RabbitCore RCM2200

C-Programmable Module with Ethernet

User's Manual

019–0097 • 010418–A

RabbitCore RCM2200: User's Manual

Part Number 019-0097 • 010418–A • Printed in U.S.A.

Notice to Users

Z-WORLD PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE-SUPPORT DEVICES OR SYSTEMS UNLESS A SPECIFIC WRITTEN AGREEMENT REGARDING SUCH INTENDED USE IS ENTERED INTO BETWEEN THE CUSTOMER AND ZWORLD PRIOR TO USE. Life-support devices or systems are devices or systems intended for surgical implantation into the body or to sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling and user's manual, can be reasonably expected to result in significant injury.

No complex software or hardware system is perfect. Bugs are always present in a system of any size. In order to prevent danger to life or property, it is the responsibility of the system designer to incorporate redundant protective mechanisms appropriate to the risk involved.

Trademarks

Rabbit Semiconductor

2932 Spafford Street
Davis, California 95616-6800
USA

Telephone: (530) 757-8400
Fax: (530) 757-8402

www.rabbitsemiconductor.com

Z-World Inc.

2900 Spafford Street
Davis, California 95616-6800
USA

Telephone: (530) 757-3737
Fax: (530) 753-5141

www.zworld.com

Table of Contents

Introduction 1

The RabbitCore RCM2200 microprocessor module is designed to be the heart of embedded control systems. The RCM2200 features an integrated Ethernet port and provides for LAN and Internet-enabled systems to be built as easily as serial-communication systems.

The RabbitCore RCM2200 has a Rabbit 2000 microprocessor operating at 22.1 MHz, static RAM, flash memory, two clocks (main oscillator and timekeeping), and the circuitry necessary for reset and management of battery backup of the Rabbit 2000's internal real-time clock and the static RAM. Two 26-pin headers bring out the Rabbit 2000 I/O bus lines, address lines, data lines, parallel ports, and serial ports.

The RabbitCore RCM2200 receives its +5 V power from the user board on which it is mounted. The RabbitCore RCM2200 can interface with all kinds of CMOS-compatible digital devices through the user board.

1.1 RabbitCore RCM2200 Features

- Small size: 1.60" × 2.30" × 0.86" (41 mm × 58 mm × 22 mm)
- Microprocessor: Rabbit 2000 running at 22.1 MHz
- 26 parallel I/O lines: 16 configurable for input or output, 7 fixed inputs, 3 fixed outputs
- 8 data lines (D0–D7)

- 4 address lines (A0–A3)
- Memory I/0 read, write
- External reset input
- Five 8-bit timers (cascadable in pairs) and two 10-bit timers
- 256K flash memory, 128K SRAM
- Real-time clock
- Watchdog supervisor
- Provision for customer-supplied backup battery via connections on header J5
- 10Base-T RJ-45 Ethernet port
- Raw Ethernet and two associated LED control signals available on 26-pin header
- Three CMOS-compatible serial ports: maximum asynchronous baud rate of 345,600 bps, maximum synchronous baud rate of 138,240 bps. One port is configurable as a clocked port.
- Six additional I/O lines are located on the programming port, can be used as I/O lines when the programming port is not being used for programming or in-circuit debugging—one synchronous serial port can also be used as two general CMOS inputs and one general CMOS output, and there are two additional inputs and one additional output.

Appendix A, "RabbitCore RCM2200 Specifications," provides detailed specifications for the RabbitCore RCM2200.

1.2 Advantages of the RabbitCore RCM2200

- Fast time to market using a fully engineered, "ready to run" microprocessor core.
- Competitive pricing when compared with the alternative of purchasing and assembling individual components.
- Easy C-language program development and debugging, including 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, royalty-free TCP/IP software.

1.3 Development and Evaluation Tools

A complete Development Kit, including a Prototyping Board, and Dynamic C development software, is available to accompany the RCM2200 module. The Development Kit puts together the essentials you need to design an embedded microprocessor-based system rapidly and efficiently.

