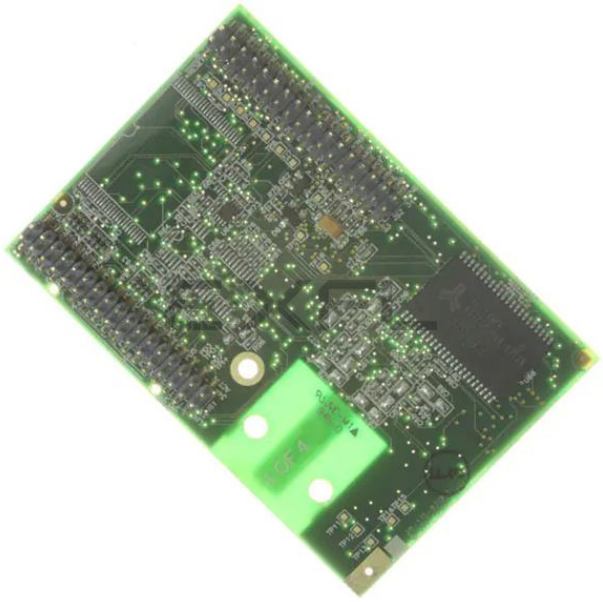
Digi - 20-101-1217 Datasheet



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 3000
Co-Processor	-
Speed	44.2MHz
Flash Size	512KB
RAM Size	768KB
Connector Type	2 IDC Headers 2x17
Size / Dimension	1.85" x 2.73" (47mm x 69mm)
Operating Temperature	-40°C ~ 85°C
Purchase URL	https://www.e-xfl.com/product-detail/digi-international/20-101-1217

1.4 Development and Evaluation Tools

1.4.1 RCM3200 Development Kit

The RCM3200 Development Kit contains the hardware you need to use your RCM3209 or RCM3229 module.

- RCM3209 module.
- Prototyping Board.
- Universal AC adapter, 12 V DC, 1 A (includes Canada/Japan/U.S., Australia/N.Z., U.K., and European style plugs).
- USB programming cable with 10-pin header.
- ***Dynamic C*** CD-ROM, with complete product documentation on disk.
- ***Getting Started*** instructions.
- Accessory parts for use on the Prototyping Board.
- Screwdriver and Cat. 5 Ethernet cables.
- ***Rabbit 3000 Processor Easy Reference*** poster.
- Registration card.

