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	51.6MHz
Flash Size	512KB (Internal), 1MB (External)
RAM Size	1MB
Connector Type	IDC Header 2x25, 2x5, 1x3mm
Size / Dimension	2.35" x 4" (60mm x 102mm)
Operating Temperature	-40°C ~ 70°C
Purchase URL	https://www.e-xfl.com/product-detail/digi-international/101-1006

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

1.1 PowerCore Features

1.1.1 Basic Features

- Small size: 2.35" × 4.00" × 1.08"
(60 mm × 102 mm × 28 mm)
- 39 configurable 5 V tolerant general-purpose I/O lines
- Three additional digital inputs, two additional digital outputs
- Five 3.3 V CMOS-compatible serial ports with a maximum asynchronous baud rate up to 6.45 Mbps. Three ports are configurable as a clocked serial port (SPI), two ports are configurable as HDLC serial ports, and one ports is configurable as an SDLC serial port. One of the serial ports is normally dedicated as a programming port.
- 512K flash memory for storing instructions
- 256K static RAM for data
- Rabbit 3000® microprocessor running at 25.8 MHz. The Rabbit 3000 includes many powerful I/O devices such as serial ports, and precision pulse generation and measurement.
- 50-pin connector brings I/O and I/O bus to customer's motherboard
- Battery backable time/date clock

1.1.2 Options

- Ethernet including RJ-45 connector (10/100 compatible)
- Clock speed 51.6 MHz
- Analog precision ramp generator that can be used in conjunction with low-cost comparators to create rugged A/D converter inputs
- Larger 512K SRAM memory for data
- Second 512K flash memory
- 1 Mbyte serial flash memory that implements file storage
- Coin cell battery to back up the SRAM and the onboard time/date clock
- Onboard +5 V DC switching power supply rated at 1 A or 2 A
- Triac support with AC zero-crossover detection
- Wi-Fi Add-On Kit to enable Wi-Fi interface

3.2.3 D/A Converter

The following sample programs can be found in the **SAMPLES\PowerCoreFLEX\DAC** folder.

- **DAC_CAL.c**—Demonstrates how to calibrate a D/A converter channel using two known voltages to generate the two coefficients, gain and offset, which will be written to the simulated EEPROM in flash memory. This sample program must be compiled to flash memory.

  In order to run this sample program, you will need a voltmeter that is connected to the D/A converter output channels and ground located at header position J3 on the Prototyping Board.

  NOTE: This sample program will overwrite any existing calibration constants.

- **DAC_VOLT.c**—Outputs a voltage that can be read with a voltmeter. The output voltage is computed using the calibration constants that are read from the simulated EEPROM in flash memory. This sample program must be compiled to flash memory.

  In order to run this sample program, you will need a voltmeter that is connected to the selected D/A converter output channel and ground located at header position J3 on the Prototyping Board.

3.2.4 Use of Serial Flash

The following sample programs can be found in the **SAMPLES\PowerCoreFLEX\SERIAL_FLASH** folder.

- **SFLASH_PATTERN_INSPECT.c**—When the sample program starts running, it attempts to initialize a serial flash chip on the PowerCore module. Once a serial flash chip is found, the sample program writes a pattern to the first 100 pages. The Dynamic C **STDIO** window will then display information about the page size. The user can then either print out the contents of a specified page or clear (set to zero) all the bytes in a specified page.
- **SFLASH_TEST.c**—This sample program tests a serial flash chip by performing a write and a read. The results of the test are displayed in the Dynamic C **STDIO** window.

3.2.6 Triacs

The following sample programs can be found in the **SAMPLES\PowerCoreFLEX\TRIAC** folder.

3.2.6.1 Phase-Angle Triac Control

The sample programs demonstrate phase-angle triac control for the PowerCore module and its Prototyping Board. Phase-angle triac control provides you with the ability to fire a triac at a given phase angle of a 50/60 Hz sine wave to provide the desired control for your hardware application.

Once one of the following sample programs is compiled and is running, select the desired triac and the phase angle of where to fire the selected triac in the Dynamic C **STDIO** window.

