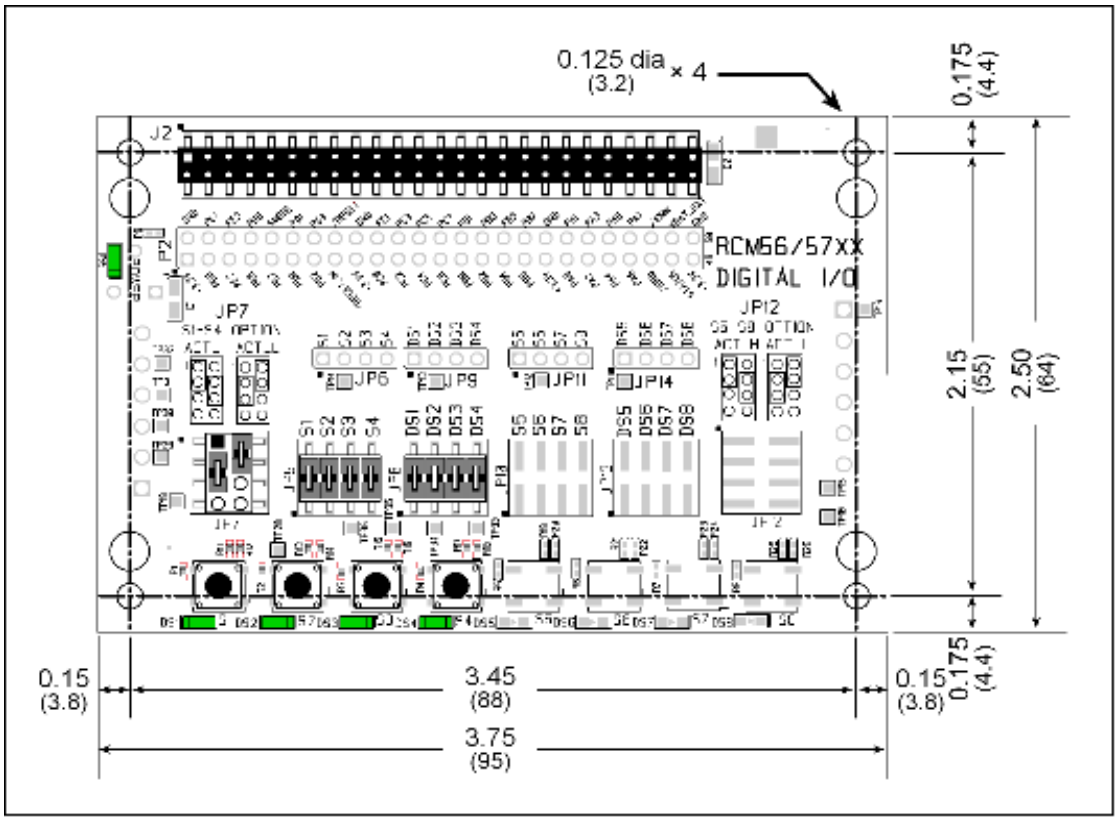


Figure D-2. Digital I/O Accessory Board Dimensions
(All measurements are in inches followed by millimeters enclosed in parentheses.)

Table D-1 lists the electrical, mechanical, and environmental specifications for the Digital I/O accessory board.

Table D-1. Digital I/O Accessory Board Specifications

Parameter	Specification
Board Size	2.50" × 3.75" × 0.52" (64 mm × 95 mm × 13 mm)
Operating Temperature	–40°C to +85°C
Humidity	5% to 95%, noncondensing
Operating Voltage	+3.3 V DC
Current Draw from Interface Board	6 mA (typical)
Connectors	Two 2 × 25 IDC header sockets, 0.1" pitch (a 2 × 25 IDC header strip is included to connect the Digital I/O accessory board to the board below it)
Standoffs/Spacers	4

D.2.1 Headers

The Digital I/O accessory board has a header socket at J2 for physical connection to other boards above it, and a header socket at J12 on the bottom side to connect to boards below it. J2 and J12 are 2 × 25 SMT header sockets with a 0.1" pin spacing. Figure D-3 shows the layout of another board to be plugged into the Digital I/O accessory board — this footprint is identical for the Prototyping Board and the two accessory boards. The values are relative to the mounting hole.

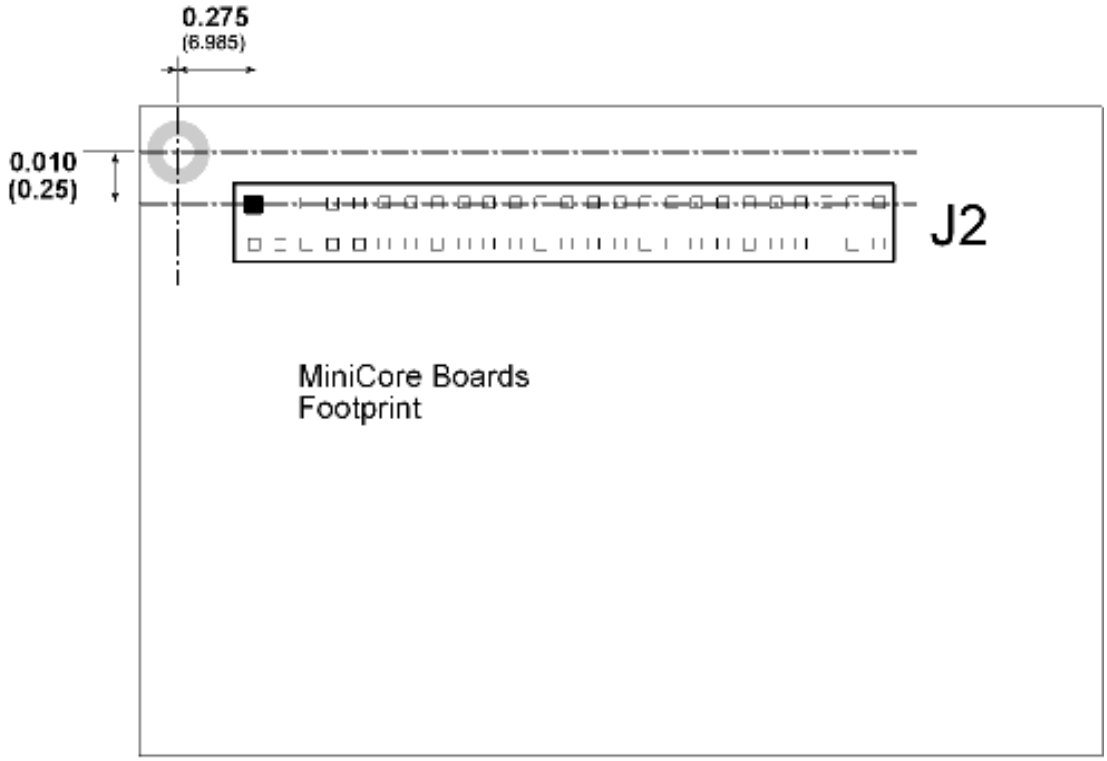


Figure D-3. MiniCore Boards Footprint
(All measurements are in inches followed by millimeters enclosed in parentheses.)

D.3 Using the Digital I/O Accessory Board

The Digital I/O accessory board provides the user with MiniCore connection points brought out conveniently to labeled points below header J2. The pinouts for header socket J2 are shown in Figure D-4.

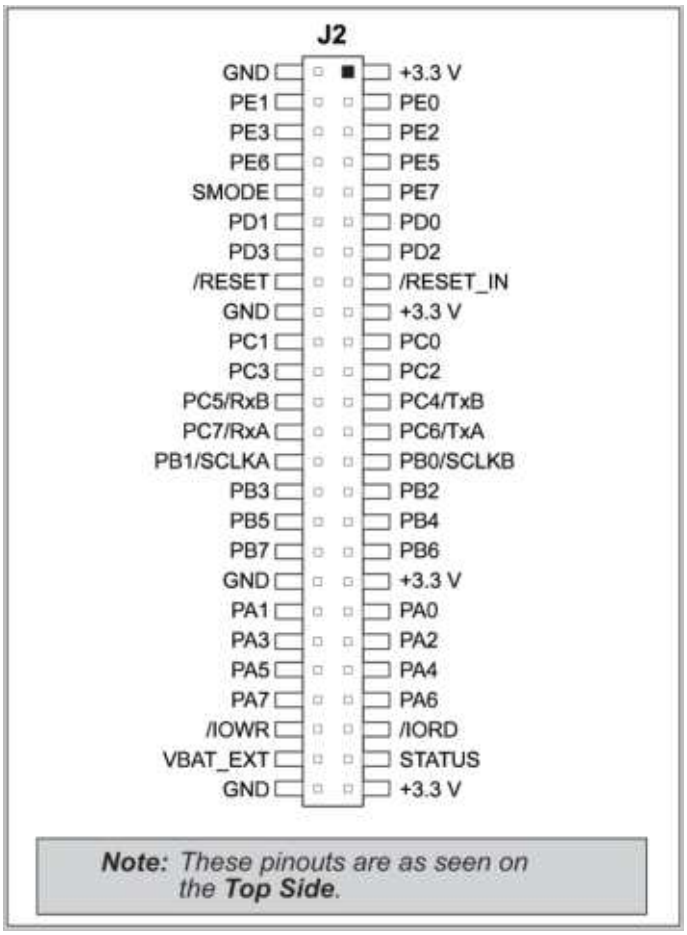


Figure D-4. MiniCore Boards Pinout

D.3.1 Configuration

The pushbutton switches may be configured active high (pulled down) or active low (pulled up) via jumper settings on header JP7 for the four switches installed. Jumpers on JP12 may be set up in a similar way after additional switches are installed at S5–S8.