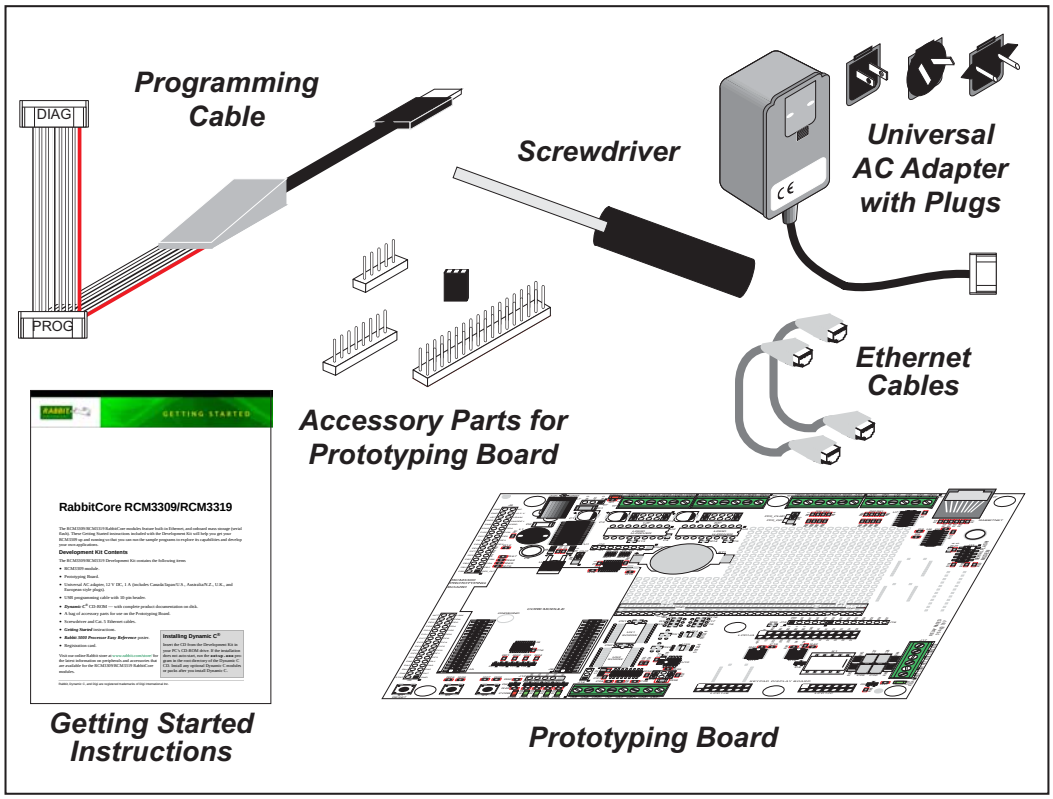


Figure 1. RCM3200 Development Kit

Figure 7 shows the use of the Rabbit 3000 microprocessor ports in the RCM3209/ RCM3229 modules.

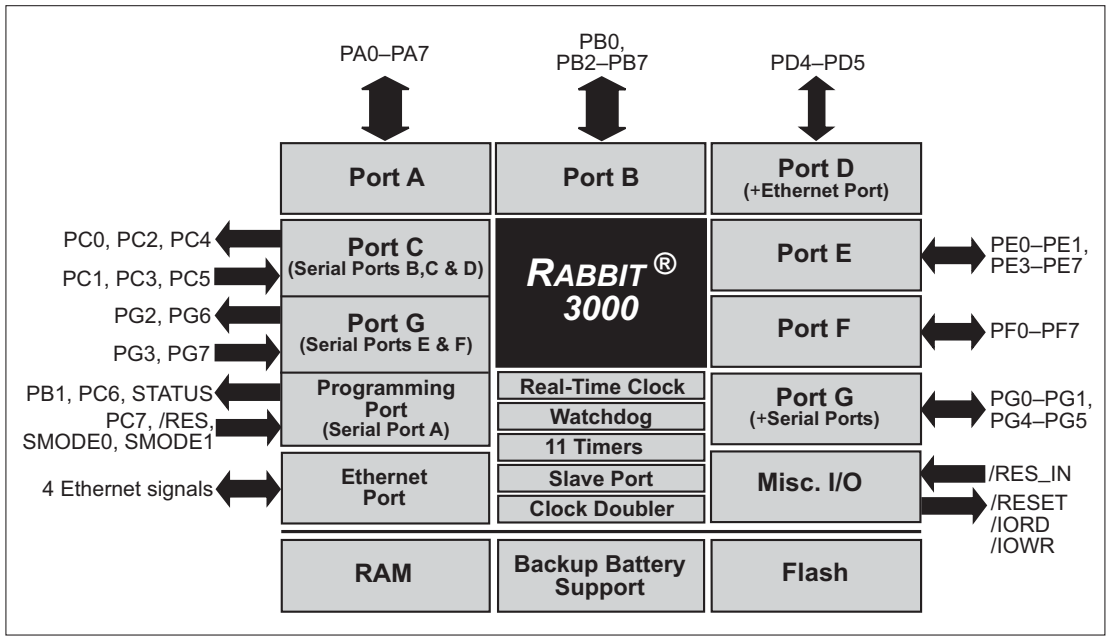


Figure 7. Use of Rabbit 3000 Ports

The ports on the Rabbit 3000 microprocessor used in the RCM3209/RCM3229 are configurable, and so the factory defaults can be reconfigured. Table 2 lists the Rabbit 3000 factory defaults and the alternate configurations.