See the ***RabbitCore RCM2200 Getting Started Manual*** for complete information on the Development Kit.

1.4 How to Use This Manual

This user's manual is intended to give users detailed information on the RCM2200 module. It does not contain detailed information on the Dynamic C development environment or the TCP/IP software support for the integrated Ethernet port. Most users will want more detailed information on some or all of these topics in order to put the RCM2200 module to effective use.

1.4.1 Additional Product Information

Introductory information about the RabbitCore RCM2200 and its associated Development Kit and Prototyping Board will be found in the printed ***RabbitCore RCM2200 Getting Started Manual,*** which is also provided on the accompanying CD-ROM in both HTML and Adobe PDF format.

We recommend that any users unfamiliar with Z-World products, or those who will be using the prototyping board for initial evaluation and development, begin with at least a read-through of the ***Getting Started*** manual.

1.4.2 Additional Reference Information

In addition to the product-specific information contained in the ***RabbitCore RCM2200 Getting Started Manual*** and the ***RabbitCore RCM2200 User's Manual*** (this manual), several higher level reference manuals are provided in HTML and PDF form on the accompanying CD-ROM. Advanced users will find these references valuable in developing systems based on the RCM2200 modules:

- *Dynamic C Premier User's Manual*
- *Introduction to TCP/IP*
- *TCP/IP Function Reference*
- *Rabbit 2000 Microprocessor User's Manual*

1.4.3 Using Online Documentation

We provide the bulk of our user and reference documentation in two electronic formats, HTML and Adobe PDF. We do this for several reasons.

We believe that providing all users with our complete library of product and reference manuals is a useful convenience. However, printed manuals are expensive to print, stock, and ship. Rather than include and charge for manuals that every user may not want, or provide only product-specific manuals, we choose to provide our complete documentation and reference library in electronic form with every Development Kit and with our Dynamic C development environment.

> **NOTE:** The most current version of Adobe Acrobat Reader can always be downloaded from Adobe's Web site at **http://www.adobe.com**. We recommend that you use version 4.0 or later.

Providing this documentation in electronic form saves an enormous amount of paper by not printing copies of manuals that users don't need. It reduces the number of outdated manuals we have to discard from stock as well, and it makes providing a complete library of manuals an almost cost-free option. For one-time or infrequent reference, electronic documents are more convenient than printed ones—after all, they aren't taking up shelf or desk space!

Finding Online Documents

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.

Printing Electronic Manuals

We recognize that many users prefer printed manuals for some uses. Users can easily print all or parts of those manuals provided in electronic form. The following guidelines may be helpful:

- Print from the Adobe PDF versions of the files, not the HTML versions.
- Print only the sections you will need to refer to more than once.
- Print manuals overnight, when appropriate, to keep from tying up shared resources during the work day.
- If your printer supports duplex printing, print pages double-sided to save paper and increase convenience.
- If you do not have a suitable printer or do not want to print the manual yourself, most retail copy shops (e.g., Kinkos, AlphaGraphics, CopyMax) will print the manual from the PDF file and bind it for a reasonable charge—about what we would have to charge for a printed and bound manual.

Hardware Reference 2

Chapter 2 describes the hardware components and principal hardware subsystems of the RabbitCore RCM2200. Appendix A, "RabbitCore RCM2200 Specifications," provides complete physical and electrical specifications.

2.1 RabbitCore RCM2200 Digital Inputs and Outputs

Figure 2–1 shows the subsystems designed into the RabbitCore RCM2200.