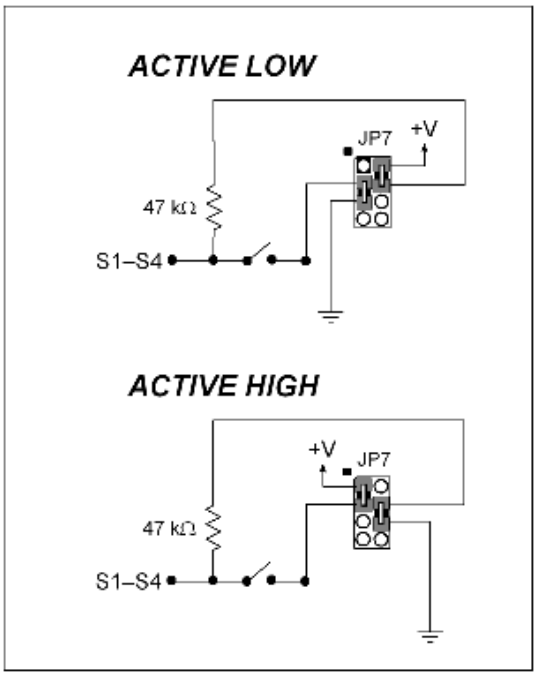


Figure D-5. Pushbutton Switch Configuration

The four LED output indicators are set up as sinking outputs. Four additional LEDs may be installed at DS5–DS8.

Jumpers on headers JP5 and JP8 connect the MiniCore signals to the pushbutton switches or LEDs. These jumpers may be removed and other MiniCore signals may be connected to the switch or LED positions above these headers via headers JP6 and JP9. Table D-2 lists the connection options for the switches and LEDs.

Table D-2. Digital I/O Accessory Board Switch/LED Connection Options

Default MiniCore Signal	Switch/LED	Connected via		Alternate Connection	
		Header	Pins	Header	Pin
PB4	S1*	JP5	1–2	JP6	1
PB5	S2		3–4		2
PB6	S3		5–6		3
PB7	S4		7–8		4

Table D-2. Digital I/O Accessory Board Switch/LED Connection Options

Default MiniCore Signal	Switch/LED	Connected via		Alternate Connection	
		Header	Pins	Header	Pin
PA4	DS1	JP8	1–2	JP9	1
PA5	DS2		3–4		2
PA6	DS3		5–6		3
PA7	DS4		7–8		4
PB0	S5†	JP10	1–2	JP11	1
PB1	S6		3–4		2
PB2	S7		5–6		3
PB3	S8		7–8		4
PA0	DS5	JP13	1–2	JP14	1
PA1	DS6		3–4		2
PA2	DS7		5–6		3
PA3	DS8		7–8		4

* Switches S1–S4 are pulled high or low via jumpers on header JP7.
† Switches S5–S8 are pulled high or low via jumpers on header JP12 (not stuffed).

NOTE: Switches S5–S8, LEDs DS5–DS8, and the corresponding configuration headers JP10–JP14 and circuits are not stuffed.

Figure D-6 shows the locations of the configurable jumpers.

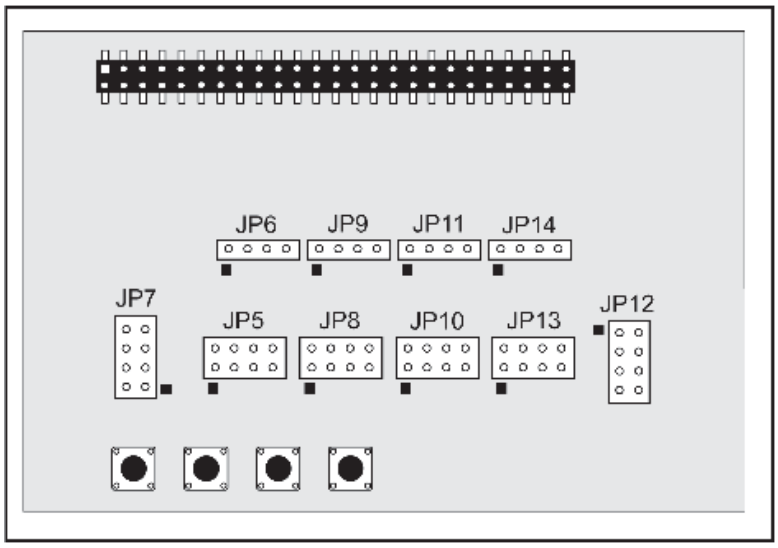


Figure D-6. Location of Configurable Jumpers on Digital I/O Accessory Board

D.3.2 Add Additional Boards

The Prototyping Board and the two accessory boards included with the Deluxe Development Kit may be installed on the Interface Board as shown in Figure D-7.

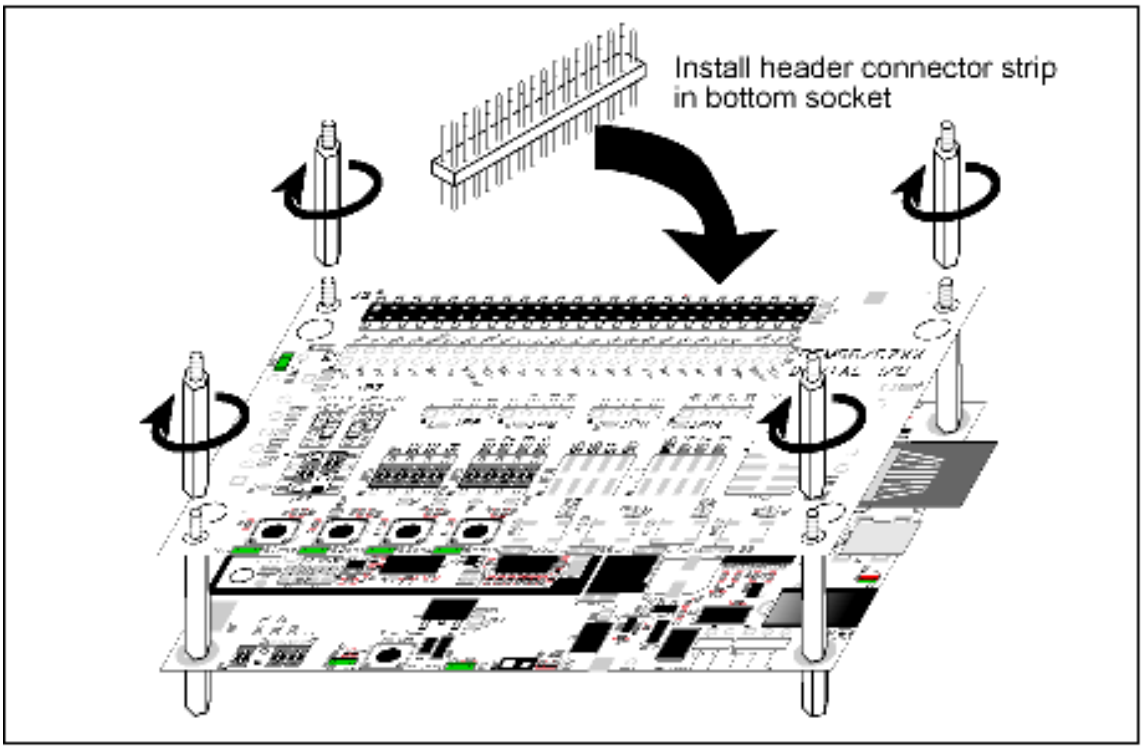


Figure D-7. Install Additional Boards

1. Insert the header strip into header socket J2 on the Interface Board or the board already installed above the Interface Board.
2. Line up the board being installed above the pins extending from the header socket and the stand-offs/connectors.
3. Press down to install the board.
4. Insert additional plastic standoffs/connectors as shown to hold the board firmly in place and to hold another board if desired.

When additional boards are installed, the board-to-board spacing is 0.7" (17.8 mm). Multiple boards should be installed in this sequence from bottom to top.

1. Interface Board with MiniCore installed.
2. Prototyping Board.
3. Serial Communication accessory board.
4. Digital I/O accessory board.

Appendix E. Serial Communication Accessory Board

Appendix E describes the features and accessories of the Serial Communication accessory board, and explains how to use the Serial Communication accessory board. The Serial Communication accessory board mounts on the Interface Board or other board already installed on the Interface Board from which it receives its power and signals.