Table 2. RCM3209/RCM3229 Pinout Configurations (continued)

Pin		Pin Name	Default Use	Alternate Use	Notes
Header J62	20	PG7	Input/Output	RXE	Serial Port E
	21	PG6	Input/Output	TXE	
	22	PG5	Input/Output	RCLKE	Serial Clock E input
	23	PG4	Input/Output	TCLKE	Serial Clock E ouput
	24	/IOWR	Output		External write strobe
	25	/IORD	Input		External read strobe
	26–27	SMODE0, SMODE1	(0,0)—start executing at address zero (0,1)—cold boot from slave port (1,0)—cold boot from clocked Serial Port A SMODE0 =1, SMODE1 = 1 Cold boot from asynchronous Serial Port A at 2400 bps (programming cable connected)		Also connected to programming cable
	28	/RESET_IN	Input		Input to Reset Generator
	29	VRAM	Output		See **Notes** below table
	30	VBAT_EXT	3 V battery Input		Minimum battery voltage 2.85 V
	31	+3.3V	Input		3.15–3.45 V DC
	32	GND			
	33	n.c.			Reserved for future use
	34	GND			

Notes

1. When using pins 33–34 on header J61 to drive LEDs, you must use an external buffer to drive these external LEDs. These pins are not connected on the RCM3229, which does not have the LEDs installed.
2. The VRAM voltage is temperature-dependent. If the VRAM voltage drops below about 1.2 V to 1.5 V, the contents of the battery-backed SRAM may be lost. If VRAM drops below 1.0 V, the 32 kHz oscillator could stop running. Pay careful attention to this voltage if you draw any current from this pin.

6.2.2 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 IP address on the local network to which the packet must be sent. A table is maintained to allow the protocol driver to determine the MAC address corresponding to a particular IP address. If the table is empty, the MAC address is determined by sending an Ethernet broadcast packet to all devices on the local network asking the device with the desired IP address to answer with its MAC address. In this way, the table entry can be filled in. If no device answers, then the device is nonexistent or inoperative, and the packet cannot be sent.

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

Each RCM3209 RabbitCore module has its own unique MAC address, which consists of the prefix 0090C2 followed by the code that appears on the label affixed to the RCM3209 module. For example, a MAC address might be 0090C2C002C0.

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

Table A-1. RabbitCore RCM3209/RCM3229 Specifications (continued)

Feature	RCM3209	RCM3229
Power	3.15 V to 3.45 V DC 325 mA @ 3.3 V	3.15 V to 3.45 V DC 190 mA @ 3.3 V
Operating Temperature	–40°C to +85°C	
Humidity	5% to 95%, noncondensing	
Connectors	Two 2 × 17, 2 mm pitch	
Board Size	1.850" × 2.725" × 0.86" (47 mm × 69 mm × 22 mm)	

A.1.1 Headers

The RCM3209/RCM3229 uses headers at J61 and J62 for physical connection to other boards. J61 and J62 are 2 × 17 SMT headers with a 2 mm pin spacing. J1, the programming port, is a 2 × 5 header with a 1.27 mm pin spacing.

Figure A-3 shows the layout of another board for the RCM3209/RCM3229 to be plugged into. These values are relative to the mounting hole.

A.1.2 Physical Mounting

A 9/32” (7 mm) standoff with a 2-56 screw is recommended to attach the RCM3209/RCM3229 to a user board at the hole position shown in Figure A-3. Either use plastic hardware, or use insulating washers to keep any metal hardware from shorting out signals on the RCM3209/RCM3229.

