**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 (Internal), xD-Picture Card (External) |
| RAM Size | 1MB |
| Connector Type | 2 IDC Headers 2x17, 1 IDC Header 2x5. 1 xD-Picture Card |
| Size / Dimension | 1.85" x 2.73" (47mm x 69mm) |
| Operating Temperature | 0°C ~ 70°C |
| Purchase URL | https://www.e-xfl.com/product-detail/digi-international/101-1087 |

- **Ethernet chip** — A different Ethernet controller chip is used on the RCM3900/RCM3910. The Ethernet chip is able to detect automatically whether a crossover cable or a straight-through cable is being used in a particular setup, and will configure the signals on the Ethernet jack interface.
- **Dynamic C** — As long as no low-level FAT file system calls or direct *xD-Picture Card* access calls to the **NFLASH.LIB** library were used in your application developed for the RCM3365/RCM3375, you may run that application on the RCM3900/RCM3910 after you recompile it using Dynamic C v. 9.60.

  **NOTE:** The Dynamic C RabbitSys option for programming an RCM3365 over an Ethernet link is not supported for the RCM3900.

## 1.3 Advantages of the RCM3365 and RCM3375

- 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
- Program download utility (Rabbit Field Utility) and cloning board options for rapid production loading of programs.
- Generous memory size allows large programs with tens of thousands of lines of code, and substantial data storage.
- Integrated Ethernet port for network connectivity, with royalty-free TCP/IP software.
- Ideal for network-enabling security and access systems, home automation, HVAC systems, and industrial controls

### 1.4.2 Software

The RCM3365 and the RCM3375 are programmed using version 9.24 or later of Dynamic C. A compatible version is included on the Development Kit CD-ROM.

Rabbit is also offering RCM3365 RabbitCore modules preloaded with Dynamic C RabbitSys firmware to allow these modules to run Dynamic C RabbitSys. Dynamic C RabbitSys requires Dynamic C version 9.30 or later, and allows the RCM3365 to be accessed via an Ethernet connection for remote application updates, and for remote monitoring and control. A RabbitSys Development Kit is available with all the hardware and software tools that are needed to develop a RabbitSys application.

Dynamic C v. 9.60 includes the popular µC/OS-II real-time operating system, point-to-point protocol (PPP), FAT file system, RabbitWeb, and other select libraries that were previously sold as individual Dynamic C modules.

Rabbit also offers for purchase 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, or contact your Rabbit sales representative or authorized distributor.

**NOTE:** Version 2.10 or later of the Dynamic C FAT file system module is required to use the FAT file system with the RCM3365 and RCM3375 models.

### 1.4.3 Accessories

Rabbit has available a USB Removable Memory Card Reader and a Connector Adapter Board.

- USB Removable Memory Card Reader (Part No. 20-101-1104)—allows you to read data from the *xD-Picture Card* via your PC.
- Connector Adapter Board (Part No. 151-0114)—allows you to plug the RCM3365/RCM3375 whose headers have a 2 mm pitch into header sockets with a 0.1" pitch.

Visit our Web site at www.rabbit.com or contact your Rabbit sales representative or authorized distributor for further information.

### 1.4.4 Online Documentation

The online documentation is installed along with Dynamic C, and an icon for the documentation menu is placed on the workstation's desktop. Double-click this icon to reach the menu. If the icon is missing, 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 free, unregistered download from our Web sites as well.

*Table 2. RCM3365/RCM3375 Pinout Configurations (continued)*

| Pin | | Pin Name | Default Use | Alternate Use | Notes |
|---|---|---|---|---|---|
| Header J4 | 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 | Output | | External read strobe |
| | 26–27 | SMODE0, SMODE1 | (0,0)—start executing at address zero<br>(0,1)—cold boot from slave port<br>(1,0)—cold boot from clocked Serial Port A<br><br>SMODE0 =1, SMODE1 = 1<br>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 | +3.3 VIN | Power 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 J3 to drive LEDs, these pins can handle a sinking current of up to 8 mA.
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.
3. Do not overload the /IOWR line because the NAND flash memories have critical timing requirements. In some cases it may be necessary to buffer /IOWR on the motherboard.