E.1 Introduction

The Serial Communication accessory board included in the Deluxe Development Kit provides two 3-wire serial ports to use in conjunction with selected sample programs. The Serial Communication accessory board is shown below in Figure E-1, with its main features identified.

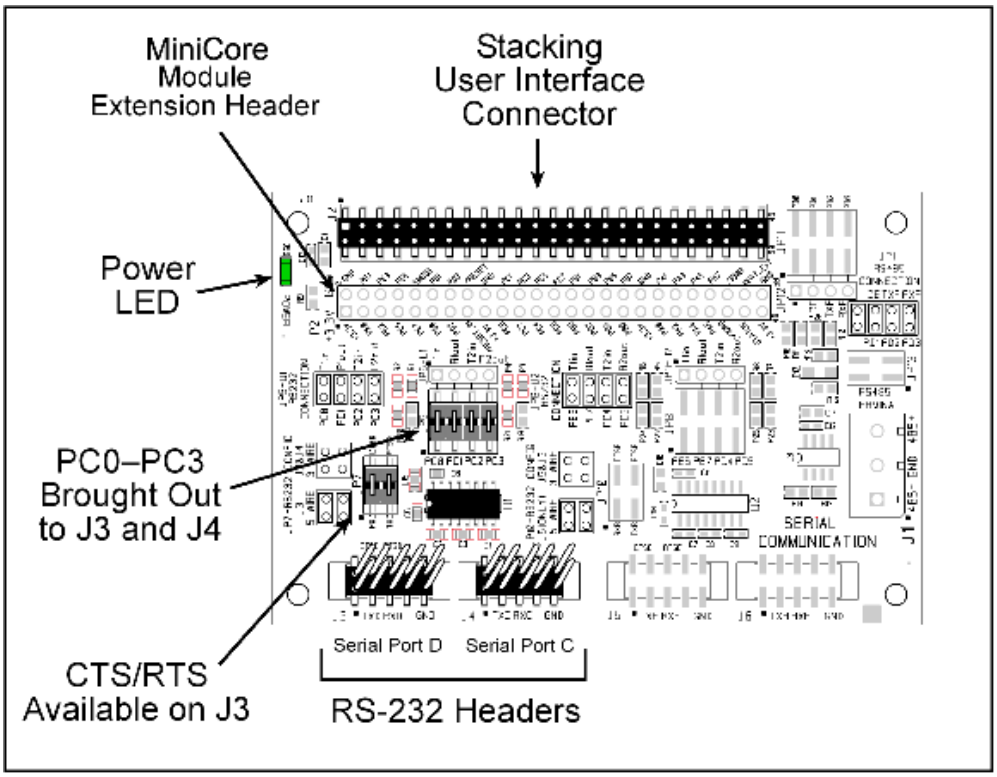


Figure E-1. Serial Communication Accessory Board

E.1.1 Serial Communication Accessory Board Features

- **Power Connection**—Power is supplied to the Serial Communication accessory board via the MiniCore header socket connections.
- **Power LED**—The power LED lights whenever power is connected to the Serial Communication accessory board.
- **Module Extension Headers**—The complete pin set of the MiniCore module is duplicated below header J2. Developers can solder wires directly into the appropriate holes, or, for more flexible development, a 2 × 25 header strip with a 0.1" pitch can be soldered into place. See Figure E-4 for the header pinouts.
- **RS-232 Headers**—Serial Ports C and D are brought out as 3-wire RS-232 ports on headers J4 and J3 respectively. Header J4 can be set up as a 5-wire RS-232 serial port with flow control provided by Serial Port C.

E.2 Mechanical Dimensions and Layout

Figure E-2 shows the mechanical dimensions and layout for the Serial Communication accessory board. (All measurements are in inches followed by millimeters enclosed in parentheses.)

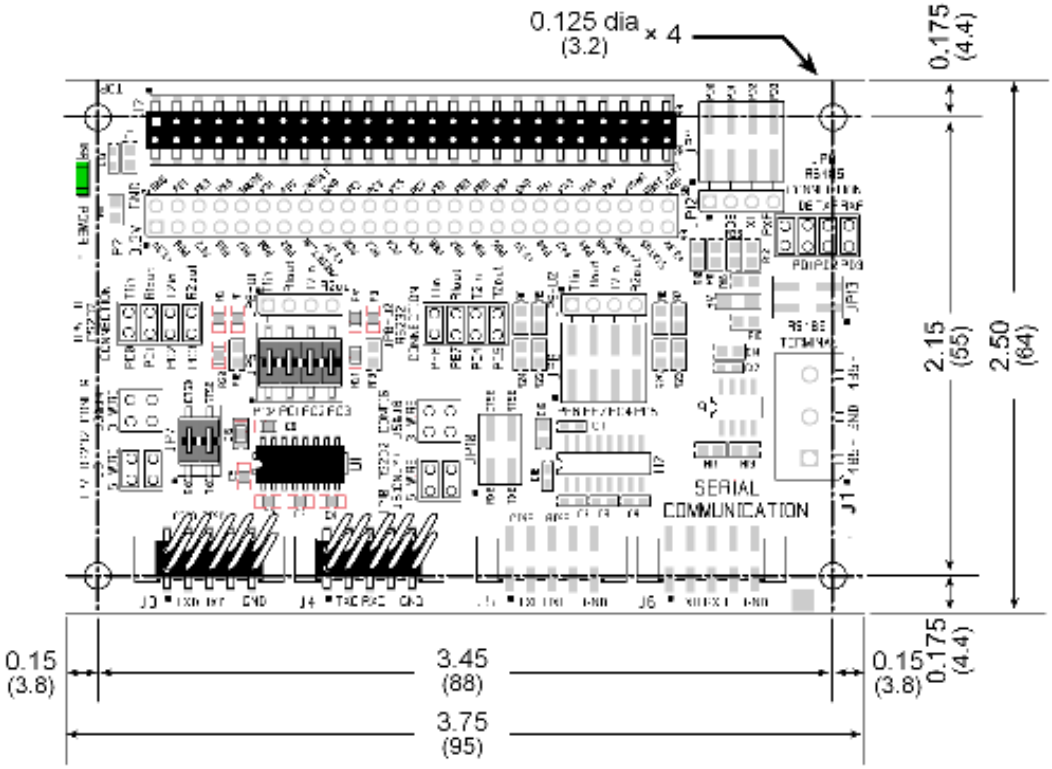


Figure E-2. Serial Communication Accessory Board Dimensions

Table E-1 lists the electrical, mechanical, and environmental specifications for the Serial Communication accessory board.

Table E-1. Serial Communication Accessory Board Specifications

Parameter	Specification
Board Size	2.50" × 3.75" × 0.52" (64 mm × 95 mm × 13 mm)
Operating Temperature	–40°C to +85°C
Humidity	5% to 95%, noncondensing
Operating Voltage	+3.3 V DC
Current Draw from Interface Board	10 mA (typical)
Connectors	Two 2 × 25 IDC header sockets, 0.1" pitch (a 2 × 25 IDC header strip is included to connect the Serial Communication accessory board to the board below it) Two 2 × 5 IDC headers, 0.1" pitch
Standoffs/Spacers	4

E.2.1 Headers

The Serial Communication accessory board has a header socket at J2 for physical connection to other boards above it, and a header socket at J12 on the bottom side to connect to boards below it. J2 and J12 are 2 × 25 SMT header sockets with a 0.1" pin spacing. Figure E-3 shows the layout of another board to be plugged into the Serial Communication accessory board — this footprint is identical for the Prototyping Board and the two accessory boards. The values are relative to the mounting hole.