You may use an oscilloscope or some other load circuit such as the incandescent lamps provided in the Tool Kit to monitor the output. If you are using an oscilloscope, monitor the triac control pin to see the control pin going active at the phase angle from 0 to 180 that you selected. The control pins are located on header J4 of the Prototyping Board:

Pin 41—PF2_SCR-0
Pin 40—PF3_SCR-1

If you are not using an oscilloscope, solder in the incandescent lamps provided in the Tool Kit at DS1 and DS2 on the Prototyping Board before running any of these sample programs. Select option 7 (ramping triacs) from the Dynamic C **STDIO** window to monitor the lamps visually.

DS1—PF2_SCR-0
DS2—PF3_SCR-1

You should see lamps go from being fully ON, then dimming down to fully OFF.

- **TRIAC_PHASE.c**—This program demonstrates basic phase-angle triac control.
- **TRIAC_PHASE_ADC.c**—This program demonstrates phase-angle triac control and reading the A/D ramp circuit when you use the PowerCore module and its Prototyping Board. The Dynamic C **STDIO** window will show the A/D ramp output voltages for the 2.5 V reference voltage, the end-of-ramp-voltage, and the thermistor.
- **TRIAC_PHASE_FLASH.c**—This program demonstrates phase-angle triac control by writing to the flash memory (but ***not*** the serial flash memory).

4.2.3 Programming Port

The PowerCore module's programming port is accessed using header J2 or when programming with a RabbitLink EG2110 through the Ethernet jack. The programming port uses the Rabbit 3000's Serial Port A for communication. Dynamic C uses the programming port to download and debug programs.

The programming port is also used for the following operations.

- Cold-boot the Rabbit 3000 on the PowerCore module after a reset.
- Remotely download and debug a program over an Ethernet connection using the RabbitLink EG2110.
- Fast copy designated portions of flash memory from one Rabbit-based board (the master) to another (the slave) using the Rabbit Cloning Board.

Alternate Uses of the Programming Port

All three clocked Serial Port A signals are available as

- a synchronous serial port
- an asynchronous serial port, with the clock line usable as a general CMOS input

The programming port may also be used as a serial port via the **DIAG** connector on the programming cable.

In addition to Serial Port A, the Rabbit 3000 startup-mode (SMODE0, SMODE1), status, and reset pins are available on the programming port.

The two startup mode pins determine what happens after a reset—the Rabbit 3000 is either cold-booted or the program begins executing at address 0x0000. These two SMODE pins can be used as general inputs once the cold boot is complete.

The status pin is used by Dynamic C to determine whether a Rabbit microprocessor is present. The status output has three different programmable functions:

1. It can be driven low on the first op code fetch cycle.
2. It can be driven low during an interrupt acknowledge cycle.
3. It can also serve as a general-purpose CMOS output.

The /RESET_IN pin is an external input that is used to reset the Rabbit 3000 and the PowerCore module's onboard peripheral circuits.

Refer to the ***Rabbit 3000 Microprocessor User's Manual*** for more information.

5.2.4 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 Technical Note TN213, ***Rabbit Serial Port Software***.

Section 5.2.8.5 provides sample drivers for serial communication when using the Prototyping Board.

The sample programs in the Dynamic C **SAMPLES/PowerCoreFLEX/RS232** folder provide further examples.

5.2.5 TCP/IP Drivers

The TCP/IP drivers are located in the **LIB\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

The Dynamic C **SerialFlash\SFLASH.LIB** library is used to interface to serial flash memory devices on an SPI bus such as the serial flash on board the PowerCore, which uses Serial Port B as an SPI port. The library has two sets of function calls—the first is maintained for compatibility with previous versions of the **SFLASH.LIB** library. The functions are all blocking and only work for single flash devices. The new functions, which should be used for the PowerCore, make use of an **sf_device** structure as a handle for a specific serial flash device. This allows multiple devices to be used by an application.

More information on these function calls is available in the ***Dynamic C Function Reference Manual***.

The sample programs in the Dynamic C **SAMPLES/PowerCoreFLEX/SERIAL_FLASH** folder provide further examples.

digOut

```
void digOut(int channel, int state);
```

DESCRIPTION

Sets the state of digital outputs OUT00–OUT03 on Prototyping Board header J2.

PARAMETERS

channel the digital output channel OUT00–OUT03:

0 = OUT00 (sinking type output)
1 = OUT01 (sinking type output)
2 = OUT02 (sourcing type output)
3 = OUT03 (sourcing type output)

state the output logic value (0 or 1) to output:

Sinking driver
0 = connects the load to GND
1 = puts the output in a high-impedance state.