## 4.4 Memory

### 4.4.1 SRAM

RCM3365/RCM3375 boards have 512K of program-execution fast SRAM at U11. The program-execution SRAM is not battery-backed. There are 512K of battery-backed data SRAM installed at U10.

### 4.4.2 Flash EPROM

RCM3365/RCM3375 boards also have 512K of flash EPROM at U9.

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

Writing to arbitrary flash memory addresses at run time is also discouraged. Instead, use a portion of the "user block" area to store persistent data. The functions **`writeUserBlock`** and **`readUserBlock`** are provided for this. Refer to the ***Rabbit 3000 Microprocessor Designer's Handbook*** and the ***Dynamic C Function Reference Manual*** for additional information.

### 4.4.3 NAND Flash

The RCM3365 and the RCM3375 support a removable *xD-Picture Card*™ to store data and Web pages. The RCM3365 and the RCM3375 both can handle up to a 128MB removable *xD-Picture Card*, and the RCM3365 model also has a 32MB onboard NAND flash.*

> **NOTE:** Rabbit-based systems do not implement the *xD-Picture Card*™ specification for data storage, and are neither compatible nor compliant with *xD-Picture Card*™ card readers.

The NAND flash and *xD-Picture Card* are particularly suitable for mass-storage applications, but are generally unsuitable for direct program execution. The NAND flash differs from parallel NOR flash (the type of flash memory used to store program code on Rabbit-based boards and RabbitCore modules currently in production) in two respects. First, the NAND flash requires error-correcting code (ECC) for reliability. Although NAND flash manufacturers do guarantee that block 0 will be error-free, most manufacturers guarantee that a new NAND flash chip will be shipped with a relatively small percentage of errors, and will not develop more than some maximum number or percentage of errors over its rated lifetime of up to 100,000 writes. Second, the standard NAND flash addressing method multiplexes commands, data, and addresses on the same I/O pins, while requiring that certain control lines must be held stable for the duration of the NAND flash access. The software function calls provided by Rabbit for the NAND flash take care of the data-integrity and reliability attributes.

* RCM3365 modules sold before 2008 had 16MB fixed NAND flash memory.

Figure 10 shows how to insert or remove the *xD-Picture Card*. While you remove or insert the *xD-Picture Card*, take care to avoid touching the electrical contacts on the bottom of the card to prevent electrostatic discharge damage to the card and to keep any moisture or other contaminants off the contacts. Do ***not*** remove or insert the *xD-Picture Card* while it is being accessed.

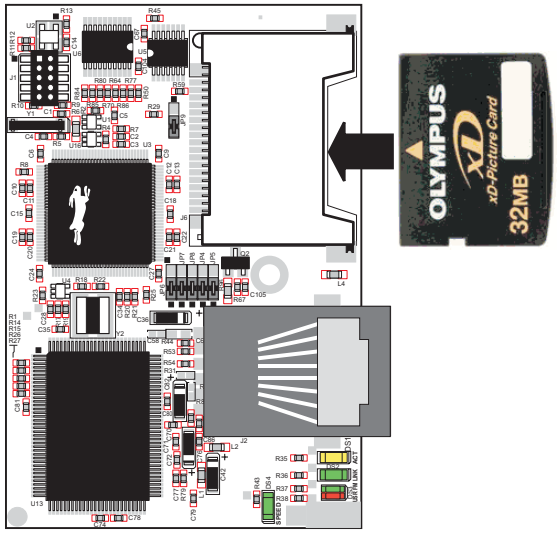



***Figure 10. Insertion/Removal of xD-Picture Card***

It is possible to hot-swap *xD-Picture Cards* without removing power from the RCM3365/RCM3375 module. The file system must be closed before the cards can be hot-swapped. The chip selects associated with the NAND flash and the *xD-Picture Card* must be set to their inactive state, and read/write operations addressed to the NAND flash area cannot be allowed to occur. These operations can be initiated in software by sensing an external switch actuated by the user, and the *xD-Picture Card* can then be removed and replaced with a different one. Once the application program detects a new card, the file system can be opened. These steps allow the *xD-Picture Card* to be installed or removed without affecting either the program, which continues to run on the RCM3365/RCM3375 module, or the data stored on the *xD-Picture Card*.