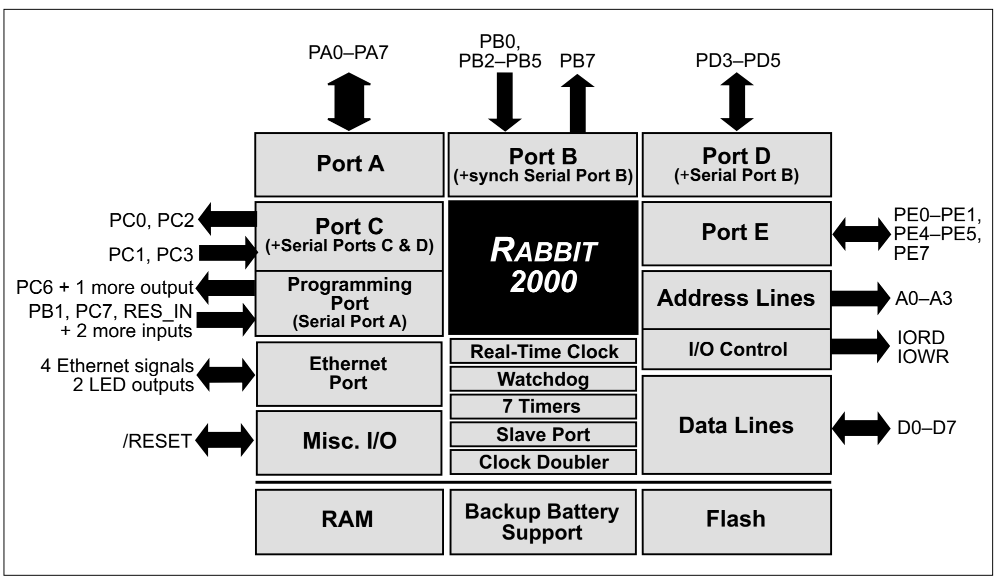


Figure 2–1: Rabbit Subsystems

The RabbitCore RCM2200 has 26 parallel I/O lines grouped in five 8-bit ports available on headers J4 and J5. The 16 bidirectional I/O lines are located on pins PA0–PA7, PD3–PD5, and PE0–PE1, PE4, PE5, and PE7. The pinouts for headers J4 and J5 are shown in Figure 2–2.

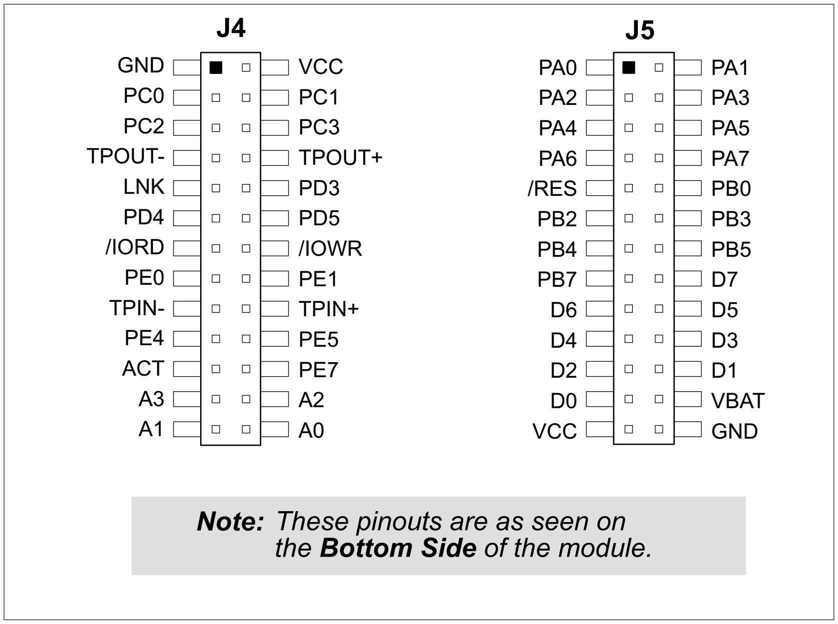


Figure 2–2: RabbitCore RCM2200 I/O Pinouts

2.1.1 Dedicated Inputs

PB0 is a general CMOS input when the Rabbit 2000 is either not using Serial Port B or is using Serial Port B in an asynchronous mode. Four other general CMOS input-only pins are located on PB2–PB5. These pins can also be used for the slave port. PB2 and PB3 are slave write and slave read strobes, while PB4 and PB5 serve as slave address lines SA0 and SA1, and are used to access the slave registers. PC1 and PC3 are general CMOS inputs only. These pins can instead be selectively enabled to serve as the serial data inputs for Serial Ports D and C.

