Welcome to E-XFL.COM

Understanding Embedded - Microcontroller, Microprocessor, FPGA Modules

Embedded - Microcontroller, Microprocessor, and FPGA Modules are fundamental components in modern electronic systems, offering a wide range of functionalities and capabilities. Microcontrollers are compact integrated circuits designed to execute specific control tasks within an embedded system. They typically include a processor, memory, and input/output peripherals on a single chip. Microprocessors, on the other hand, are more powerful processing units used in complex computing tasks, often requiring external memory and peripherals. FPGAs (Field Programmable Gate Arrays) are highly flexible devices that can be configured by the user to perform specific logic functions, making them invaluable in applications requiring customization and adaptability.

Applications of Embedded - Microcontroller,

Details	
Product Status	Obsolete
Module/Board Type	MPU Core
Core Processor	Rabbit 5000
Co-Processor	-
Speed	73.73MHz
Flash Size	3MB
RAM Size	1.5MB
Connector Type	IDC Header 2x25, 2x5, 1xAntenna
Size / Dimension	1.84" x 2.85" (47mm x 72mm)
Operating Temperature	-30°C ~ 75°C
Purchase URL	https://www.e-xfl.com/product-detail/digi-international/20-101-1247

2.2.5 Step 5 — Connect Power

Once all the other connections have been made, you can connect power to the Prototyping Board.

If you have the universal AC adapter, prepare the AC adapter for the country where it will be used by selecting the appropriate plug. Snap in the top of the plug assembly into the slot at the top of the AC adapter as shown in Figure 5, then press down on the plug until it clicks into place.

Connect the AC adapter to 3-pin header J1 on the Prototyping Board as shown in Figure 5 above. The connector may be attached either way as long as it is not offset to one side—the center pin of J1 is always connected to the positive terminal, and either edge pin is ground.

Plug in the AC adapter. The **PWR** LED on the Prototyping Board next to the power connector at J1 should light up. The RCM5400W and the Prototyping Board are now ready to be used.

> **NOTE:** A **RESET** button is provided on the Prototyping Board next to the battery holder to allow a hardware reset without disconnecting power.

To power down the Prototyping Board, unplug the power connector from J1. You should disconnect power before making any circuit adjustments in the prototyping area, changing any connections to the board, or removing the RCM5400W from the Prototyping Board.

2.3.1 Troubleshooting

If you receive the message **`Could Not Open Serial Port`**, check that the COM port assigned to the USB programming cable was identified and set up in Dynamic C as described in the preceding section.

If you receive the message **`No Rabbit Processor Detected`**, the programming cable may be connected to the wrong COM port, a connection may be faulty, or the target system may not be powered up. First, check to see that the power LED on the Prototyping Board is lit. If the LED is lit, check both ends of the programming cable to ensure that it is firmly plugged into the PC and the programming header on the RCM5400W with the marked (colored) edge of the programming cable towards pin 1 of the programming header. Ensure that the module is firmly and correctly installed in its connectors on the Prototyping 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.

2.4 Where Do I Go From Here?

If the sample program ran fine, you are now ready to go on to the sample programs in Chapter 3 and to develop your own applications. The sample programs can be easily modified for your own use. The user's manual also provides complete hardware reference information and software function calls for the RCM5400W series of modules and the Prototyping Board.

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

2.4.1 Technical Support

> **NOTE:** If you purchased your RCM5400W or RCM5450W 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:

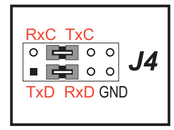
- Use the Dynamic C **Help** menu to get further assistance with Dynamic C.
- Check the Rabbit Technical Bulletin Board and forums at www.rabbit.com/support/bb/ and at www.rabbit.com/forums/.
- Use the Technical Support e-mail form at www.rabbit.com/support/.

3.2.2 Serial Communication

The following sample programs are found in the **SAMPLES\RCM5400W\SERIAL** folder.

- **FLOWCONTROL.C**—This program demonstrates how to configure Serial Port D for CTS/RTS flow control with serial data coming from Serial Port C (TxC) at 115,200 bps. The serial data received are displayed in the **STDIO** window.

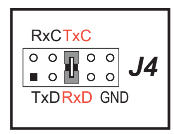
  To set up the Prototyping Board, you will need to tie TxD and RxD together on the RS-232 header at J4, and you will also tie TxC and RxC together using the jumpers supplied in the Development Kit as shown in the diagram.