The **FAT_HOT_SWAP_336x0.C** sample program in the **SAMPLES\FileSystem** folder illustrates this hot-swapping procedure.

Rabbit recommends that you use header J6 only for the *xD-Picture Card* since other devices are not supported. Be careful to remove and insert the *xD-Picture Card* as shown, and be careful ***not*** to insert any foreign objects, which may short out the contacts and lead to the destruction of your *xD-Picture Card*.

Sample programs in the **SAMPLES\RCM3360\NANDFlash** folder illustrate the use of the NAND flash. These sample programs are described in Section 3.2.1, “Use of NAND Flash.” Pay careful attention to the sample programs to see how to close files and secure any data on the *xD-Picture Card* before you remove it.

### 5.1.1 Developing Programs Remotely with Dynamic C

Dynamic C is an integrated development environment that allows you to edit, compile, and debug your programs. Dynamic C has the ability to allow programming over the Internet or local Ethernet. This is accomplished in one of three ways.

1. RCM3365 RabbitCore modules that are preloaded with Dynamic C RabbitSys firmware can be used with Dynamic C RabbitSys to be accessed via an Ethernet connection for remote application updates, and for remote monitoring and control. Dynamic C RabbitSys requires Dynamic C version 9.30 or later, and allows the RCM3365. The ***Dynamic C RabbitSys User's Manual*** provides complete information on RabbitSys.
2. Via the Rabbit RabbitLink, which allows a Rabbit-based target to have programs downloaded to it and debugged with the same ease as exists when the target is connected directly to a PC.
3. Dynamic C provides sample programs to illustrate the use of a download manager, but these sample programs are not intended for use with the NAND flash on the RCM3365 and RCM3375 RabbitCore modules. The **DLM_TCP.C** and **DLP_TCP.C** sample programs found in the Dynamic C **SAMPLES\DOWN_LOAD** folder, are intended to be compiled to the program flash memory (which is a parallel flash memory). Custom applications based on these sample programs may use the NAND flash for data storage.

# 6. USING THE TCP/IP FEATURES

## 6.1 TCP/IP Connections

Programming and development can be done with the RCM3365/RCM3375 modules 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 RCM3365/RCM3375 module's Ethernet port 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 RJ-45 straight-through Ethernet cables and a hub, or an RJ-45 crossover Ethernet cable.

A straight-through and a crossover Ethernet cable are included in the RCM3365/RCM3375 Development Kit. Figure 11 shows how to identify the two cables based on the wires in the transparent RJ-45 connectors.

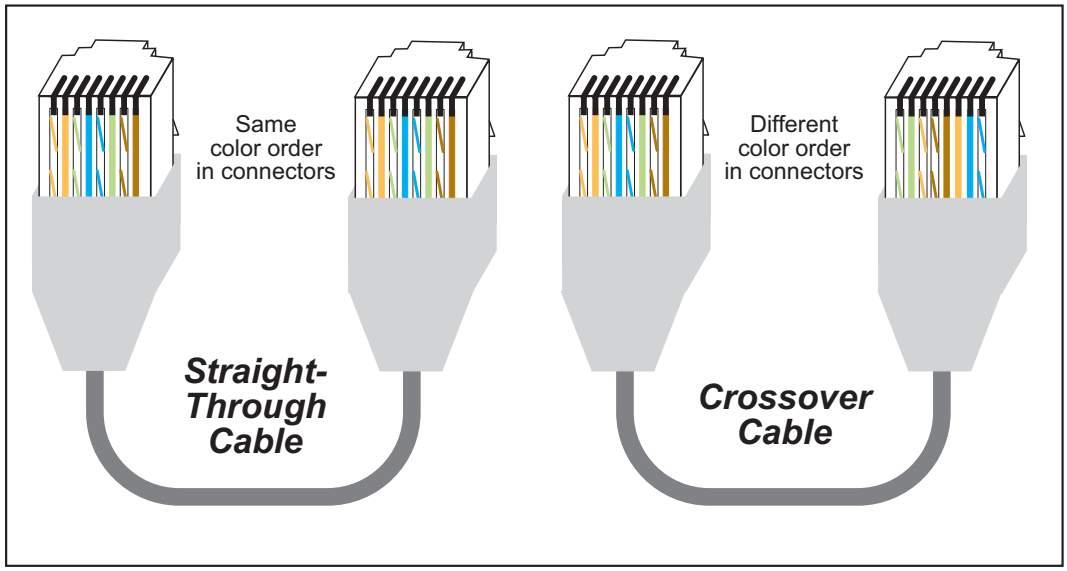



***Figure 11. How to Identify Straight-Through and Crossover Ethernet Cables***

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

# APPENDIX B. PROTOTYPING BOARD

Appendix B describes the features and accessories of the Prototyping Board.