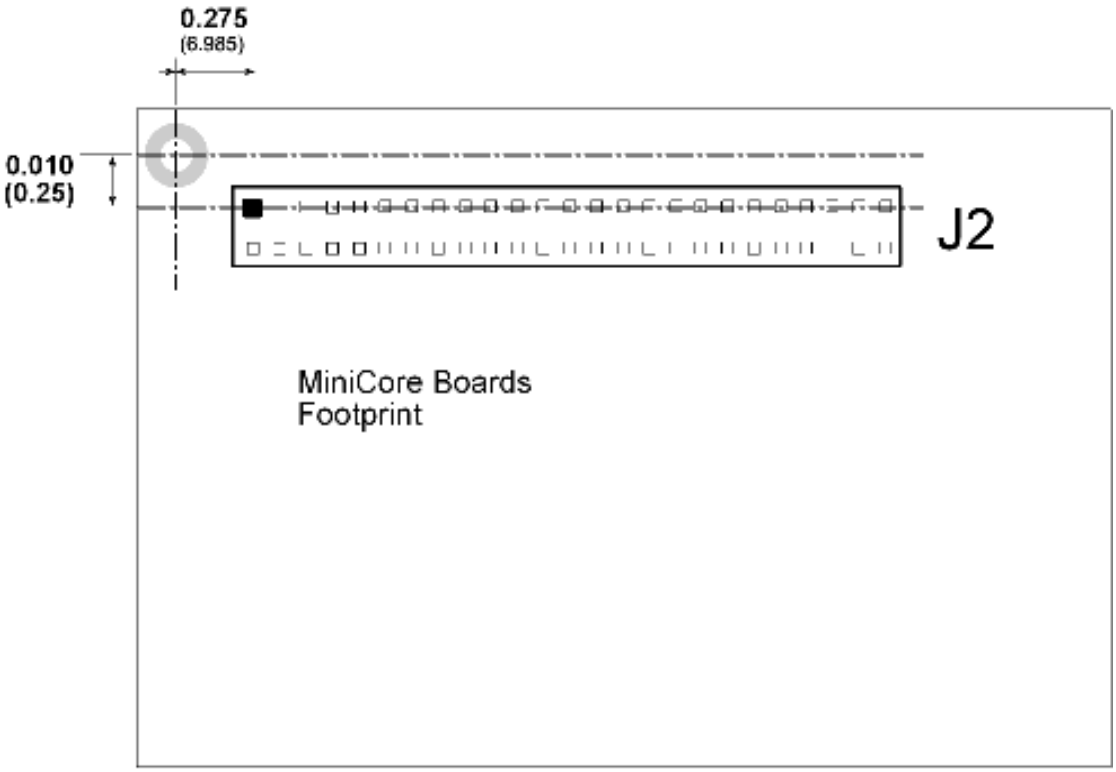


Figure E-3. MiniCore Boards Footprint
(All measurements are in inches followed by millimeters enclosed in parentheses.)

E.3 Using the Serial Communication Accessory Board

The Serial Communication accessory board provides the user with MiniCore connection points brought out conveniently to labeled points below header J2. The pinouts for header socket J2 and the RS-232 headers at J3 and J4 are shown in Figure E-4.

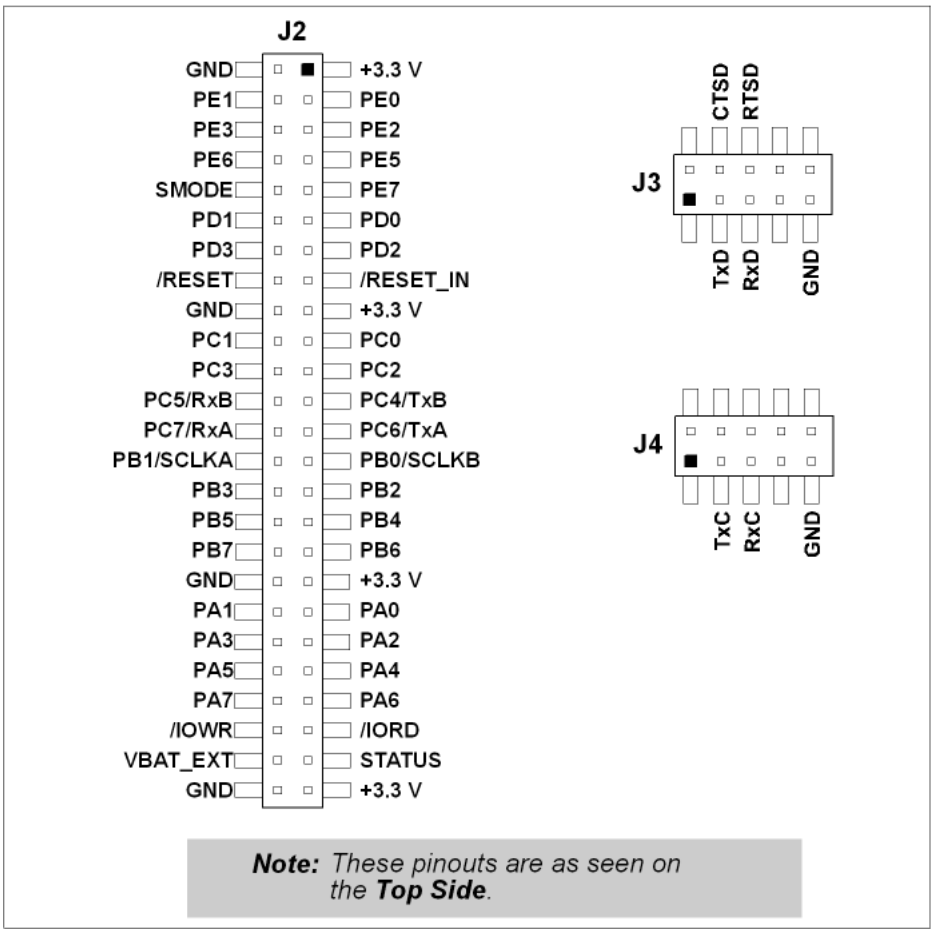


Figure E-4. Serial Communication Accessory Board Pinout

The remaining RS-232 header positions at J5 and J6, and the RS-485 screw-terminal header position at J1 are unstuffed.

E.3.1 Configuration

Serial Ports C and D are brought out as 3-wire RS-232 serial ports on headers J4 and J3 respectively. Jumpers may be installed on header JP7 to use header J3 as a 5-wire RS-232 serial port with flow control provided by Serial Port C.

Jumpers on headers JP5 connect the MiniCore signals to the RS-232 transceiver. Jumpers may be installed on header JP7 to use header J3 as a 5-wire RS-232 serial port with flow control provided by Serial Port C. Note that Serial Port C does not support flow control using serial DMA, so the following macro must be used with flow control via Serial Port C on the Serial Communication accessory board.

```
#define SER_DMA_DISABLE
```

The jumpers at header JP5 connect the Serial Port D and Serial Port C signals to the RS-232 transceiver. These jumpers may be removed so that other MiniCore serial port signals may be connected via JP6 to the RS-232 transceiver.

Table E-2. Serial Communication Accessory Board RS-232 Connection Options

Default MiniCore Signal	Header	Serial Port	Connected via		Alternate Connection	
			Header	Pins	Header	Pin
PC0	J3	Serial Port D (RS-232)	JP5	1–2	JP6	1
PC1				3–4		2
PC2	J3/J4*	Serial Port C (RS-232)		5–6		3
PC3				7–8		4
PE6	J5	Serial Port E (RS-232)	JP8	1–2	JP9	1
PE7				3–4		2
PC4	J5/J6†	Serial Port B (RS-232)		5–6		3
PC5				7–8		4
PD0	—		JP11	1–2	JP12	1
PD1	J1‡	Flow Enable		3–4		2
PD2		Serial Port F (RS-485)		5–6		3
PD3				7–8		4

* Configured via header JP7.
† Configured via header JP10 (unstuffed).
‡ Termination and bias resistors enabled via header JP13 (unstuffed).

NOTE: Headers J1, J5, J6, and the associated circuits and configuration headers are not stuffed.

Figure E-5 shows the locations of the configurable header positions.

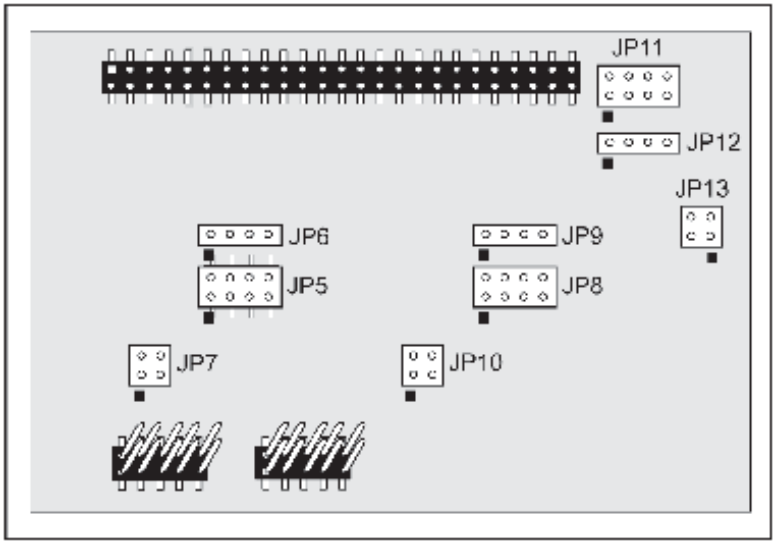


Figure E-5. Location of Configurable Jumpers on Serial Communication Accessory Board

E.3.2 Add Additional Boards

The Prototyping Board and the two accessory boards included with the Deluxe Development Kit may be installed on the Interface Board as shown in Figure E-6.