2.1.2 Dedicated Outputs

One of the general CMOS output-only pins is located on PB7. PB7 can also be used with the slave port as the /SLAVEATTN output. This configuration signifies that the slave is requesting attention from the master. PC0 and PC2 are also output-only pins; PC0 and PC2 can instead serve as the serial data outputs for Serial Ports D and C.

2.1.3 Memory I/O Interface

Four of the Rabbit 2000 address lines (A0–A3) and all the data lines (D0–D7) are availabl. I/0 write (/IOWR) and I/0 read (/IORD) are also available for interfacing to external devices.

The ports on the Rabbit 2000 microprocessor used in the RabbitCore RCM2200 are configurable, and so the factory defaults can be reconfigured. Table 2–1 lists the Rabbit 2000 factory defaults and the alternate configurations.

2.1.4 Other Inputs and Outputs

As shown in Table 2–1, pins PA0–PA7 can be used to allow the Rabbit 2000 to be a slave to another processor. The slave port also uses PB2–PB5, PB7, and PE7.

Table 2–1: RabbitCore RCM2200 Pinout Configurations

Pin		Pin Name	Default Use	Alternate Use	Notes
Header J4	1	GND			
	2	VCC			
	3	PC0	Output	TXD	
	4	PC1	Input	RXD	
	5	PC2	Output	TXC	
	6	PC3	Input	RXC	
	7	TPOUT–			Ethernet transmit port
	8	TPOUT+			
	9	LNK			Ethernet LNK LED indicator
	10	PD3	Bitwise or parallel programmable I/O		
	11	PD4		ATXB output	
	12	PD5		ARXB input	
	13	/IORD	Input (I/O read strobe)		
	14	/IOWR	Output (I/O write strobe)		
	15	PE0	Bitwise or parallel programmable I/O	I0 control or INT0A input	
	16	PE1		I1 control or INT1A input	
	17	TPIN–			Ethernet receive port
	18	TPIN+			
	19	PE4	Bitwise or parallel programmable I/O	I4 control or INT0B input	
	20	PE5		I5 control or INT1B input	
	21	ACT			Ethernet active (ACT) LED indicator
	22	PE7	Bitwise or parallel programmable I/O	I7 control or slave port chip select /SCS	
	23–26	A[3:0]			Rabbit 2000 address bus

Table 2–1: RabbitCore RCM2200 Pinout Configurations (continued)

Pin		Pin Name	Default Use	Alternate Use	Notes
Header J5	1–8	PA[0:7]	Bytewide programmable parallel I/O	Slave port data bus SD0–SD7	
	9	/RESET	Reset output	Reset input	This weak output can be driven externally
	10	PB0	Input	Serial port clock CLKB input or output	
	11	PB2	Input	Slave port write /SWR	
	12	PB3	Input	Slave port read /SRD	
	13	PB4	Input	SA0	Slave port address lines
	14	PB5	Input	SA1	
	15	PB7	Output	Slave port attention line /SLAVEATTN	
	16–23	D[7:0]	Input/Output		Rabbit 2000 data bus
	24	VBAT	3 V battery input		
	25	VCC			
	26	GND			

PE0, PE1, PE4, and PE5 can be used for up to two external interrupts. PB0 can be used to access the clock on Serial Port B of the Rabbit microprocessor. PD4 can be programmed to be a serial output for Serial Port B. PD5 can be used as a serial input by Serial Port B.

PC4, PC5, PD0, PD1, PE2, PE3, and PE6 are used for internal communication with the RealTek Ethernet interface chip.

2.2 Serial Communication

The RabbitCore RCM2200 board does not have an RS-232 or an RS-485 transceiver directly on the board. However, an RS-232 or RS-485 interface may be incorporated on the board the RCM2200 is mounted on. For example, the Prototyping Board supports a standard RS-232 transceiver chip.