### B.4.6.1 RS-232

RS-232 serial communication on the Prototyping Board is supported by an RS-232 transceiver installed at U9. This transceiver provides the voltage output, slew rate, and input voltage immunity required to meet the RS-232 serial communication protocol. Basically, the chip translates the Rabbit 3000's signals to RS-232 signal levels. Note that the polarity is reversed in an RS-232 circuit so that a +5 V output becomes approximately -10 V and 0 V is output as +10 V. The RS-232 transceiver also provides the proper line loading for reliable communication.

RS-232 can be used effectively at the RCM3365/RCM3375 module's maximum baud rate for distances of up to 15 m.

RS-232 flow control on an RS-232 port is initiated in software using the **serXflowcontrolOn** function call from **RS232.LIB**, where **X** is the serial port (E or F). The locations of the flow control lines are specified using a set of five macros.

**SERX_RTS_PORT**—Data register for the parallel port that the RTS line is on (e.g., PGDR).

**SERX_RTS_SHADOW**—Shadow register for the RTS line's parallel port (e.g., PGDRShadow).

**SERX_RTS_BIT**—The bit number for the RTS line.

**SERX_CTS_PORT**—Data register for the parallel port that the CTS line is on (e.g., PCDRShadow).

**SERX_CTS_BIT**—The bit number for the CTS line.

Standard 3-wire RS-232 communication using Serial Ports E and F is illustrated in the following sample code.

```
#define EINBUFSIZE  15   // set size of circular buffers in bytes
#define EOUTBUFSIZE 15

#define FINBUFSIZE  15
#define FOUTBUFSIZE 15

#define MYBAUD 115200   // set baud rate
#endif

main(){
    serEopen(_MYBAUD);   // open Serial Ports E and F
    serFopen(_MYBAUD);
    serEwrFlush();       // flush their input and transmit buffers
    serErdFlush();
    serFwrFlush();
    serFrdFlush();
    serEclose(_MYBAUD);  // close Serial Ports C and D
    serFclose(_MYBAUD);
}
```

### B.4.8 Other Prototyping Board Modules

An optional LCD/keypad module is available that can be mounted on the Prototyping Board. The signals on headers LCD1JB and LCD1JC will be available only if the LCD/keypad module is installed. Refer to Appendix C, "LCD/Keypad Module," for complete information.

Rabbit's SF1000 series serial flash may be installed in the socket labeled J11. The J11 interface is enabled in software by setting PD2 = 0. Header JP3 must have pins 2–3 jumpered when using the J11 interface. Note that the RabbitNet port and the J11 interface cannot be used simultaneously.

### B.4.9 Quadrature Decoder

Four quadrature decoder inputs are available on screw-terminal header J5. To use the PF0 input from the Rabbit microprocessor, which goes to the QD1B input, remember to reconfigure the jumper on header JP3 to jumper pins 1–2.

Additional information on the use of the quadrature Decoders on Parallel Port F is provided in the ***Rabbit 3000 Microprocessor User's Manual***.

### B.4.10 Stepper-Motor Control

The Prototyping Board can be used to demonstrate the use of the RCM3365/RCM3375 to control a stepper motor. Stepper motor control typically directs moves in two orthogonal directions, and so two sets of stepper-motor control circuits are provided for via screw-terminal headers J3 and J4.

In order to use the stepper-motor control, install two Texas Instruments L293DN chips at locations U2 and U3 (shown in Figure B-10). These chips are readily available from your favorite electronics parts source, and may be purchased through our Web store as part number 660-0205.