Sourcing driver
0 = puts the output in a high-impedance state
1 = connects the load to +K.

RETURN VALUE

None.

SEE ALSO

brdInit, digIn

6. Using the TCP/IP Features

6.1 TCP/IP Connections

Programming and development can be done with the PowerCore FLEX 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 PowerCore FLEX 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.

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

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

   There are four options for connecting the PowerCore FLEX module to a network for development and run-time 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 network users. We recommend one of these options for initial development.

- **No LAN —** The simplest alternative for desktop development. Connect the PowerCore module's Ethernet port directly to the PC's network interface card using an RJ-45 *crossover cable*. A crossover cable is a special cable that flips some connections between the two connectors and permits direct connection of two client systems. A standard RJ-45 network cable will not work for this purpose.
- **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 PowerCore module's Ethernet port to it using standard network cables.

APPENDIX A. POWERCORE SPECIFICATIONS

Appendix A provides the specifications for the PowerCore, and describes the conformal coating.

B.4.4 Triac Outputs

The Prototyping Board has two triacs, each of which can handle up to 1 A. The Rabbit 3000 can enable or disable the triacs via software calls. Figure B-6 shows a schematic diagram for the two triac outputs.

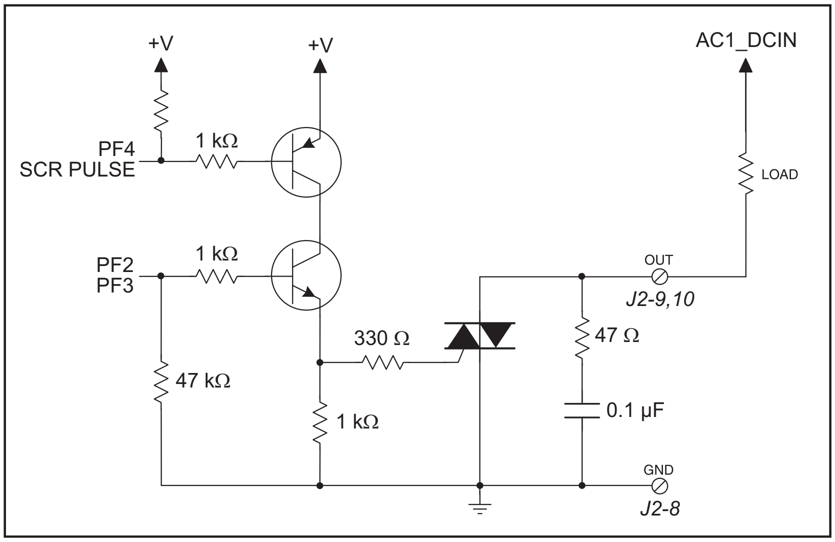

Figure B-6. PowerCore Prototyping Board Triac Outputs

When the triac outputs are being used, the AC must be supplied by a transformer with one side of the transformer connected to logic ground. This connection is made automatically when you use the center-tapped transformer supplied with the PowerCore Tool Kit, or you can use a half-wave rectified power supply.

B.4.5.2 D/A Converter Circuits

The Prototyping Board has three D/A converter circuits, each with a different settling time:

- DAC0—settling time 0.4 ms
- DAC1—settling time 0.9 ms
- DAC2—settling time 2.5 ms

Figure B-9 shows a schematic for the D/A converter circuits used for DAC1 and DAC2.