2.2.1 Serial Ports

There are four serial ports designated as Serial Ports A, B, C, and D. All four serial ports can operate in an asynchronous mode up to the baud rate of the system clock divided by 64. An asynchronous port can handle 7 or 8 data bits. A 9th bit address scheme, where an additional bit is sent to mark the first byte of a message, is also supported. Serial Ports A and B can also be operated in the clocked serial mode. In this mode, a clock line synchronously clocks the data in or out. Either of the two communicating devices can supply the clock. When the Rabbit 2000 provides the clock, the baud rate can be up to 80% of the system clock frequency divided by 128, or 138,240 bps for a 22.1 MHz clock speed.

Serial Port A is available only on the programming port, and so is likely to be inconvenient to interface with.

2.2.2 Ethernet Port

Figure 2–3 shows the pinout for the RJ-45 Ethernet port (J2). Note that some Ethernet connectors are numbered in reverse to the order used here.

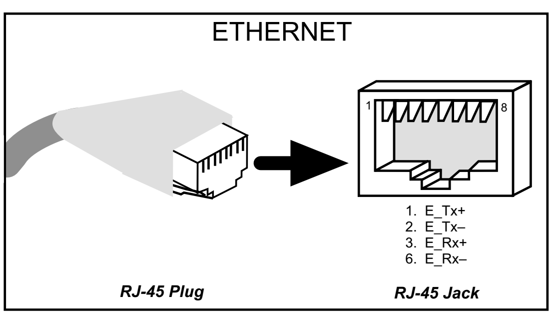


Figure 2–3: RJ-45 Ethernet Port Pinout

Two LEDs are placed next to the RJ-45 Ethernet jack, one to indicate an Ethernet link (**LNK**) and one to indicate Ethernet activity (**ACT**).

The Ethernet signals are also available on header J4. The **ACK** and **LNK** signals can be used to drive LEDs on the user board the RCM2200 is connected to.

The transformer/connector assembly ground is connected to the RabbitCore RCM2200 printed circuit board digital ground via a 0 Ω resistor, R29, as shown in Figure 2–4.

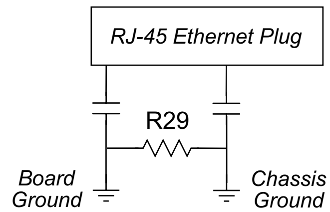


Figure 2–4: Isolation Resistor R29

The RJ-45 connector is shielded to minimize EMI effects to/from the Ethernet signals. Z-World recommends that an equivalent RJ-45 connector be used on the user board if the customer wishes to have an RJ-45 connector on the user board.

> **NOTE:** Z-World may offer large-volume orders for the RCM2200 without the LEDs and the RJ-45 connector if you plan to use your own RJ-45 connector on your user board. Ask your Z-World or Rabbit Semiconductor sales representative for more information.

2.2.3 Programming Port

Serial Port A has special features that allow it to cold-boot the system after reset. Serial Port A is also the port that is used for software development under Dynamic C.

The RabbitCore RCM2200 has a 10-pin program header labeled J1. The Rabbit 2000 startup-mode pins (SMODE0, SMODE1) are presented to the programming port so that an externally connected device can force the RCM2200 to start up in an external bootstrap mode. The ***Rabbit 2000 Microprocessor User's Manual*** provides more information related to the bootstrap mode.

The programming port is used to start the RabbitCore RCM2200 in a mode where it will download a program from the port and then execute the program. The programming port transmits information to and from a PC while a program is being debugged in-circuit.

The RabbitCore RCM2200 can be reset from the programming port via the /RESET_IN line.

The Rabbit 2000 status pin is also presented to the programming port. The status pin is an output that can be used to send a general digital signal.

The clock line for Serial Port A is presented to the programming port, which makes synchronous serial communication possible.

2.2.3.1 Alternate Uses of the Programming Port

The programming port may also be used as an application port with the **DIAG** connector on the programming cable.