  A repeating triangular pattern should print out in the **STDIO** window.
  The program will periodically switch flow control on or off to demonstrate the effect of flow control.

  If you have two Prototyping Boards with modules, run this sample program on the sending board, then disconnect the programming cable and reset the sending board so that the module is operating in the Run mode. Connect TxC, TxD, and GND on the sending board to RxC, RxD, and GND on the other board, then, with the programming cable attached to the other module, run the sample program.

- **PARITY.C**—This program demonstrates the use of parity modes by repeatedly sending byte values 0–127 from Serial Port C to Serial Port D. The program will switch between generating parity or not on Serial Port C. Serial Port D will always be checking parity, so parity errors should occur during every other sequence.

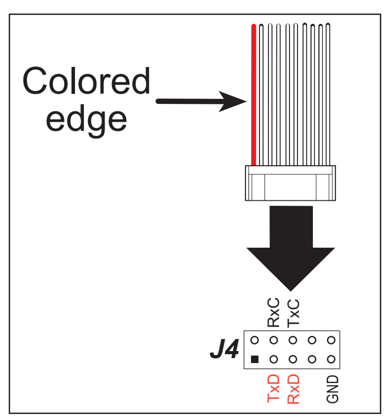
  

  To set up the Prototyping Board, you will need to tie TxC and RxD together on the RS-232 header at J4 using one of the jumpers supplied in the Development Kit as shown in the diagram.

  The Dynamic C **STDIO** window will display the error sequence.

- **SERDMA.C**—This program demonstrates using DMA to transfer data from a circular buffer to the serial port and vice versa. The Dynamic C **STDIO** window is used to view or clear the buffer.

  Before you compile and run the sample program, you will need to connect the RS-232 header at J4 to your PC as shown in the diagram using the serial to DB9 cable supplied in the Development Kit.

  Once you have compiled and run the sample program, start Tera Term or another terminal emulation program to connect to the selected PC serial port at a baud rate of 115,200 bps. You can observe the output in the Dynamic C **STDIO** window as you type in Tera Term, and you can also use the Dynamic C **STDIO** window to clear the buffer.

  The Tera Term utility can be downloaded from hp.vector.co.jp/authors/VA002416/teraterm.html.

4. Hardware Reference

Chapter 4 describes the hardware components and principal hardware subsystems of the RCM5400W. Appendix A, "RCM5400W Specifications," provides complete physical and electrical specifications.

Figure 6 shows the Rabbit-based subsystems designed into the RCM5400W.

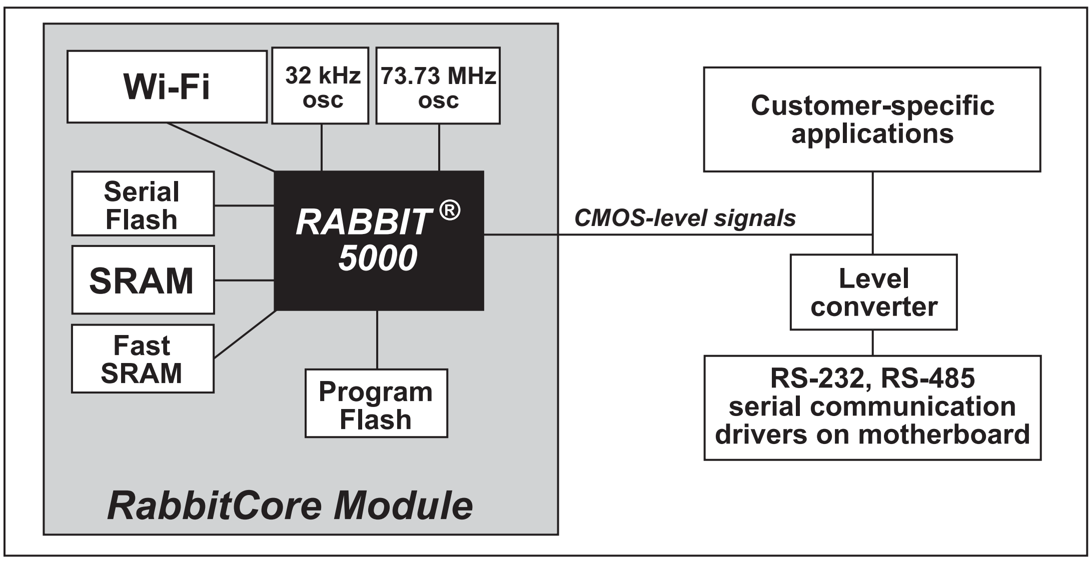


Figure 6. RCM5400W Subsystems

The 73.73 MHz frequency shown for the RCM5400W is generated using a 36.864 MHz crystal with the Rabbit 5000 clock doubler enabled.