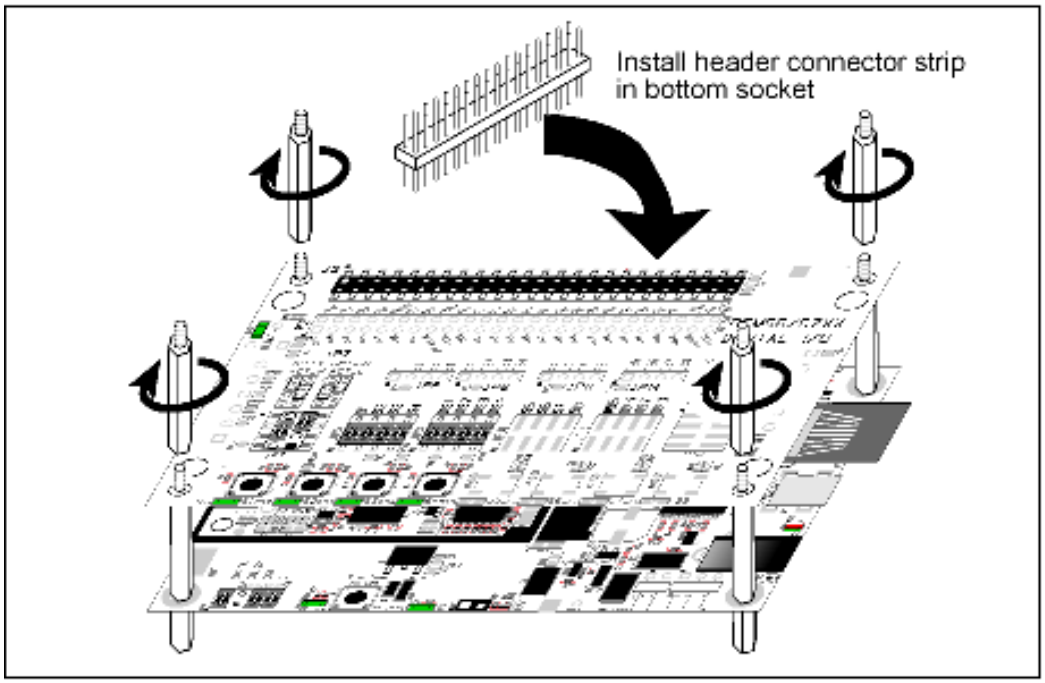


Figure E-6. Install Additional Boards

1. Insert the header strip into header socket J2 on the Interface Board or the board already installed above the Interface Board.
2. Line up the board being installed above the pins extending from the header socket and the stand-offs/connectors.
3. Press down to install the board.
4. Insert additional plastic standoffs/connectors as shown to hold the board firmly in place and to hold another board if desired.

When additional boards are installed, the board-to-board spacing is 0.7" (17.8 mm). Multiple boards should be installed in this sequence from bottom to top.

1. Interface Board with MiniCore installed.
2. Prototyping Board.
3. Serial Communication accessory board.
4. Digital I/O accessory board.

APPENDIX F. USING THE TCP/IP FEATURES

F.1 TCP/IP Connections

Programming and development can be done with the MiniCore without connecting the Ethernet port to a network. However, if you will be running the sample programs that use the Ethernet capability or will be doing Ethernet-enabled development, you should connect the Ethernet port on the Interface Board at this time.

Before proceeding you will need to have the following items.

- If you don't have Ethernet access, you will need at least a 10Base-T Ethernet card (available from your favorite computer supplier) installed in a PC.
- Two CAT 4/5 Ethernet cables and a hub, or one CAT 4/5 Ethernet cable.

NOTE: The Ethernet cables and a 10Base-T Ethernet hub are available from Rabbit in a TCP/IP tool kit. More information is available at www.digi.com.

Now you should be able to make your connections.

1. Connect the AC adapter and the USB cable as shown in Chapter 2, "Getting Started."
2. Ethernet Connections

   There are four options for connecting the Interface Board to a network for development and runtime purposes. The first two options permit total freedom of action in selecting network addresses and use of the "network," as no action can interfere with other users. We recommend one of these options for initial development.

   No LAN — The simplest alternative for desktop development. Connect the Interface Board's Ethernet port directly to the PC's network interface card using an CAT 4/5 Ethernet cable.

   Micro-LAN — Another simple alternative for desktop development. Use a small Ethernet 10Base-T hub and connect both the PC's network interface card and the Interface Board's Ethernet port to it using standard network cables.

The following options require more care in address selection and testing actions, as conflicts with other users, servers and systems can occur:

LAN — Connect the Interface Board's Ethernet port to an existing LAN, preferably one to which the development PC is already connected. You will need to obtain IP addressing information from your network administrator.

WAN — The RCM5700/RCM6700 is capable of direct connection to the Internet and other Wide Area Networks, but exceptional care should be used with IP address settings and all network-related programming and development. We recommend that development and debugging be done on a local network before connecting a Rabbit-based system to the Internet.

NOTE: Checking and debugging the initial setup on a micro-LAN is recommended before connecting the system to a LAN or WAN.

The PC running Dynamic C does not need to be the PC with the Ethernet card.

3. Apply Power

   The MiniCore module and Interface Board are now ready to be used.

F.2 TCP/IP Primer on IP Addresses

Obtaining IP addresses to interact over an existing, operating, network can involve a number of complications, and must usually be done with cooperation from your ISP and/or network systems administrator. For this reason, it is suggested that the user begin instead by using a direct connection between a PC and the Interface Board using a CAT 4/5 Ethernet cable or a simple arrangement with a hub.

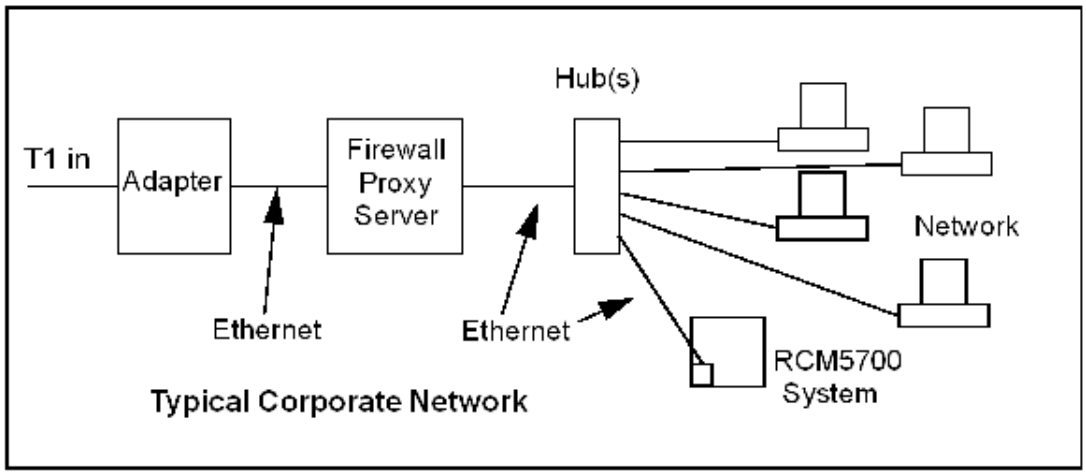
In order to set up this direct connection, the user will have to use a PC without networking, or disconnect a PC from the corporate network, or install a second Ethernet adapter and set up a separate private network attached to the second Ethernet adapter. Disconnecting your PC from the corporate network may be easy or nearly impossible, depending on how it is set up. If your PC boots from the network or is dependent on the network for some or all of its disks, then it probably should not be disconnected. If a second Ethernet adapter is used, be aware that Windows TCP/IP will send messages to one adapter or the other, depending on the IP address and the binding order in Microsoft products. Thus you should have different ranges of IP addresses on your private network from those used on the corporate network. If both networks service the same IP address, then Windows may send a packet intended for your private network to the corporate network. A similar situation will take place if you use a dial-up line to send a packet to the Internet. Windows may try to send it via the local Ethernet network if it is also valid for that network.

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

The Interface Board uses a 10/100Base-T type of Ethernet connection, which is the most common scheme. The RJ-45 connectors are similar to U.S. style telephone connectors, except they are larger and have 8 contacts.

An alternative to the direct connection using a CAT 4/5 Ethernet cable is a direct connection using a hub. The hub relays packets received on any port to all of the ports on the hub. Hubs are low in cost and are readily available. The RCM5700/RCM6700 uses 10/100Mps Ethernet, so the hub or Ethernet adapter can be a 10 Mbps unit or a 10/100 Mbps unit.

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



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

F.2.1 IP Addresses Explained

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

216.103.126.155

10.1.1.6

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

255.255.255.0

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

216.103.126.0

216.103.126.1

216.103.126.2

etc.

216.103.126.254

216.103.126.255

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

F.2.2 How IP Addresses are Used

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

Some IP address ranges are reserved for use on internal networks, and can be allocated freely as long as no two internal hosts have the same IP address. These internal IP addresses are not routed to the Internet, and any internal hosts using one of these reserved IP addresses cannot communicate on the external Internet without being connected to a host that has a valid Internet IP address. The host would either translate the data, or it would act as a proxy.

Each MiniCore module has its own unique MAC address, which consists of the prefix 0090C2 followed by a code that is unique to each MiniCore module. For example, a MAC address might be 0090C2C002C0.

> **TIP:** You can always obtain the MAC address on your module by running the sample program `DISPLAY_MAC.C` from the `SAMPLES\TCPIP` folder.