All three clocked Serial Port A signals are available as

- a synchronous serial port
- an asynchronous serial port, with the clock line useable as a general CMOS input
- two general CMOS inputs and one general CMOS output.

Two startup mode pins, SMODE0 and SMODE1, are available as general CMOS inputs after they are read during the initial boot-up. The logic state of these two pins is very important in determining the startup procedure after a reset.

/RES_IN is an external input used to reset the Rabbit 2000 microprocessor.

The status pin may also be used as a general CMOS output.

See Appendix C, "Programming Cable," for more information.

2.3 Other Hardware

2.3.1 Clock Doubler

The RabbitCore RCM2200 takes advantage of the Rabbit 2000 microprocessor's internal clock doubler. A built-in clock doubler allows half-frequency crystals to be used to reduce radiated emissions. The 22.1 MHz frequency is generated using an 11.0592 MHz crystal. The clock doubler is disabled automatically in the BIOS for crystals with a frequency above 12.9 MHz.

The clock doubler may be disabled if 22.1 MHz clock speeds are not required. Disabling the Rabbit 2000 microprocessor's internal clock doubler will reduce power consumption and further reduce radiated emissions. The clock doubler is disabled with a simple change to the BIOS as described below.

1. Open the BIOS source code file, **`RABBITBIOS.C`** in the **`BIOS`** directory.
2. Change the line

```
#define CLOCK_DOUBLED 1 // set to 1 to double the clock if XTAL<=12.9MHz,
```

   to read as follows.

```
#define CLOCK_DOUBLED 0 // set to 1 to double the clock if XTAL<=12.9MHz,
```

3. Change the serial baud rate to 57,600 bps when the RabbitCore RCM2200 is operated at 11.05 MHz.
4. Save the change using **File > Save**.

2.3.2 Backup Battery Circuit

The RabbitCore RCM2200 does not have a battery, but there is provision for a customer-supplied battery to back up SRAM and keep the internal Rabbit 2000 real-time clock running.

Header J5, shown in Figure 2–5, allows access to the external battery. This header makes it possible to connect an external 3 V power supply. This allows the internal Rabbit 2000 real-time clock to run and allows the SRAM to retain data when the RabbitCore RCM2200 is powered down.

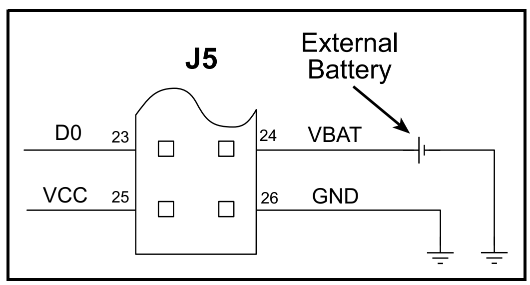


Figure 2–5: External Battery Connections at Header J5

A lithium battery with a nominal voltage of 3 V and a minimum capacity of 950 mA·h is recommended. A lithium battery is needed because of its nearly constant nominal voltage over most of its life.

The drain on the battery by the RabbitCore RCM2200 is typically 16 µA when no other power is supplied. If a 950 mA·h battery is used, the battery can last more than 6 years:

$$\frac{950\ \text{mA}\cdot\text{h}}{16\ \mu\text{A}} = 6.8\ \text{years.}$$

The actual life in your application will depend on the current drawn by components not on the RabbitCore RCM2200 and the storage capacity of the battery. Note that the shelf life of a lithium battery is ultimately 10 years.

The battery-backup circuit serves two purposes:

- It reduces the battery voltage to the real-time clock, thereby reducing the current consumed by the real-time clock and lengthening the battery life.
- It ensures that current can flow only ***out*** of the battery to prevent charging the battery.

Figure 2–6 shows the RabbitCore 2000 battery backup circuit.