4.1 RCM5400W Digital Inputs and Outputs

Figure 7 shows the RCM5400W pinouts for header J1.

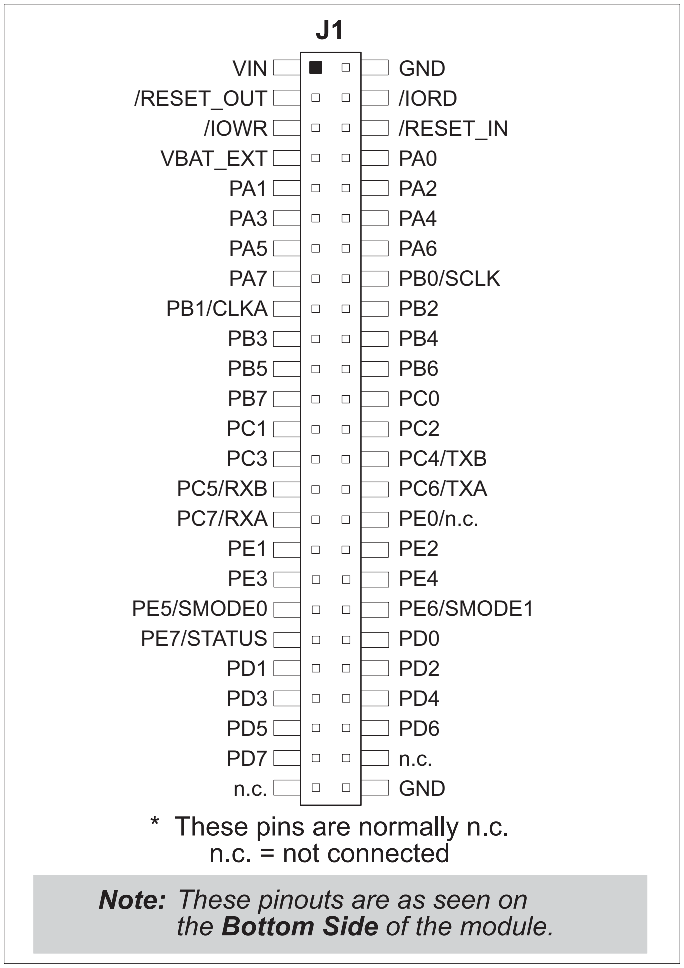


Figure 7. RCM5400W Pinout

Headers J1 is a standard 2 × 25 IDC header with a nominal 1.27 mm pitch.

Figure 8 shows the use of the Rabbit 5000 microprocessor ports in the RCM5400W modules.