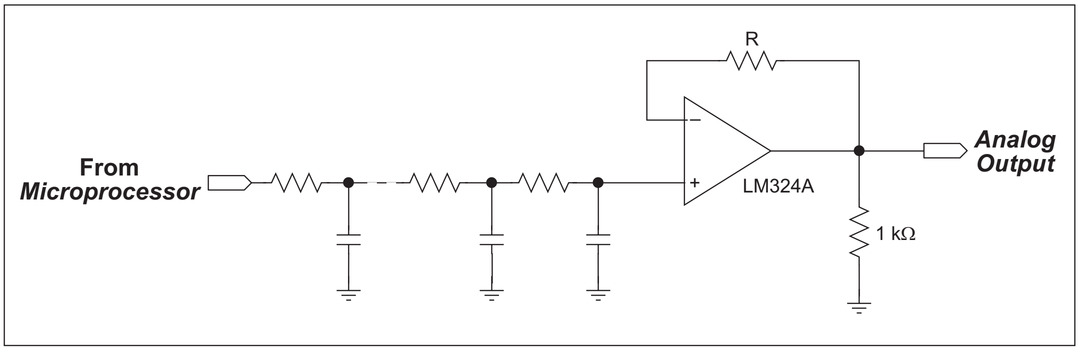

Figure B-9. PowerCore Prototyping Board D/A Converter Circuits DAC1 and DAC2

The three D/A converters differ in the number and kind of RC filtering stages.

- DAC0 has a faster response filter that is still able to filter out PWM noise. It uses a 5-pole active filter configuration with two op-amps.
- DAC1 has a less expensive 5-pole passive filter that has a slower response, but still filters out PWM noise.
- DAC2 has the least expensive filter, a 3-pole passive filter. The trade-off is a slower response time for the D/A converter.

Each D/A converter outputs 3.0 V into a typical 10 kΩ load.

C.6 Mounting LCD/Keypad Module on the Prototyping Board

Install the LCD/keypad module on header sockets LCD1:JA, LCD1:JB, and LCD1:JC of the Prototyping Board as shown in Figure C-8. 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.

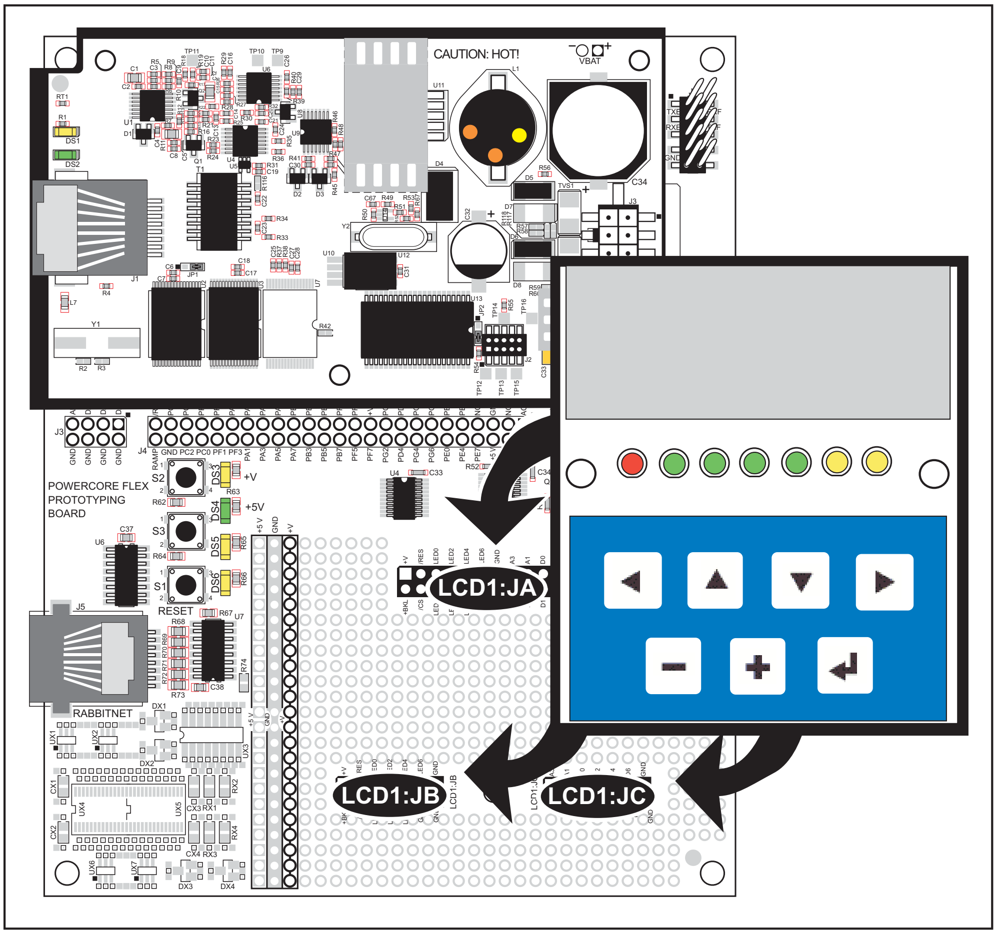

Figure C-8. Install LCD/Keypad Module on Prototyping Board

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.