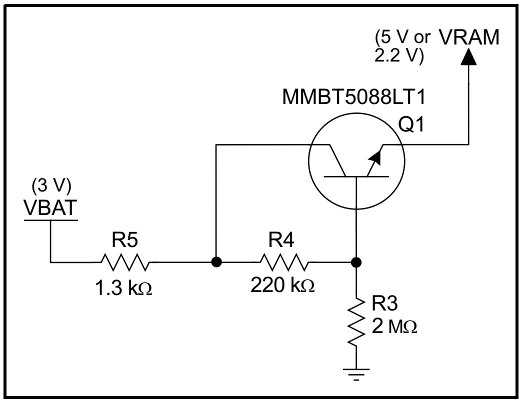


Figure 2–6: RabbitCore RCM2200 Battery Backup Circuit

It is important not to charge a lithium battery with any appreciable current. Resistor R5, shown in Figure 2–6, prevents any catastrophic failure of Q1 by limiting current to the customer-supplied battery. VRAM and Vcc are nearly equal (<100 mV, typically 10 mV) when power is supplied to the RabbitCore RCM2200.

Resistors R3 and R4 make up a voltage divider that biases the base of Q1 to about 0.9 × VBAT. V_{BE} on Q1 is about 0.55 V. Therefore, VRAM is about 0.9 × VBAT - 0.55 V, or about 2.15 V for a 3 V battery.

2.3.2.1 Power to VRAM Switch

The VRAM switch, shown in Figure 2–7, allows a customer-supplied external battery to provide power when the external power goes off. The switch provides an isolation between Vcc and the battery when Vcc goes low. This prevents the Vcc line from draining the battery.

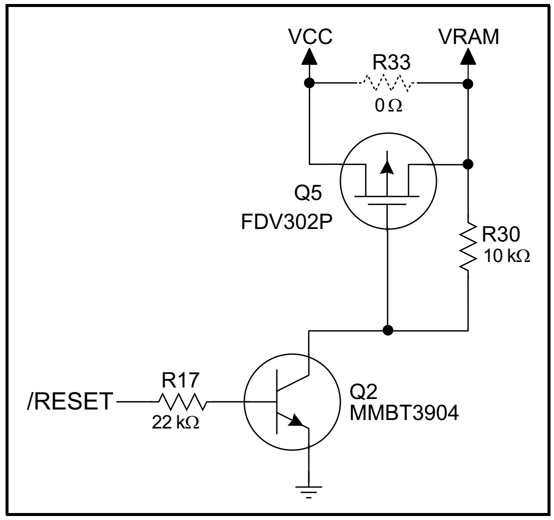


Figure 2–7: VRAM Switch

Transistor Q5 is needed to ensure a very small voltage drop between Vcc and VRAM (<100 mV, typically 10 mV) so that the processor lines powered by Vcc will not have a significantly different voltage than VRAM.

When the RCM2200 is not resetting (pin 2 on U1 is high), the /RESET line will be high. This turns on Q2, causing its collector to go low. This turns on Q5, allowing VRAM to nearly equal Vcc.

When the RCM2200 is resetting, the /RESET line will go low. This turns off Q2 and Q5, providing an isolation between Vcc and VRAM.

The battery backup circuit keeps VRAM from dropping below 2 V.

2.3.2.2 Reset Generator

The RabbitCore RCM2200 uses a reset generator, U1, to reset the Rabbit 2000 microprocessor when the voltage drops below the voltage necessary for reliable operation. The reset occurs between 4.50 V and 4.75 V, typically 4.63 V.

The RCM2200 has a reset output, pin 9 on header J5. The reset output may be overdriven with at least 5 mA, essentially turning pin 9 on header J5 into a reset input, which may be used to reset the RCM2200.