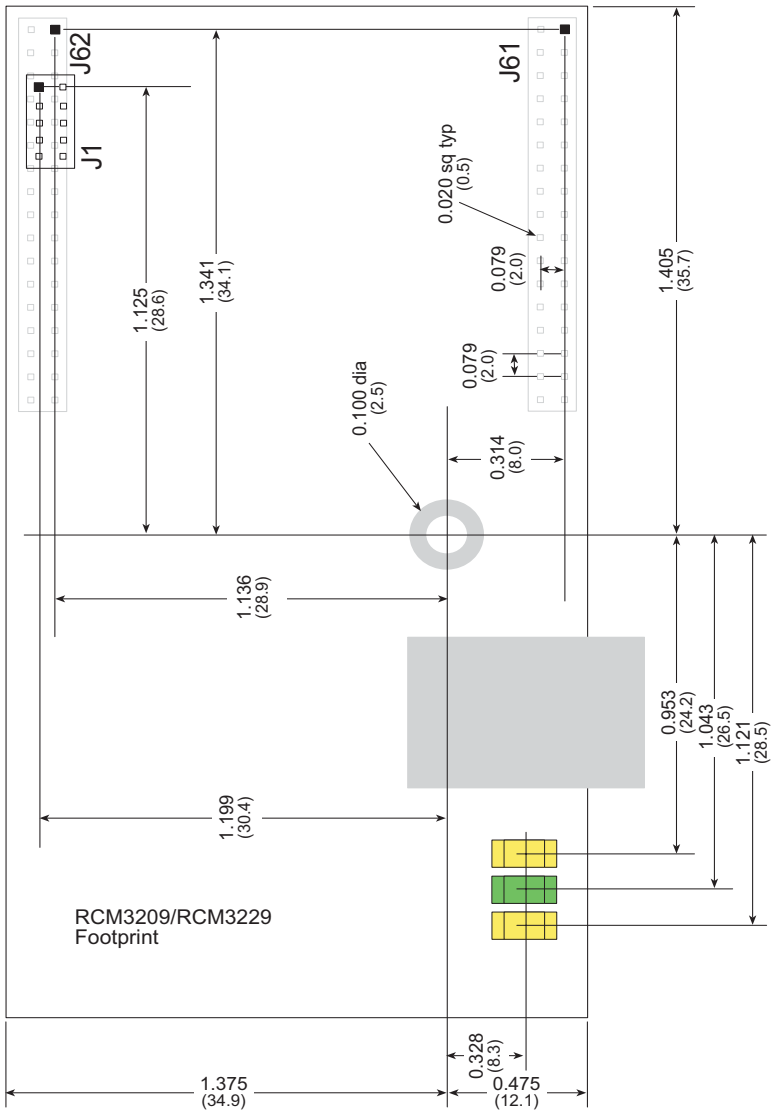


Figure A-3. User Board Footprint for RCM3209/RCM3229

Table A-4 lists the delays in gross memory access time for V_{DD} = 3.3 V.

Table A-4. Data and Clock Delays V_{DD} ±10%, Temp, -40°C–+85°C (maximum)

VDD	Clock to Address Output Delay (ns)			Data Setup Time Delay (ns)	Spectrum Spreader Delay (ns)	
	30 pF	60 pF	90 pF		Normal dbl/no dbl	Strong dbl/no dbl
3.3	6	8	11	1	3/4.5	4.5/9

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

- T = -40°C to 85°C, V = V_{DD} ±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.

Technical Note TN227, ***Interfacing External I/O with Rabbit 2000/3000 Designs***, contains suggestions for interfacing I/O devices to the Rabbit 3000 microprocessors.

A.4 I/O Buffer Sourcing and Sinking Limit

Unless otherwise specified, the Rabbit I/O buffers are capable of sourcing and sinking 6.8 mA of current per pin at full AC switching speed. Full AC switching assumes a 29.4 MHz CPU clock and capacitive loading on address and data lines of less than 70 pF per pin. The absolute maximum operating voltage on all I/O is 5.5 V.

Table A-6 shows the AC and DC output drive limits of the parallel I/O buffers when the Rabbit 3000 is used in the RCM3209/RCM3229.

Table A-6. I/O Buffer Sourcing and Sinking Capability

Pin Name	Output Drive (Full AC Switching) Sourcing/Sinking Limits (mA)	
	Sourcing	Sinking
All data, address, and I/O lines with clock doubler enabled	6.8	6.8

Under certain conditions, you can exceed the limits outlined in Table A-7. See the ***Rabbit 3000 Microprocessor User's Manual*** for additional information.