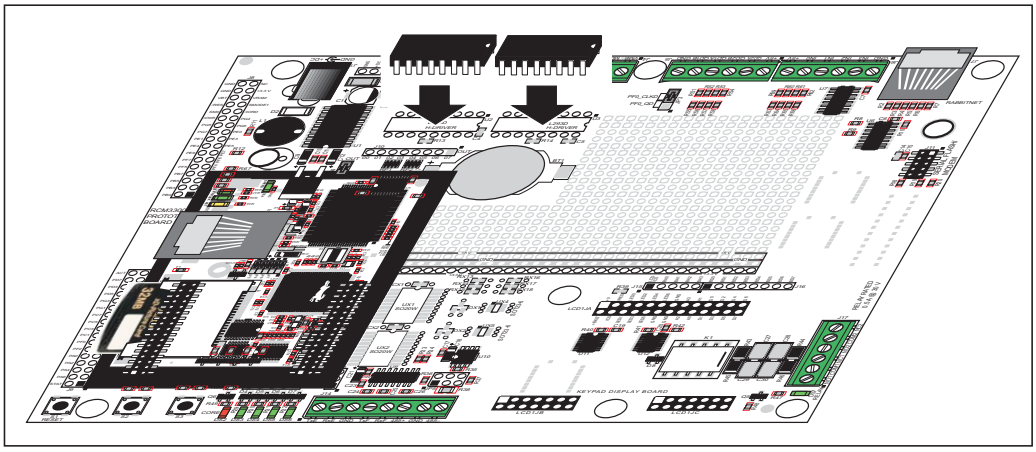


***Figure B-10. Install Four-Channel Push-Pull Driver Chips***

## B.6 Use of Rabbit 3000 Parallel Ports

Table B-5 lists the Rabbit 3000 parallel ports and their use for the Prototyping Board.

***Table B-5. Prototyping Board Use of Rabbit 3000 Parallel Ports***

| Port | I/O | Use | | Initial State |
|---|---|---|---|---|
| PA0–PA3 | Data Bus | LCD/keypad module, motor driver, LEDs | | Active high |
| PA4 | Data Bus | LCD/keypad module, motor driver, relay and relay LED | | Active high |
| PA5–PA7 | Data Bus | LCD/keypad module, motor control | | Active high |
| PB0 | Input | CLKB, *xD-Picture Card* Detect | | High |
| PB1 | Input | CLKA Programming Port | | High (when not driven by CLKA) |
| PB2–PB5 | Address Bus | LCD/keypad module | | High |
| PB6–PB7 | Address Bus | — | | High |
| PC0 | Output | TXD SPI, serial flash | Serial Port D | High (SPI disabled) |
| PC1 | Input | RXD SPI, serial flash | | High (SPI disabled) |
| PC2 | Output | TXC RS-485 | Serial Port C | High (RS-485 disabled) |
| PC3 | Input | RXC RS-485 | | High (RS-485 disabled) |
| PC4* | Output | TXB | Serial Port B | High (disabled) |
| PC5* | Input | RXB | | High (disabled) |
| PC6 | Output | TXA Programming Port | Serial Port A | High |
| PC7 | Input | RXA Programming Port | | High |
| PD0 | Output | RCM3365/RCM3375 **USR** LED off (shared with NAND flash busy) | | High |
| PD1 | Output | Soldered-in NAND flash chip enable | | High (disabled) |
| PD2 | Output | SPI, serial flash | | Low (SPI disabled) |
| PD3 | Output | SPI, serial flash | | High (SPI CS disabled) |
| PD4–PD6 | Input | Serial flash | | High (disabled) |
| PD7 | Output | RS-485 Tx enable | | Low (RS-485 Tx disabled) |
| PE0–PE1 | Input | IN0–IN1 | | High |
| PE2 | Output | Ethernet AEN, NAND flash function enable | | High (disabled) |
| PE3 | Output | Motor driver A clock pulse | | Low (disabled) |
| PE4–PE5 | Input | IN2–IN3, J8 | | High |
| PE6 | Output | LCD/keypad module | | High (disabled) |

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.