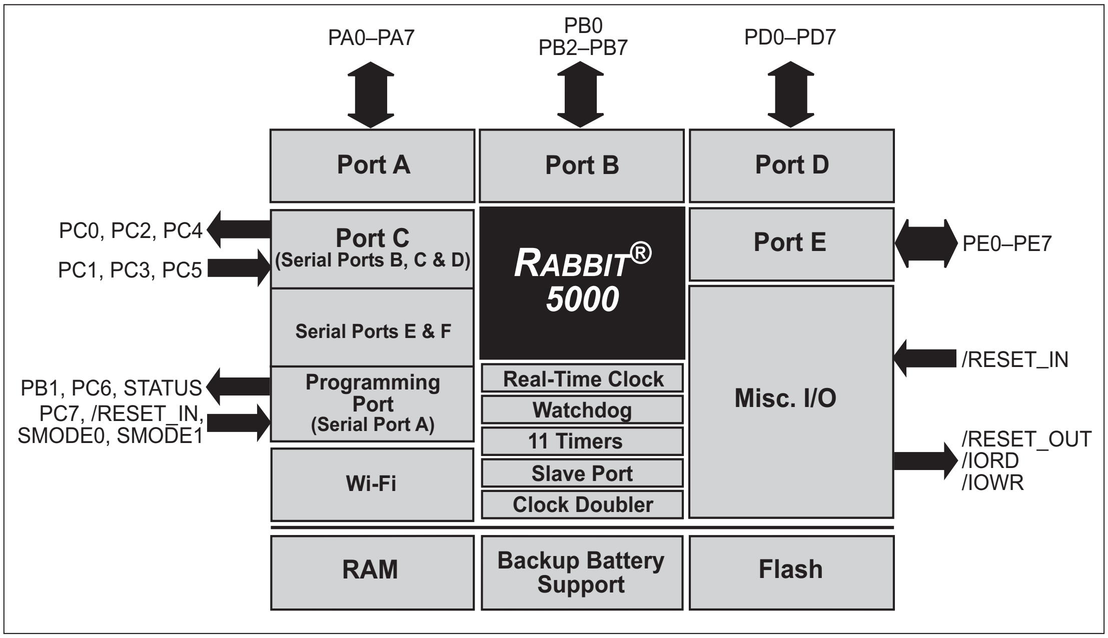


Figure 8. Use of Rabbit 5000 Ports

The ports on the Rabbit 5000 microprocessor used in the RCM5400W are configurable, and so the factory defaults can be reconfigured. Table 2 lists the Rabbit 5000 factory defaults and the alternate configurations.

Table 2. RCM5400W Pinout Configurations (continued)

Pin		Pin Name	Default Use	Alternate Use	Notes
Header J1	40	PD0	Input/Output	I/O Strobe I0 Timer C0 D8 INT0 SCLKD/TCLKF QRD1B	
	41	PD1	Input/Output	IA6 I/O Strobe I1 Timer C1 D9 INT1 RXD/RCLKF QRD1A Input Capture	
	42	PD2	Input/Output	I/O Strobe I2 Timer C2 D10 DREQ0 TXF/SCLKC QRD2B	Serial Port F
	43	PD3	Input/Output	IA7 I/O Strobe I3 Timer C3 D11 DREQ1 RXC/RXF QRD2A Input Capture	
	44	PD4	Input/Output	I/O Strobe I4 D12 PWM0 TXB/TCLKE	
	45	PD5	Input/Output	IA6 I/O Strobe I5 D13 PWM1 RXB/RCLKE Input Capture	

Table 4. Wi-Fi Channel Allocations

Channel	Center Frequency (GHz)	Frequency Spread (GHz)
1	2.412	2.401–2.423
2	2.417	2.406–2.428
3	2.422	2.411–2.433
4	2.427	2.416–2.438
5	2.432	2.421–2.443
6	2.437	2.426–2.448
7	2.442	2.431–2.453
8	2.447	2.436–2.458
9	2.452	2.441–2.463
10	2.457	2.446–2.468
11	2.462	2.451–2.473
12*	2.467	2.456–2.478
13*	2.472	2.461–2.483
14 (not used)	2.484	2.473–2.495

* These channels are disabled for units delivered for sale in the United States and Canada.

Many countries specify the channel range and power limits for Wi-Fi devices operated within their borders, and these limits are set automatically in the RCM5400W in firmware according to the country or region. For example, only channels 1–11 are authorized for use in the United States or Canada, and so channels 12 and 13 are disabled. See Section 6.2.4.1 for additional information and sample programs demonstrating how to configure an end device to meet the regulatory channel range and power limit requirements. Table 5 provides additional information on which channels are allowed in selected countries. *Any attempt to operate a device outside the allowed channel range or power limits will void your regulatory approval to operate the device in that country.*

The following regions have macros and region numbers defined for convenience.

Table 5. Worldwide Wi-Fi Macros and Region Numbers

Region	Macro	Region Number	Channel Range
Americas	`IFPARAM_WIFI_REGION_AMERICAS`	0	1–11
Mexico	`IFPARAM_WIFI_REGION__MEXICO_INDOORS`	1	1–11 (indoors)
	`IFPARAM_WIFI_REGION_MEXICO_OUTDOORS`	2	9–11 (outdoors)
Canada	`IFPARAM_WIFI_REGION_CANADA`	3	1–11
Europe, Middle East, Africa, except France	`IFPARAM_WIFI_REGION_EMEA`	4	1–13
France	`IFPARAM_WIFI_REGION_FRANCE`	5	10–13
Israel	`IFPARAM_WIFI_REGION_ISRAEL`	6	3–11
China	`IFPARAM_WIFI_REGION_CHINA`	7	1–11
Japan	`IFPARAM_WIFI_REGION_JAPAN`	8	1–14*
Australia	`IFPARAM_WIFI_REGION_AUSTRALIA`	9	1–11

* Channel 14 is not available for the RCM4400W.