B.4 Using the Prototyping Board

The Prototyping Board is actually both a demonstration board and a prototyping board. As a demonstration board, it can be used to demonstrate the functionality of the RCM3209/RCM3229 right out of the box without any modifications to either board. There are no jumpers or dip switches to configure or misconfigure on the Prototyping Board so that the initial setup is very straightforward.

The Prototyping Board comes with the basic components necessary to demonstrate the operation of the RCM3209/RCM3229. Two LEDs (DS1 and DS2) are connected to PG6 and PG7, and two switches (S2 and S3) are connected to PG1 and PG0 to demonstrate the interface to the Rabbit 3000 microprocessor. Reset switch S1 is the hardware reset for the RCM3209/RCM3229.

The Prototyping Board provides the user with RCM3209/RCM3229 connection points brought out conveniently to labeled points at headers J2 and J4 on the Prototyping Board. Small to medium circuits can be prototyped using point-to-point wiring with 20 to 30 AWG wire between the prototyping area and the holes at locations J2 and J4. The holes are spaced at 0.1" (2.5 mm), and 40-pin headers or sockets may be installed at J2 and J4. The pinouts for locations J2 and J4, which correspond to headers J1 and J2, are shown in Figure B-4.

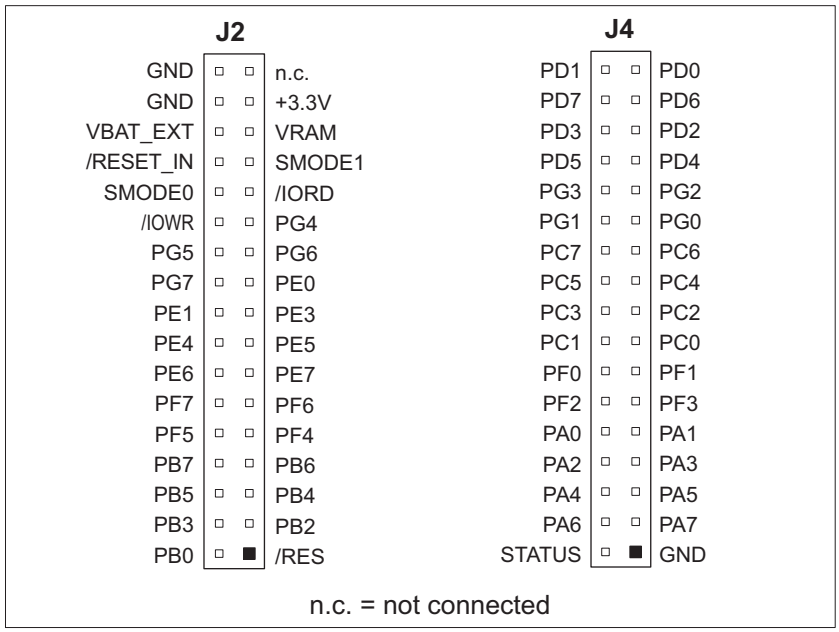


Figure B-4. Prototyping Board Pinout (Top View)

The small holes are also provided for surface-mounted components that may be installed around the prototyping area.

There is a 2.0" × 3.5" through-hole prototyping space available on the Prototyping Board. +3.3 V, +5 V, and GND traces run along the edge of the Prototyping Board for easy access.

B.5 Use of Rabbit 3000 Parallel Ports

Table B-2 lists the Rabbit 3000 parallel ports and their use for the RCM30/31/32XX Prototyping Board.