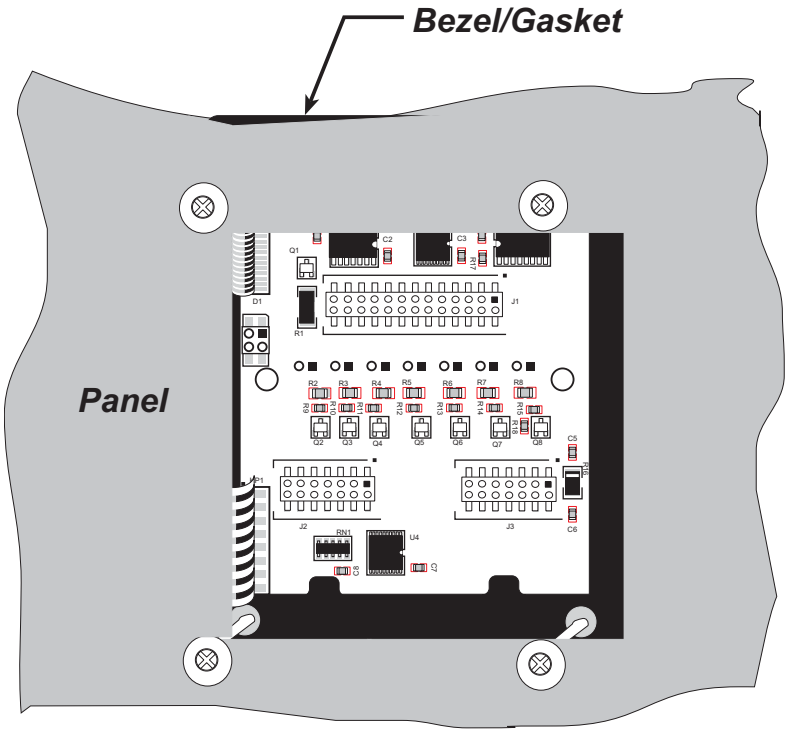



***Figure C-9. LCD/Keypad Module Mounted in Panel (rear view)***

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.

### C.6.1 Connect the LCD/Keypad Module to Your Prototyping Board

The LCD/keypad module can be located as far as 2 ft. (60 cm) away from the Prototyping Board, and is connected via a ribbon cable as shown in Figure C-10.