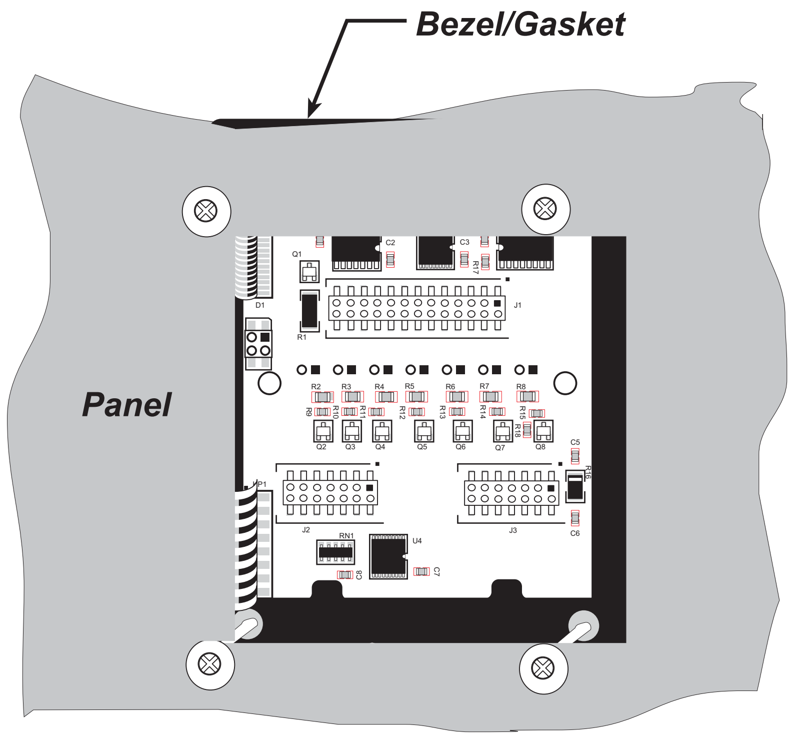

Figure C-10. LCD/Keypad Module Mounted in Panel (rear view)

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.7.1 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 PowerCore Prototyping Board, and is connected via a ribbon cable as shown in Figure C-11.