Table B-2. RCM30/31/32XX Prototyping Board Use of Rabbit 3000 Parallel Ports

Port	I/O	Use		Initial State
PA0–PA7	Output	Configurable external I/O bus		High when not driven by I/O bus
PB0–PB1	Input	Not used		Pulled up on RCM3209/ RCM3229
PB2–PB5	Input	Configurable external I/O bus		High when not driven by I/O bus
PB6–PB7	Output	Not used		Pulled up on RCM3209/ RCM3229
PC0	Output	Not used		High (disabled)
PC1	Input	Not used		Pulled up on RCM3209/ RCM3229
PC2	Output	TXC	Serial Port C	High (disabled)
PC3	Input	RXC		Pulled up on RCM3209/ RCM3229
PC4	Output	TXB	Serial Port B	High (disabled)
PC5	Input	RXB		Pulled up on RCM3209/ RCM3229
PC6	Output	TXA Programming Port	Serial Port A	High (disabled)
PC7	Input	RXA Programming Port		Pulled up on RCM3209/ RCM3229
PD0	Output	Ethernet RSTDRV		High
PD1	Input	Not used		Pulled up on RCM3209/ RCM3229
PD2–PD4	Output	Not used		High
PD5	Input	Not used		Pulled up on Prototyping Board
PD6–PD7	Output	Not used		High
PE0–PE1	Output	Not used		High
PE2	Output	Ethernet chip select		High
PE3	Output	LCD device select		Low (disabled)
PE4	Output	IrDA speed select		Low (disabled)

C.5 Mounting LCD/Keypad Module on the Prototyping Board

Install the LCD/keypad module on header sockets J7, J8, and J10 of the Prototyping Board as shown in Figure C-7. Be careful to align the pins over the headers, and do not bend them as you press down to mate the LCD/keypad module with the Prototyping Board.

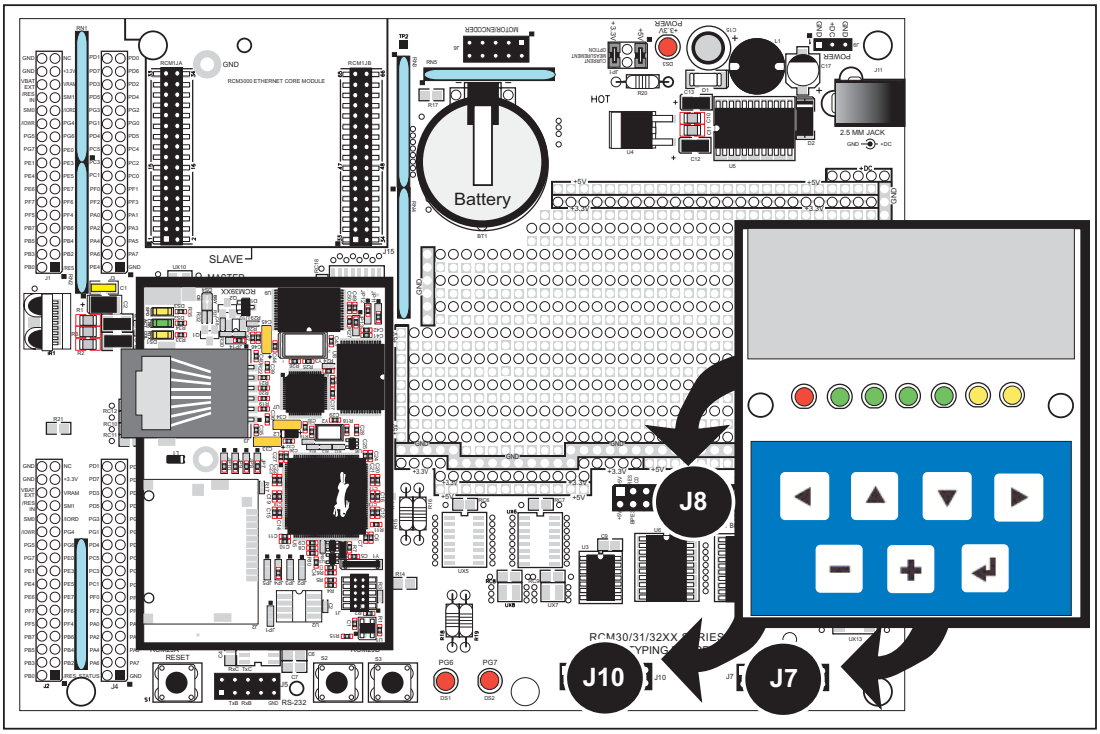


Figure C-7. Install LCD/Keypad Module on Prototyping Board

3. Fasten the unit with the four 4-40 screws and washers included with the LCD/keypad module. If your panel is thick, use a 4-40 screw that is approximately 3/16" (5 mm) longer than the thickness of the panel.