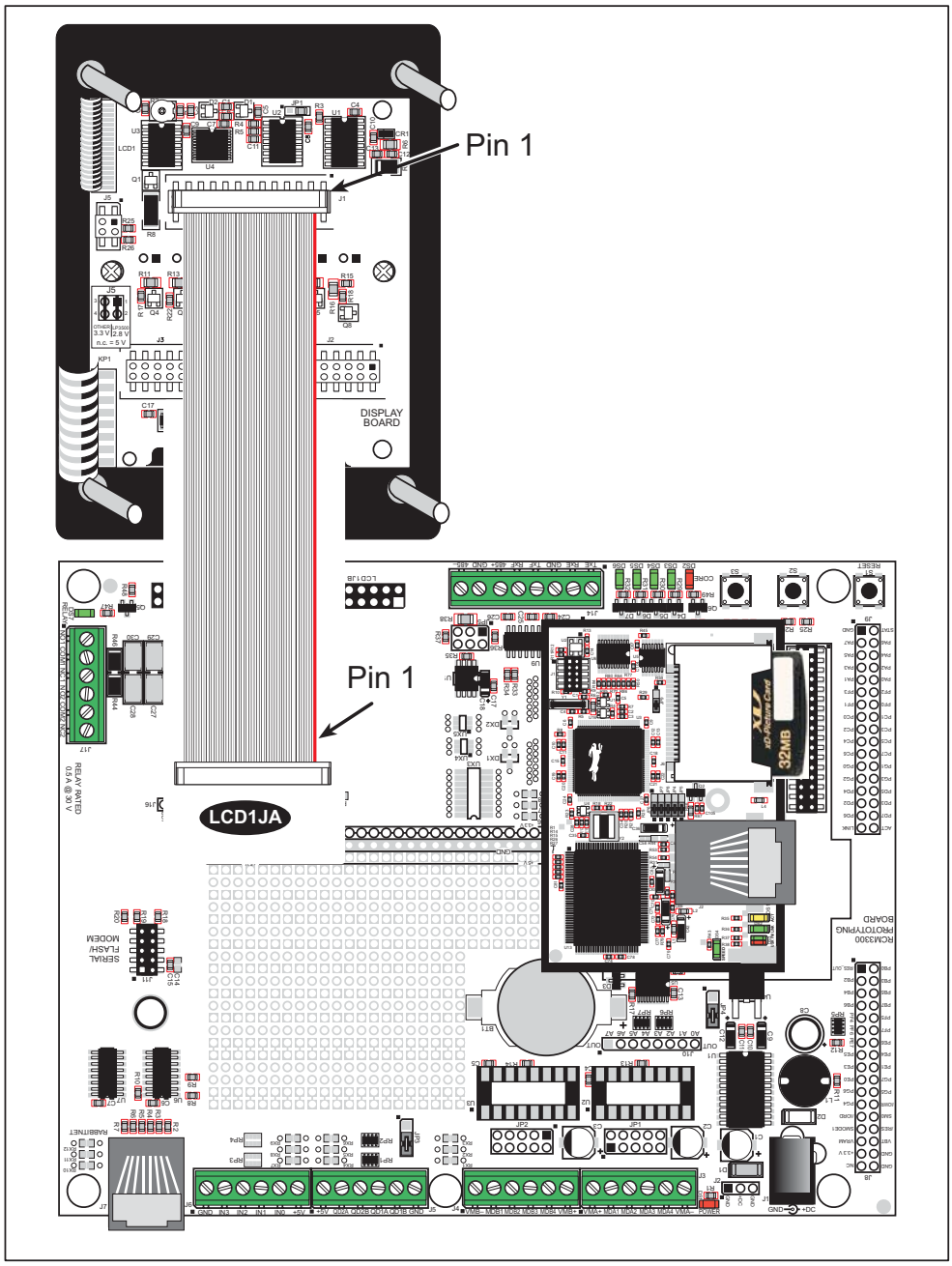



***Figure C-10. Connecting LCD/Keypad Module to Prototyping Board***

Note the locations and connections relative to pin 1 on both the Prototyping Board and the LCD/keypad module.

Rabbit offers 2 ft. (60 cm) extension cables. Contact your authorized distributor or a Rabbit sales representative for more information.

## void glPlotVPolygon(int n, int *pFirstCoord);

Plots the outline of 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** is the number of vertices.

***pFirstCoord** is a pointer to array of vertex coordinates: **x1,y1, x2,y2, x3,y3,...**

### RETURN VALUE

None.

### SEE ALSO

**glPlotPolygon, glFillPolygon, glFillVPolygon**

## void glPlotPolygon(int n, int y1, int x2, int y2, ...);

Plots the outline of 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** is the number of vertices.

**y1** is the *y* coordinate of the first vertex.

**x1** is the *x* coordinate of the first vertex.

**y2** is the *y* coordinate of the second vertex.

**x2** is the *x* coordinate of the second vertex.

**...** are the coordinates of additional vertices.

### RETURN VALUE

None.

### SEE ALSO

**glPlotVPolygon, glFillPolygon, glFillVPolygon**

## void glPrintf(int x, int y, fontInfo *pInfo, char *fmt, ...);

Prints a formatted string (much like **printf**) on the LCD screen. Only the character codes that exist in the font set are printed, all others are skipped. For example, '\b', '\t', '\n' and '\r' (ASCII backspace, tab, new line, and carriage return, respectively) will be printed if they exist in the font set, but will not have any effect as control characters. Any portion of the bitmap character that is outside the LCD display area will be clipped.

### PARAMETERS

**x** is the *x* coordinate (column) of the top left corner of the text.

**y** is the *y* coordinate (row) of the top left corner of the text.

***pInfo** is a font descriptor pointer.

***fmt** is a formatted string.

**...** are formatted string conversion parameter(s).

### EXAMPLE

```
glprintf(0,0, &fi12x16, "Test %d\n", count);
```

### RETURN VALUE

None.

### SEE ALSO

**glXFontInit**

## void glBuffLock(void);

Increments LCD screen locking counter. Graphic calls are recorded in the LCD memory buffer and are not transferred to the LCD if the counter is non-zero.