F.2.3 Dynamically Assigned Internet Addresses

In many instances, devices on a network do not have fixed IP addresses. This is the case when, for example, you are assigned an IP address dynamically by your dial-up Internet service provider (ISP) or when you have a device that provides your IP addresses using the Dynamic Host Configuration Protocol (DHCP). The MiniCore modules can use such IP addresses to send and receive packets on the Internet, but you must take into account that this IP address may only be valid for the duration of the call or for a period of time, and could be a private IP address that is not directly accessible to others on the Internet. These addresses can be used to perform some Internet tasks such as sending e-mail or browsing the Web, but it is more difficult to participate in conversations that originate elsewhere on the Internet. If you want to find out this dynamically assigned IP address, under Windows NT or later you can run the `ipconfig` command (**Start > Run >cmd**) while you are connected and look at the interface used to connect to the Internet.

Many networks use IP addresses that are assigned using DHCP. When your computer comes up, and periodically after that, it requests its networking information from a DHCP server. The DHCP server may try to give you the same address each time, but a fixed IP address is usually not guaranteed.

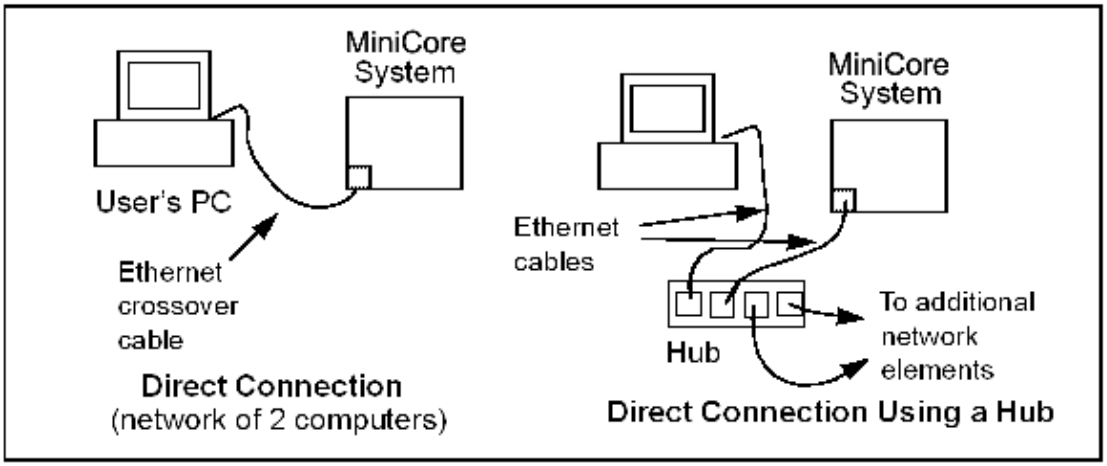
If you are not concerned about accessing the MiniCore from the Internet, you can place the MiniCore on the internal network using an IP address assigned either statically or through DHCP.

F.3 Placing Your Device on the Network

In many corporate settings, users are isolated from the Internet by a firewall and/or a proxy server. These devices attempt to secure the company from unauthorized network traffic, and usually work by disallowing traffic that did not originate from inside the network. If you want users on the Internet to communicate with your MiniCore, you have several options. You can either place the MiniCore directly on the Internet with a real Internet address or place it behind the firewall. If you place the MiniCore behind the firewall, you need to configure the firewall to translate and forward packets from the Internet to the MiniCore.

F.4 Running TCP/IP Sample Programs

We have provided a number of sample programs demonstrating various uses of TCP/IP for networking embedded systems. These programs require you to connect your PC and the MiniCore module together on the same network. This network can be a local private network (preferred for initial experimentation and debugging), or a connection via the Internet.



F.4.1 How to Set IP Addresses in the Sample Programs

With the introduction of Dynamic C 7.30 we have taken steps to make it easier to run many of our sample programs. You will see a `TCPCONFIG` macro. This macro tells Dynamic C to select your configuration from a list of default configurations. You will have three choices when you encounter a sample program with the `TCPCONFIG` macro.

1. You can replace the `TCPCONFIG` macro with individual `MY_IP_ADDRESS`, `MY_NETMASK`, `MY_GATEWAY`, and `MY_NAMESERVER` macros in each program.
2. You can leave `TCPCONFIG` at the usual default of 1, which will set the IP configurations to `10.10.6.100`, the netmask to `255.255.255.0`, and the nameserver and gateway to `10.10.6.1`. If you would like to change the default values, for example, to use an IP address of `10.1.1.2` for the MiniCore module, and `10.1.1.1` for your PC, you can edit the values in the section that directly follows the "General Configuration" comment in the `TCP_CONFIG.LIB` library. You will find this library in the `LIB\Rabbit4000\TCPIP` folder.
3. You can create a `CUSTOM_CONFIG.LIB` library and use a `TCPCONFIG` value greater than 100. Instructions for doing this are at the beginning of the `TCP_CONFIG.LIB` library in the `LIB\Rabbit4000\TCPIP` folder.

There are some other "standard" configurations for `TCPCONFIG` that let you select different features such as DHCP. Their values are documented at the top of the `TCP_CONFIG.LIB` library in the `LIB\Rabbit4000\TCPIP` folder. More information is available in the ***Dynamic C TCP/IP User's Manual***.

F.4.2 How to Set Up your Computer for Direct Connect

Follow these instructions to set up your PC or notebook. Check with your administrator if you are unable to change the settings as described here since you may need administrator privileges. The instructions are specifically for Windows 2000, but the interface is similar for other versions of Windows.

> **TIP:** If you are using a PC that is already on a network, you will disconnect the PC from that network to run these sample programs. Write down the existing settings before changing them to facilitate restoring them when you are finished with the sample programs and reconnect your PC to the network.

1. Go to the control panel (**Start > Settings > Control Panel**), and then double-click the Network icon.
2. Select the network interface card used for the Ethernet interface you intend to use (e.g., **TCP/IP Xircom Credit Card Network Adapter**) and click on the "Properties" button. Depending on which version of Windows your PC is running, you may have to select the "Local Area Connection" first, and then click on the "Properties" button to bring up the Ethernet interface dialog. Then "Configure" your interface card for a "10Base-T Half-Duplex" or an "Auto-Negotiation" connection on the "Advanced" tab.

   > **NOTE:** Your network interface card will likely have a different name.

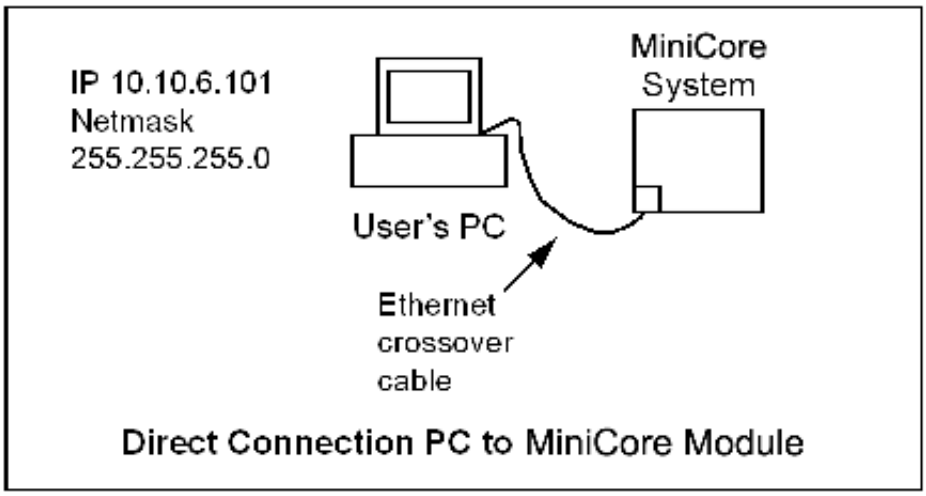
3. Now select the **IP Address** tab, and check **Specify an IP Address**, or select TCP/IP and click on "Properties" to assign an IP address to your computer (this will disable "obtain an IP address automatically"):

   IP Address : 10.10.6.101

   Netmask : 255.255.255.0

   Default gateway : 10.10.6.1

4. Click **<OK>** or **<Close>** to exit the various dialog boxes.



Direct Connection PC to MiniCore Module

F.5 Run the `PINGME.C` Sample Program

Connect the crossover cable from your computer's Ethernet port to the MiniCore module's RJ-45 Ethernet connector. Open this sample program from the `SAMPLES\TCPIP\ICMP` folder, compile the program, and start it running under Dynamic C. The crossover cable is connected from your computer's Ethernet adapter to the MiniCore module's RJ-45 Ethernet connector. When the program starts running, the green **LINK** light on the MiniCore module should be on to indicate an Ethernet connection is made. (Note: If the **LNK** light does not light, you may not be using a cross-over cable, or if you are using a hub with straight-through cables perhaps the power is off on the hub.)