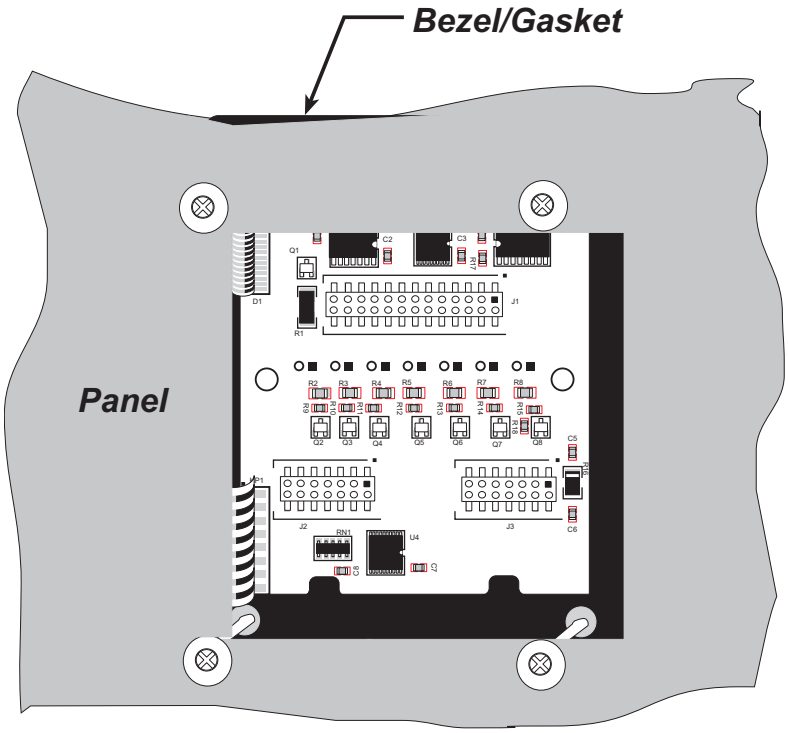


Figure C-9. LCD/Keypad Module Mounted in Panel (rear view)

Carefully tighten the screws until the gasket is compressed and the plastic bezel face-plate is touching the panel.

Do not tighten each screw fully before moving on to the next screw. Apply only one or two turns to each screw in sequence until all are tightened manually as far as they can be so that the gasket is compressed and the plastic bezel faceplate is touching the panel.

glFastFillRegion

```
void glFastFillRegion(int left, int top, int width, int height,
   char pattern);
```

DESCRIPTION

Fills a rectangular block in the LCD buffer with the pattern specified. The block left and width parameters must be byte-aligned. Any portion of the block that is outside the LCD display area will be clipped.

PARAMETERS

left	the *x* coordinate of the top left corner of the block.
top	the *y* coordinate of the top left corner of the block.
width	the width of the block.
height	the height of the block.
pattern	the bit pattern to display (all black if **pattern** is 0xFF, all white if **pattern** is 0x00, and vertical stripes for any other pattern).

RETURN VALUE

None.

SEE ALSO

glFillScreen, glBlankScreen, glBlock, glBlankRegion

glFillPolygon

```
void glFillPolygon(int n, int x1, int y1, int x2, int y2, ...);
```

DESCRIPTION

Fills a polygon in the LCD page buffer and on the LCD if the buffer is unlocked. Any portion of the polygon that is outside the LCD display area will be clipped. If fewer than 3 vertices are specified, the function will return without doing anything.

PARAMETERS

n	the number of vertices.
x1	the *x* coordinate of the first vertex.
y1	the *y* coordinate of the first vertex.
x2	the *x* coordinate of the second vertex.
y2	the *y* coordinate of the second vertex.
...	the coordinates of additional vertices.

RETURN VALUE

None.