> **NOTE:** **glBuffLock()** and **glBuffUnlock()** can be nested up to a level of 255, but be sure to balance the calls. It is not a requirement to use these procedures, but a set of **glBuffLock()** and **glBuffUnlock()** bracketing a set of related graphic calls speeds up the rendering significantly.

### RETURN VALUE

None.

### SEE ALSO

**glBuffUnlock, glSwap**

## void glBuffUnlock(void);

Decrements the LCD screen locking counter. The contents of the LCD buffer are transferred to the LCD if the counter goes to zero.

### RETURN VALUE

None.

### SEE ALSO

**glBuffLock, glSwap**

## void glSwap(void);

Checks the LCD screen locking counter. The contents of the LCD buffer are transferred to the LCD if the counter is zero.

**RETURN VALUE**

None.

**SEE ALSO**

**glBuffUnlock, glBuffLock, _glSwapData** (located in the library specifically for the LCD that you are using)

## void glSetBrushType(int type);

Sets the drawing method (or color) of pixels drawn by subsequent graphic calls.

**PARAMETER**

**type** value can be one of the following macros.

**PIXBLACK** draws black pixels (turns pixel on).
**PIXWHITE** draws white pixels (turns pixel off).
**PIXXOR** draws old pixel XOR'ed with the new pixel.

**RETURN VALUE**

None.

**SEE ALSO**

**glGetBrushType**

## int glGetBrushType(void);

Gets the current method (or color) of pixels drawn by subsequent graphic calls.

**RETURN VALUE**

The current brush type.

**SEE ALSO**

**glSetBrushType**

## void glPlotDot(int x, int y);

Draws a single pixel in the LCD buffer, and on the LCD if the buffer is unlocked. If the coordinates are outside the LCD display area, the dot will not be plotted.

**PARAMETERS**

**x** is the *x* coordinate of the dot.

**y** is the *y* coordinate of the dot.

**RETURN VALUE**

None.

**SEE ALSO**

**glPlotline, glPlotPolygon, glPlotCircle**

## void glUp1(int left, int top, int cols, int rows);

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

### PARAMETERS

**left** is the top left corner of bitmap, must be evenly divisible by 8, otherwise truncates.

**top** is the top left corner of the bitmap.

**cols** is the number of columns in the window, must be evenly divisible by 8, otherwise truncates.

**rows** is the number of rows in the window.

### RETURN VALUE

None.

### SEE ALSO

glVScroll, glDown1

## void glDown1(int left, int top, int cols, int rows);

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

### PARAMETERS

**left** is the top left corner of bitmap, must be evenly divisible by 8, otherwise truncates.

**top** is the top left corner of the bitmap.

**cols** is the number of columns in the window, must be evenly divisible by 8, otherwise truncates.

**rows** is the number of rows in the window.

### RETURN VALUE

None.

### SEE ALSO

glVScroll, glUp1

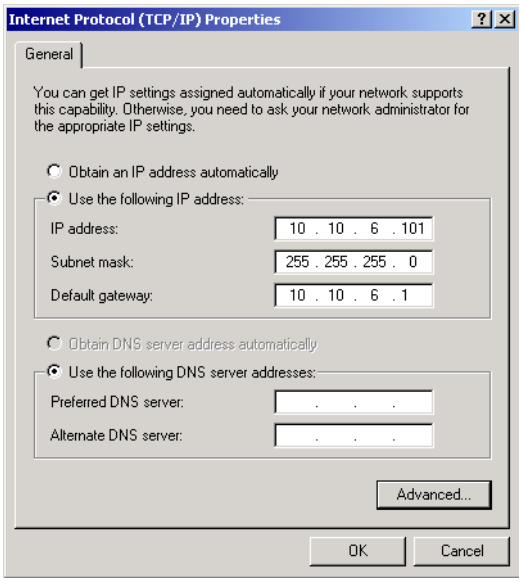
3. Now select the **IP Address** tab, and check **Specify an IP Address**, or select TCP/IP and click on “Properties” to fill in the following fields:

   IP Address : 10.10.6.101

   Netmask : 255.255.255.0

   Default gateway : 10.10.6.1

   **TIP:** If you are using a PC that is normally on a network, you will have disconnected the PC from that network. 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.




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

## J



## K



## L



## M



## P

## R



## S