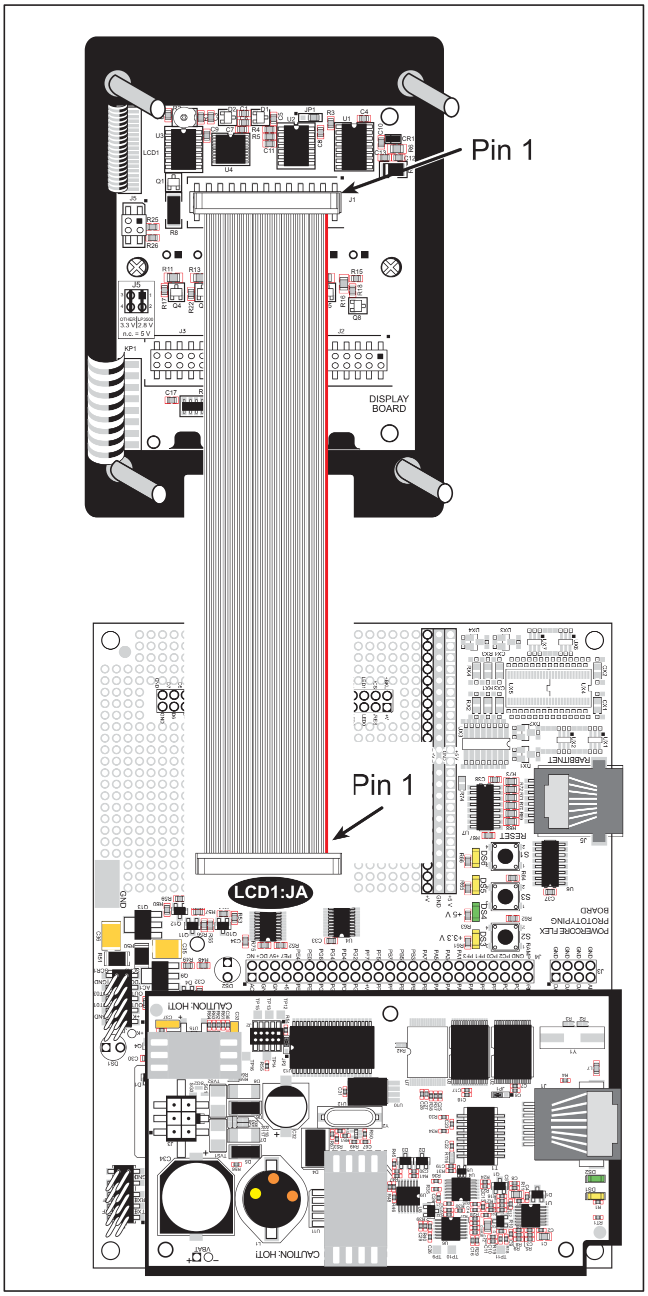

Figure C-11. Connecting LCD/Keypad Module to PowerCore Prototyping Board

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

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

glPlotDot

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

DESCRIPTION

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`	the *x* coordinate of the dot.
`y`	the *y* coordinate of the dot.

RETURN VALUE

None.

SEE ALSO

`glPlotline, glPlotPolygon, glPlotCircle`

glPlotLine

```
void glPlotLine(int x0, int y0, int x1, int y1);
```

DESCRIPTION

Draws a line in the LCD buffer, and on the LCD if the buffer is unlocked. Any portion of the line that is beyond the LCD display area will be clipped.

PARAMETERS

`x0`	the *x* coordinate of one endpoint of the line.
`y0`	the *y* coordinate of one endpoint of the line.
`x1`	the *x* coordinate of the other endpoint of the line.
`y1`	the *y* coordinate of the other endpoint of the line.

RETURN VALUE

None.

SEE ALSO

`glPlotDot, glPlotPolygon, glPlotCircle`

glXPutBitmap

```
void glXPutBitmap(int left, int top, int width, int height,
   unsigned long bitmap);
```

DESCRIPTION

Draws bitmap in the specified space. The data for the bitmap are stored in **xmem**. This function calls **glXPutFastmap** automatically if the bitmap is byte-aligned (the left edge and the width are each evenly divisible by 8).

Any portion of a bitmap image or character that is outside the LCD display area will be clipped.

PARAMETERS

left	the top left corner of the bitmap.
top	the top left corner of the bitmap.
width	the width of the bitmap.
height	the height of the bitmap.
bitmap	the address of the bitmap in **xmem**.

RETURN VALUE

None.

SEE ALSO

glXPutFastmap, glPrintf

C.9.4 Keypad

The functions used to control the keypad are contained in the Dynamic C **`LIB\KEYPADS\KEYPAD7.LIB`** library.

keyInit

```
void keyInit(void);
```

DESCRIPTION

Initializes keypad process.

RETURN VALUE

None.

SEE ALSO

`brdInit`

D.3 Reset Generator

The PowerCore module uses a reset generator to reset the Rabbit 3000 microprocessor when the voltage drops below the voltage necessary for reliable operation. The reset typically occurs at 4.38 V.

The PowerCore module has a reset pin, pin 49 on header J4. This pin provides access to the reset output of the reset generator, which drives the reset input of the Rabbit 3000 and peripheral circuits. The /RESET output can be used to reset user-defined circuits on the motherboard on which the PowerCore module is mounted.

rn_read

```
int rn_read(int handle, int regno, char *recdata, int datalen);
```

DESCRIPTION

Reads a string from the specified device and register. Waits for results. This function will check device information to determine that the peripheral card is connected to a master.

PARAMETERS

handle	is an address index to device information. Use **rn_device()** or **rn_find()** to establish the handle.
regno	is the command register number as designated by each device.
recdata	is a pointer to the address of the string to read from the device.
datalen	is the number of bytes to read (0–15).

NOTE: A data length of 0 will transmit the one-byte command register number.

RETURN VALUE

The status byte from the previous command. -1 means that device information indicates the peripheral card is not connected to the master, and -2 means that the data length was greater than 15.

SEE ALSO

rn_write