The next step is to ping the module from your PC. This can be done by bringing up the MS-DOS window and running the pingme program:

```
ping 10.10.6.101
```

or by **Start > Run**

and typing the entry

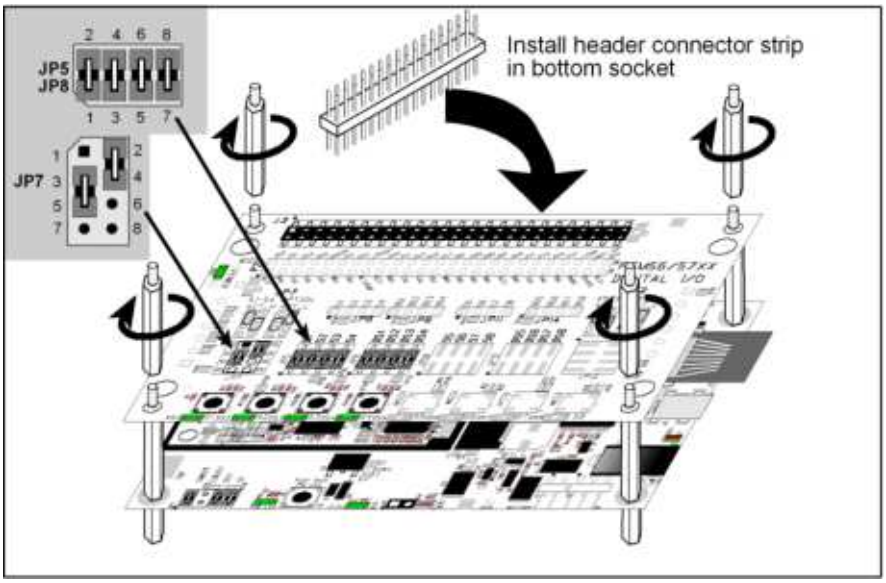
```
ping 10.10.6.101
```

The ping routine will ping the module four times and write a summary message on the screen describing the operation.

F.6 Running Additional Sample Programs With Direct Connect

The sample programs discussed here are in the Dynamic C `SAMPLES\RCM5700\TCPIP\` or `SAMPLES\RCM6700\TCPIP` folder.

- `BROWSELED.C`—The Digital I/O accessory board must be installed with the jumpers set up as shown below to run this sample program.

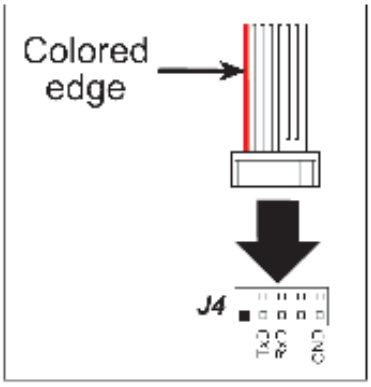


This program demonstrates a basic controller running a Web page. Four "LEDs" are created on the Web page, and four buttons to toggle them. Users can change the status of the lights from the Web browser. The LEDs on the Digital I/O accessory board match the ones on the Web page. As long as you have not modified the `TCPCONFIG 1` macro in the sample program, enter the following server address in your Web browser to bring up the Web page served by the sample program.

http://10.10.6.100

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

- `ETHERNET_TO_SERIAL.C`—Before you compile and run this sample program, use the long 10-pin header to DB9 cable (Part No. 540-0085) to connect header J4 to a PC COM port. Line up the colored edge of the cable with pin 1 on header J4 as shown in the diagram (pin 1 is indicated by a small square on the silkscreen). Note that two of the lines in the cable opposite the colored side are *not* connected within the 10-pin connector.



Open a Hyperterminal session (**Start > Accessories > Communications**). Select the PC COM port the cable is connected to and set the default serial parameters:

Bits per second: 115200

Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

As long as you have not modified the `TCPCONFIG 1` macro in the sample program, enter the following server address in your Web browser to bring up the Web page served by the sample program.

http://10.10.6.100

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

Now compile and run the sample program. With the Hyperterminal client active (click on the Hyperterminal window appearing on your PC desktop), anything you type on your keyboard will be echoed back by the Rabbit and will appear in the Web browser. Note that you will not see what you are typing in the active window unless you have enabled the local echo.

This sample program uses the RabbitWeb HTTP enhancements to configure a simple Ethernet-to-serial converter. The sample program only supports listening TCP sockets, meaning that Ethernet-to-serial devices can only be started by another device initiating the network connection to the Rabbit.

Each serial port can be associated with a specific TCP port. The Rabbit will listen on each of these TCP ports for a connection, which will then be associated with a specific serial port. Data will then be shuttled between the serial and Ethernet connections.

F.7 Where Do I Go From Here?

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

If there are any problems at this point:

- Use the Dynamic C **Help** menu to get further assistance with Dynamic C.
- Check the Rabbit Technical Bulletin Board and forums at www.digi.com/support/forum/.
- Use the Technical Support e-mail form at www.digi.com/support/.

If the sample programs ran fine, you are now ready to go on.

Additional sample programs from the `SAMPLES\TCPIP` folder are described in the *Dynamic C TCP/IP User's Manual.*

Please refer to the *Dynamic C TCP/IP User's Manual* to develop your own applications. *An Introduction to TCP/IP* provides background information on TCP/IP, and is available on the CD and on our Web site.

APPENDIX G. POWER SUPPLY

Appendix G provides information on the current requirements of the RCM5700/RCM6700, and includes some background on the chip select circuit used in power management.

G.1 Power Supplies

The RCM5700/RCM6700 requires a regulated 3.15 V – 3.45 V DC power source. The MiniCore design presumes that the voltage regulator is on the user board, and that the power is made available to the RCM5700 board through the edge connectors.

An RCM5700 with no loading at the outputs operating at 50.0 MHz typically draws 130 mA. An RCM6700 with no loading at the outputs operating at 187.5 MHz typically draws 75 mA.

G.1.1 Battery Backup for the RCM5700/RCM5710 and RCM6700 Family

The RCM5700/RCM6700 modules do not include a battery, but there is provision for customer-supplied battery backup on some module types. Battery backup keeps the Rabbit 5000 real-time clock running on the RCM5700/RCM5710 modules and keeps the Rabbit 6000 real-time clock running while also preserving content of the internal 32 KB SRAM on the RCM6700 family of modules. Battery backup is not available on the RCM5750/5760 modules.

The edge connector, shown in Figure G-1, allows access to the external battery. This makes it possible to connect an external 3 V power supply. This allows the internal Rabbit 5000/6000 real-time clock to retain data with the RCM5700/RCM6700 powered down.

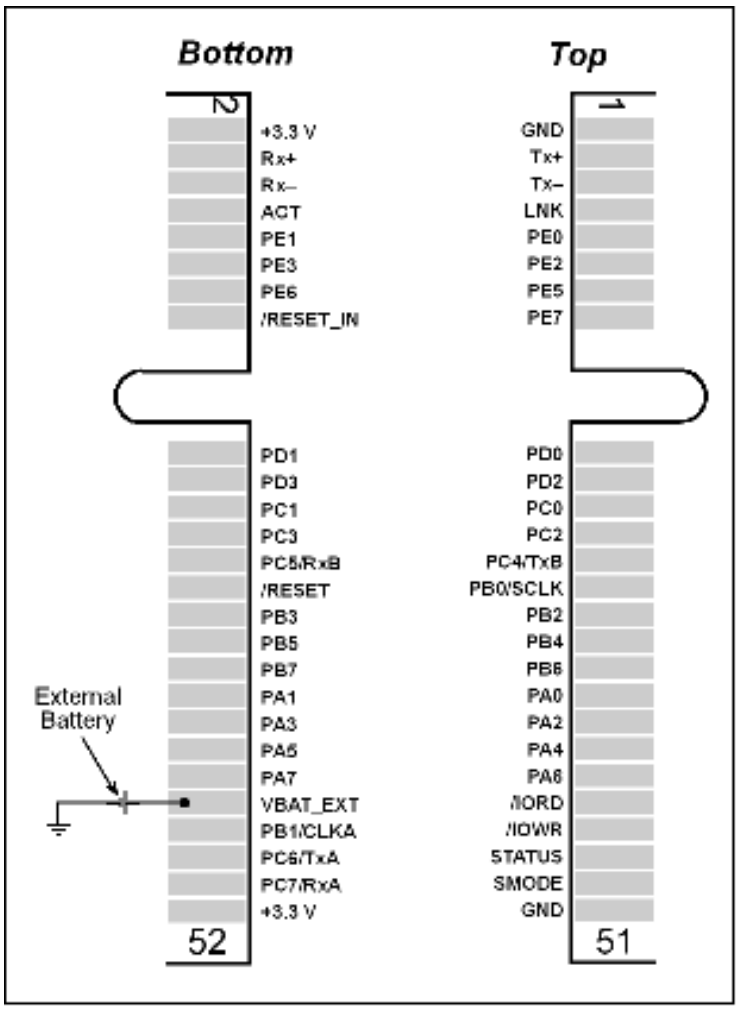


Figure G-1. External Battery Connections

A lithium battery with a nominal voltage of 3 V and a minimum capacity of 165 mA·h is recommended. A lithium battery is strongly recommended because of its nearly constant nominal voltage over most of its life.