The same omnidirectional antenna is used to transmit and receive the 802.11b/g RF signal. An antenna switch isolates the high-power RF Tx signal path from the RF Rx signal path. The antenna switch works by alternately connecting the antennas to either the AL2236 Tx output or to the AL2236 Rx input. In order to support this antenna-sharing scheme, the RCM5400W module operates the radio in a half-duplex mode so that receive and transmit operations never occur at the same time The antenna switch at U19 switches the receive/transmit functionality between the outputs at P2 and P1 (not stuffed) so that P2 is transmitting while P1 would be receiving and vice versa. Dynamic C does not support a P1 output.

The RF connector on P2 is an RP-SMA connector with its outer casing attached to the RCM5400W ground. It is recommended that the OEM integrator of this device improve ESD protection by attaching P2 to chassis ground.

There are two LEDs close to the RP-SMA antenna connector atP2, a green LED at DS2 (**LINK**) to indicate association with the Wi-Fi access point, and a yellow LED at DS1 (**ACT**) to indicate activity.

4.5 Other Hardware

4.5.1 Clock Doubler

The RCM5400W 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 73.73 MHz frequency specified for the RCM5400W is generated using a 36.864 MHz crystal.

The clock doubler should not be disabled since Wi-Fi operations depend highly on CPU resources.

4.5.2 Spectrum Spreader

The Rabbit 5000 features a spectrum spreader, which helps to mitigate EMI problems. The spectrum spreader is on by default, but it may also be turned off or set to a stronger setting. The spectrum spreader settings may be changed 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 needed in a real application.

3. Click **OK** to save the macro. The spectrum spreader will now remain off whenever you are in the project file where you defined the macro.

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

4.6 Memory

4.6.1 SRAM

All RCM5400W modules have 512K of battery-backed data SRAM installed at U3, and 512K or 1 MB of fast SRAM are installed at U2 and at U11.

4.6.2 Flash Memory

All RCM5400W modules also have 512K or 1MB of flash memory installed at U4 or U12.

> **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.

Writing to arbitrary flash memory addresses at run time is strongly discouraged. Instead, define a “user block” area to store persistent data. The functions **`writeUserBlock()`** and **`readUserBlock()`** are provided for this. Refer to the ***Dynamic C Function Reference Manual*** for additional information.

4.6.3 Serial Flash

Up to 2MB of serial flash memory is available to store data and Web pages. Sample programs in the **`SAMPLES\RCM5400W\Serial_Flash`** folder illustrate the use of the serial flash memory.

5.3 Upgrading Dynamic C

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

> **NOTE:** The RCM5400W was originally released with Dynamic C version 10.40. We have since determined that the BIOS startup sequence used in Dynamic C versions 10.40 and 10.46 leads to a timing violation on the fast SRAM, which stores the application developed to run on the RCM5400W and RCM5450W. The timing violation occurs when the Rabbit 5000 clock doubler is enabled before its early output enable timing is enabled. Under certain conditions, this prevents the application from loading correctly, and may cause it to crash completely. Therefore, the RCM5400W and RCM5450W require Dynamic C v. 10.50 or later to work properly.

If you developed an application using Dynamic C versions 10.40 or 10.46, the application may be used with Dynamic C v. 10.50 after it is recompiled.

5.3.1 Add-On Modules

Dynamic C version 10.50, which is included with the RCM5400W Development Kit, 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.

Visit our Web site at www.rabbit.com for further information and complete documentation.

- **PINGLED_WPA2_CCMP.C**—This sample program is an extension of **PINGLED.C**. It demonstrates the use of WPA2 PSK (Wi-Fi Protected Access with Pre-Shared Key).). WPA is a more secure replacement for WEP. The implementation in the sample program uses the Advanced Encryption Standard (AES) based algorithm, also known as the CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol) cypher suite.

  Apart from the configuration of **WPA2_CCMP** at the top of the sample program, the rest of the code is identical to the case without WPA2 PSK. Indeed, most of the TCP/IP sample programs should work with WPA2 CCMP simply by using the same configuration settings.