SEE ALSO

glFillVPolygon, glPlotPolygon, glPlotVPolygon

glXGetFastmap

```
void glXGetFastmap(int left, int top, int width, int height,
   unsigned long xmemptr);
```

DESCRIPTION

Draws bitmap in the specified space. The data for the bitmap are stored in **xmem**. This function is similar to **glXPutBitmap()**, except that it's faster. The bitmap must be byte-aligned. Any portion of a bitmap image or character that is outside the LCD display area will be clipped.

This function call is intended for use only when a graphic engine is used to interface with the LCD/keypad module.

PARAMETERS

left	the *x* coordinate of the top left corner of the bitmap (*x* must be evenly divisible by 8).
top	the *y* coordinate in pixels of the top left corner of the bitmap.
width	the width of the bitmap (must be evenly divisible by 8).
height	the height of the bitmap.
xmemptr	the **xmem** RAM storage address of the bitmap.

RETURN VALUE

None.

SEE ALSO

glXPutBitmap, glPrintf

glRight1

```
void glRight1(int left, int top, int cols, int rows);
```

DESCRIPTION

Scrolls byte-aligned window right one pixel, left column is filled by current pixel type (color).

PARAMETERS

left	the top left corner of bitmap, must be evenly divisible by 8, otherwise truncates.
top	the top left corner of the bitmap.
cols	the number of columns in the window, must be evenly divisible by 8, otherwise truncates.
rows	the number of rows in the window.

RETURN VALUE

None.

SEE ALSO

glHScroll, glLeft1

TextBorder

```
void TextBorder(windowFrame *wPtr);
```

DESCRIPTION

This function displays the border for a given window frame. This function will automatically adjust the text window parameters to accommodate the space taken by the text border. This adjustment will only occur once after the **TextBorderInit()** function executes.

NOTE: Execute the **TextWindowFrame()** function before using this function.

PARAMETER

wPtr	a pointer to the window frame descriptor.

RETURN VALUE

None.

SEE ALSO

TextBorderInit, TextGotoXY, TextPutChar, TextWindowFrame, TextCursorLocation

TextGotoXY

```
void TextGotoXY(windowFrame *window, int col, int row);
```

DESCRIPTION

Sets the cursor location to display the next character. The display location is based on the height and width of the character to be displayed.

NOTE: Execute the **TextWindowFrame()** function before using this function.

PARAMETERS

window	a pointer to a font descriptor.
col	a character column location.
row	a character row location.

RETURN VALUE

None.

SEE ALSO

TextPutChar, TextPrintf, TextWindowFrame

APPENDIX D. POWER SUPPLY

Appendix D provides information on the current requirements of the RCM3209/RCM3229, and includes some background on the chip select circuit used in power management.

D.1 Power Supplies

The RCM3209/RCM3229 requires a regulated 3.3 V ± 0.15 V DC power source. The RabbitCore design presumes that the voltage regulator is on the user board, and that the power is made available to the RCM3209/RCM3229 board through header J62.

An RCM3209/RCM3229 with no loading at the outputs operating at 29.4 MHz typically draws 145 mA. The RCM3209/RCM3229 will consume an additional 10 mA when the programming cable is used to connect the programming header, J1, to a PC.

D.1.1 Battery Backup

The RCM3209/RCM3229 does not have a battery, but there is provision for a customer-supplied battery to back up the data SRAM and keep the internal Rabbit 3000 real-time clock running.

Header J62, shown in Figure D-1, allows access to the external battery. This header makes it possible to connect an external 3 V power supply. This allows the SRAM and the internal Rabbit 3000 real-time clock to retain data with the RCM3209/RCM3229 powered down.

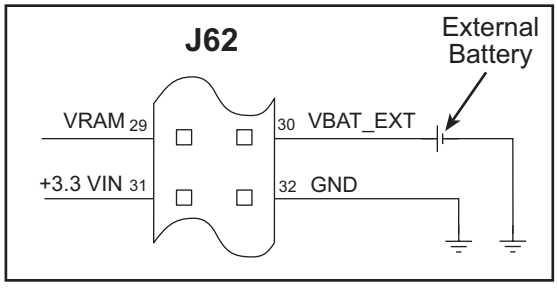


Figure D-1. External Battery Connections at Header J2