The drain on the battery is typically 5uA for the RCM5700 and maximum 50uA for the RCM6700 (R6K processor version marked "CC") when no other power is supplied.

The actual life in your application will depend on the current drawn by components not on the RCM5700/RCM6700 and on the storage capacity of the battery. The RCM5700/RCM6700 does not drain the battery while it is powered up normally.

Cycle the main power off/on on the MiniCore after you install a backup battery for the first time, and whenever you replace the battery. This step will minimize the current drawn by the real-time clock oscillator circuit from the backup battery should the MiniCore experience a loss of main power.

> **NOTE:** Remember to cycle the main power off/on any time the MiniCore is removed from the Interface Board or motherboard since that is where the backup battery would be located.

Rabbit's Technical Note TN235, *External 32.768 kHz Oscillator Circuits*, provides additional information about the current draw by the real-time clock oscillator circuit.

G.1.2 Battery-Backup Circuit

Figure G-2 shows the battery-backup circuit.

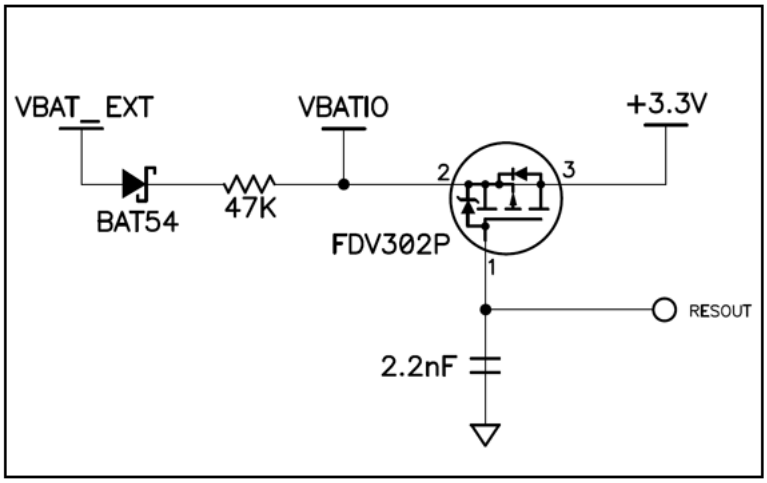


Figure G-2. MiniCore Backup Battery Circuit

The battery-backup circuit serves the following purposes:

- It reduces the battery voltage to real-time clock, thereby limiting the current consumed by the real-time clock and lengthening the battery life.
- It ensures that current can flow only ***out*** of the battery to prevent charging the battery.
- Switches to battery power only when the +3.3 V system power supply is off.

G.1.3 Reset Generator

The RCM5700/RCM6700 uses a reset generator to reset the Rabbit microprocessor when the voltage drops below the voltage necessary for reliable operation. The reset occurs between 2.85 V and 3.00 V, typically 2.93 V. The MiniCore has a reset output on the edge connector. The push-pull type output of the reset generator of the MiniCore module is routed to the /RESET pin of the edge connector through a series current limiting resistor followed by a pull-down resistor. Therefore, depending on the application, the /RESET line on the customer's board may require a buffer circuit to feed external connections properly.

Index

A

accessory boards
Digital I/O 75
configuration options ... 79
LED outputs 79
pushbutton switches 79
dimensions 76
specifications 76
Serial Communication 83
configuration options ... 87
RTS/CTS 87
dimensions 84
specifications 84
additional information
online documentation 10

B

battery backup
battery life 106
circuit 106
reset generator 106

C

clock doubler 38
cloning 43
connectors
design and layout recommendations 52
mini PCI Express 50
PCB footprint with latch connector 52
PCB footprint with standoffs .. 53

D

Development Kits 9
Deluxe Development Kit 9
AC adapter 9
accessory boards 9
cables 9
Standard Development Kit 9
Dynamic C 9
Getting Started instructions 9
Interface Board 9
Prototyping Board 9
USB cable 9
digital I/O 26
function calls 42
memory interface 33
SMODE0 33
SMODE1 33
Digital I/O accessory board .. 74, 75
features 75
dimensions
Digital I/O accessory board 76
Interface Board 60
Prototyping Board 69
RCM5700 45
Serial COmmunication accessory board 84
Dynamic C 10, 11, 17, 40
add-on modules 11, 43
installation 11
Rabbit Embedded Security Pack 43
sample programs 20
standard features
debugging 41
telephone-based technical support 10, 43
troubleshooting 17
upgrades and patches 43

E

Ethernet cables 90
how to tell them apart 90
Ethernet connections 90, 92
10/100Base-T 92
10Base-T Ethernet card 90
additional resources 103
direct connection 92
Ethernet cables 92
Ethernet hub 90
IP addresses 92, 94
MAC addresses 95
steps 90, 91
Ethernet port 36
pinout 36
exclusion zone 47
external I/O bus 33
software 33

F

features
Digital I/O accessory board 75
Interface Board 59
Prototyping Board 68
RCM5700 6
Serial Communication accessory board 83

H

hardware connections 12
install RCM5700 on Interface Board 13
USB cable 15
http
//forums.digi.com/support/forum/index 18

I

install additional boards . 65, 73, 81
Interface Board 58
  dimensions 60
  features 59
  jumper configurations 66
  jumper locations 66
  mounting RCM5700 13
  power supply 63
  power supply jack polarity 58
  specifications 60
IP addresses 94
  how to set in sample programs 99
  how to set PC IP address 100

J

jumper configurations
  accessory boards
    Digital I/O 79
    Serial Communication . 87
  Interface Board 66
  Prototyping Board
    JP2 (analog inputs reference) 66
  RCM5700 55
    JP1 (flash memory size) 55
    JP2 (Tx+) 55
    JP3 (Tx–) 55
    JP4 (Rx–) 55
    JP5 (Rx+) 55
    jumper locations 55

M

MAC addresses 95

P

PCB footprint
  mini PCI Express connector and latch 52
  mini PCI Express connector and standoffs 53
pinout
  Digital I/O accessory board ... 78
  Ethernet port 36
  RCM5700
    alternate configurations 28
  RCM5700 edge connectors 26
  Serial Communication accessory board 86
power supplies
  +3.3 V 104
  battery backup 105
Program Mode 37
  switching modes 37
programming
  Remote Program Update 8
programming port 36
Prototyping Board 68
  dimensions 69
  expansion area 68
  features 68
  prototyping area 72
  specifications 70

R

Rabbit 5000
  tamper detection 39
  VBAT RAM memory 39
Rabbit subsystems 27
RCM5700
  mounting on Interface Board . 13
Run Mode 37
  switching modes 37

S

sample programs 20
  accessory boards
    Digital I/O 21
    Serial Communication . 23
  getting to know the RCM5700
    FLASHLED01.C 20
    FLASHLED01A.C 20
    FLASHLED02.C 20
    FLASHLED02A.C 20
    SERIALTOSERIAL.C 21, 22
    SIMPLE5WIRE.C 23
    SWITCHLEDS.C 21
  hardware setup 19
  how to run TCP/IP sample programs 98, 99
  how to set IP address 99
  onboard serial flash
    SERIAL_FLASHLOG.C 24
    SFLASH_INSPECT.C 24
  PONG.C 17
  TCP/IP
    BROWSELED.C 102
    DISPLAY_MAC.C 95
    ETHERNET_TO_SERIAL.C 102
    PINGME.C 101
  USERBLOCK_CLEAR.C 42
  USERBLOCK_INFO.C 42
serial communication 34
  function calls 42
  software
    PACKET.LIB 42
    RS232.LIB 42
Serial Communication accessory board 83
  features 83
serial flash
  software
    FAT_CONFIG.LIB 43

SFLASH.LIB 43
serial ports 34, 35
Ethernet port 36
programming port 36
Serial Port B
manage conflicts with RCM5750/RCM5760 serial flash 34
Serial Port E
configuration information 34
Serial Port F
configuration information 34
software 10
external I/O bus 42
I/O drivers 42
libraries
TCP/IP 43
sample programs 20
serial communication drivers . 42
serial flash 43
TCP/IP drivers 43
troubleshooting 17
specifications 44
accessory boards
headers 77, 85
Digital I/O accessory board ... 76
dimensions 45
electrical, mechanical, and environmental 48, 50
exclusion zone 47
Interface Board 60
headers 61
Prototyping Board 70
headers 71
Rabbit 5000 DC characteristics 54
Serial Communication accessory board 84
spectrum spreader
settings 38
subsystems
digital inputs and outputs 26
switching modes 37

T

tamper detection 39
TCP/IP primer 92
technical support 18
troubleshooting 17

U

USB cable
connections 15
user block
function calls 42
readUserBlock() 39
writeUserBlock() 39

V

VBAT RAM memory 39