  Configure your access point for WPA2 PSK before you run this sample program. Specify the CCMP cypher suite, and enter a suitable pre-shared key. The key may be entered either as 64 hexadecimal digits or as an ASCII string of up to 63 characters.

  TIP: There is a good chance of typos since the key is long. First, enter the key in this sample program macro, then copy and paste it to your access point. This ensures that both the RCM5400W and the access point have the same key.

  TIP: For an initial test, it may be easier to use the 64 hex digit form of the key rather than the ASCII passphrase. A passphrase requires considerable computation effort, which delays the startup of the sample program by about 30 seconds.

  Now change **PING_WHO** to the address of the host you want to ping.

  You may modify the **PING_DELAY** define to change the amount of time (in milliseconds) between the outgoing pings.

  Uncomment the VERBOSE define to see the incoming ping replies.

  Finally, compile and run this sample program. LED DS2 will flash when a ping is sent. LED DS3 will flash when a ping is received.

- **POWERDOWN.C**—This program demonstrates how to power down the radio transmitter (U18) to reduce power consumption. Note that powering down the Wi-Fi portion of the RCM5400W module results in a loss of the network interface (unlike an Ethernet connection), and so is only suitable for applications such as data logging where only intermittent network connectivity is required.

  The sample program demonstrates the powerdown operation as a simple sequential state machine. LED DS2 on the Prototyping Board will be on when the network interface is up, and LED DS3 will be on when the Wi-Fi circuit is powered up.

  Before you compile and run this sample program, modify the configuration macros, including the **DOWNTIME** and the **UPTIME** values. The interface will be powered up and down for these intervals.

  Now set up a continuous ping on another host, and observe the pings time out successively, then succeed, depending on the LED state.

- **SMTP.C**—This program demonstrates using the SMTP library to send an e-mail when the S2 and S3 switches on the Prototyping Board are pressed. LEDs DS2 and DS3 on the Prototyping Board will light up when e-mail is being sent.

Table A-1 lists the electrical, mechanical, and environmental specifications for the RCM5400W.

Table A-1. RCM5400W Specifications

Parameter	RCM5400W	RCM5450W
Microprocessor	Rabbit® 5000 at 73.73 MHz	
Data SRAM	512K	512K
Program Execution Fast SRAM	512K	1MB
Flash Memory	512K	1MB
Serial Flash Memory	1MB	2MB
Backup Battery	Connection for user-supplied backup battery (to support RTC and data SRAM)	
General Purpose I/O	up to 40 parallel digital I/0 lines configurable with four layers of alternate functions	
Additional Inputs	Reset in	
Additional Outputs	Reset out	
External I/O Bus	Can be configured for 8 data lines and 6 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 asynchronous clocked serial port shared with programming port • 1 clocked serial port shared with serial flash	
Serial Rate	Maximum asynchronous baud rate = CLK/8	
Slave Interface	Slave port allows the RCM5400W 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 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	
Quadrature Decoder	2-channel quadrature decoder accepts inputs from external incremental encoder modules	

Table A-6 lists the delays in gross memory access time for several values of VDD_{IO}.

Table A-6. Preliminary Data and Clock Delays

VDD_{IO} (V)	Clock to Address Output Delay (ns)			Data Setup Time Delay (ns)	Worst-Case Spectrum Spreader Delay (ns)		
	30 pF	60 pF	90 pF		0.5 ns setting no dbl / dbl	1 ns setting no dbl / dbl	2 ns setting no dbl / dbl
3.3	6	8	11	1	2.3 / 2.3	3 / 4.5	4.5 / 9
1.8	18	24	33	3	7 / 6.5	8 / 12	11 / 22

The measurements are taken at the 50% points under the following conditions.

- T = -20°C to 85°C, V = VDD_{IO} ±10%
- Internal clock to nonloaded CLK pin delay ≤ 1 ns @ 85°C/3.0 V

The clock to address output delays are similar, and apply to the following delays.

- T_{adr}, the clock to address delay
- T_{CSx}, the clock to memory chip select delay
- T_{IOCSx}, the clock to I/O chip select delay
- T_{IORD}, the clock to I/O read strobe delay
- T_{IOWR}, the clock to I/O write strobe delay
- T_{BUFEN}, the clock to I/O buffer enable delay

The data setup time delays are similar for both T_{setup} and T_{hold}.

When the spectrum spreader is enabled with the clock doubler, every other clock cycle is shortened (sometimes lengthened) by a maximum amount given in the table above. The shortening takes place by shortening the high part of the clock. If the doubler is not enabled, then every clock is shortened during the low part of the clock period. The maximum shortening for a pair of clocks combined is shown in the table.

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

- **Analog Inputs Header**—The analog signals from a RabbitCore module are presented at header J3 on the Prototyping Board. These analog signals are connected via attenuator/filter circuits on the Prototyping Board to the corresponding analog inputs on the RabbitCore module.

  NOTE: No analog signals are available on the Prototyping Board with the RCM5400W RabbitCore module installed since no analog signals are available on the RCM5400W's header J1.

- **RS-232**—Two 3-wire or one 5-wire RS-232 serial ports are available on the Prototyping Board at header J4. A 10-pin 0.1" pitch header strip installed at J4 allows you to connect a ribbon cable that leads to a standard DE-9 serial connector.
- **Current Measurement Option**—You may cut the trace below header JP1 on the bottom side of the Prototyping Board and install a 1 × 2 header strip from the Development Kit to allow you to use an ammeter across the pins to measure the current drawn from the +5 V supply. Similarly, you may cut the trace below header JP2 on the bottom side of the Prototyping Board and install a 1 × 2 header strip from the Development Kit to allow you to use an ammeter across the pins to measure the current drawn from the +3.3 V supply.
- **Backup Battery**—A 2032 lithium-ion battery rated at 3.0 V, 220 mA·h, provides battery backup for the RCM5400W data SRAM and real-time clock.

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

Table B-1. Prototyping Board Specifications

Parameter	Specification
Board Size	3.80" × 3.80" × 0.48" (97 mm × 97 mm × 12 mm)
Operating Temperature	0°C to +70°C
Humidity	5% to 95%, noncondensing
Input Voltage	8 V to 24 V DC
Maximum Current Draw (including user-added circuits)	800 mA max. for +3.3 V supply, 1 A total +3.3 V and +5 V combined
Prototyping Area	1.3" × 2.0" (33 mm × 50 mm) throughhole, 0.1" spacing, additional space for SMT components
Connectors	One 2 × 25 header socket, 1.27 mm pitch, to accept RCM5400W One 1 × 3 IDC header for power-supply connection One 2 × 5 IDC RS-232 header, 0.1" pitch Two unstuffed header locations for analog and RCM5400W signals 25 unstuffed 2-pin header locations for optional configurations

B.3 Power Supply

The RCM5400W requires a regulated 3.0 V – 3.6 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 Prototyping Board has an onboard +5 V switching power regulator from which a +3.3 V linear regulator draws its supply. Thus both +5 V and +3.3 V are available on the Prototyping Board.

The Prototyping Board itself is protected against reverse polarity by a Shottky diode at D2 as shown in Figure B-3.

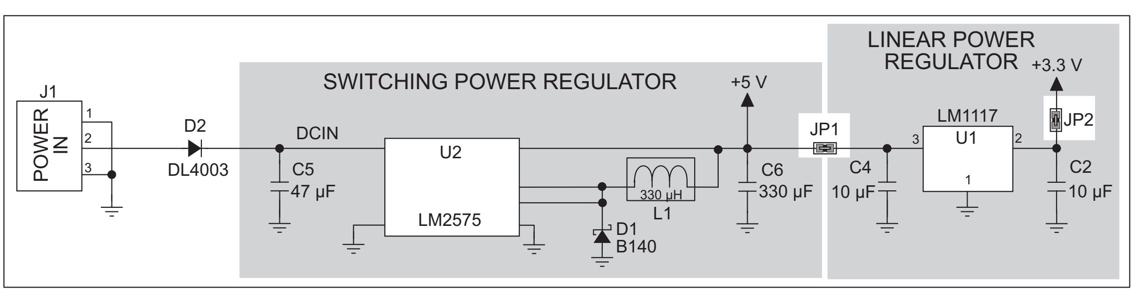


Figure B-3. Prototyping Board Power Supply

TIP: When you lay out your own power-supply circuit, place the switching voltage regulator as far away from the RCM5400W as possible to minimize RF noise, and use low-noise components such as a toroid coil.

INDEX