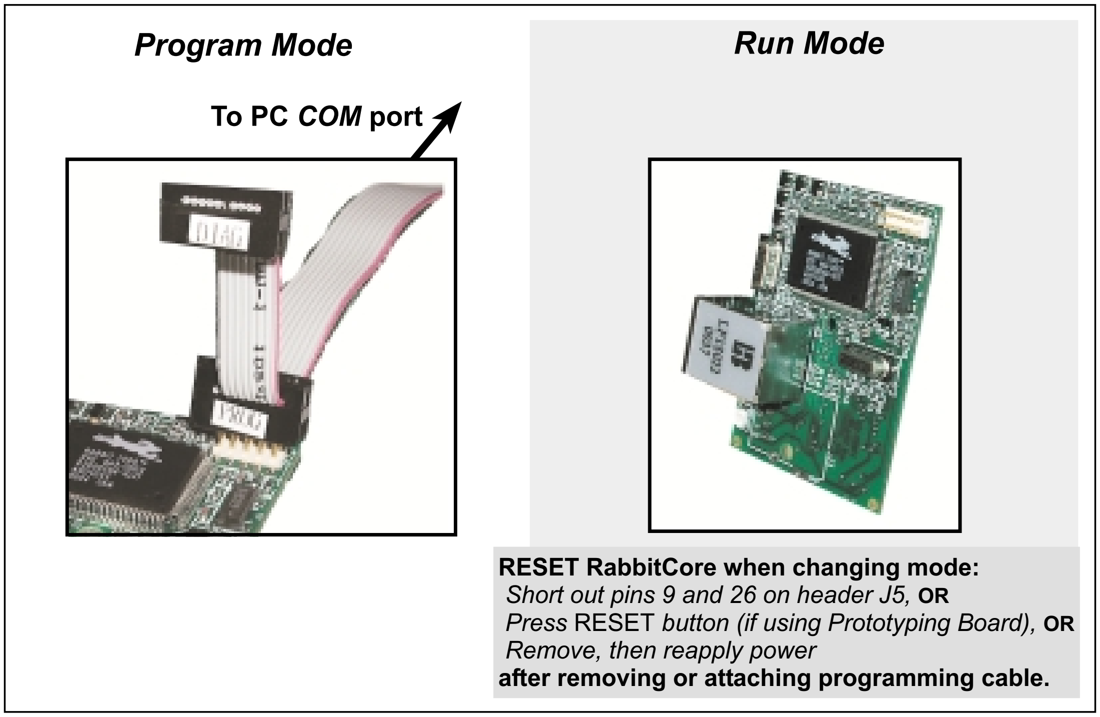


Figure 2–8: Switching Between Program Mode and Run Mode

2.4 Programming Cable

The RabbitCore RCM2200 is automatically in program mode when the **PROG** connector on the programming cable is attached, and is automatically in run mode when no programming cable is attached.

The **DIAG** connector of the programming cable may be used on header J5 of the RabbitCore RCM2200 with the board operating in the run mode. This allows the programming port to be used as an application port. See Appendix C, “Programming Cable,” for more information.

2.4.1 Changing from Program Mode to Run Mode

1. Disconnect the programming cable from header J5 of the RabbitCore RCM2200.
2. Reset the RabbitCore RCM2200. You may do this as explained in Figure 2–8.

The RabbitCore RCM2200 is now ready to operate in the run mode.

2.4.2 Changing from Run Mode to Program Mode

1. Attach the programming cable to header J3 on the RabbitCore RCM2200 series.
2. Reset the RabbitCore RCM2200 series. You may do this as explained in Figure 2–8.

The RabbitCore RCM2200 is now ready to operate in the program mode.

Software Reference 3

Dynamic C Premier is an integrated development system for writing embedded software. It runs on an IBM-compatible PC and is designed for use with Z-World controllers and other controllers based on the Rabbit microprocessor. Chapter 3 provides the libraries, function calls, and sample programs related to the RabbitCore RCM2200.

3.1 More About Dynamic C

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

Dynamic C for Rabbit 2000™ processors uses the standard Rabbit programming interface. This is a 10-pin connector that connects to the Rabbit 2000 serial port A. It is possible to reset and cold-boot a Rabbit processor via the programming port. No software needs to be present in the target system. More details are available in the ***Rabbit 2000 Microprocessor User's Manual***.

Dynamic C cold-boots the target system and compiles the BIOS. The BIOS is a basic program of a few thousand bytes in length that provides the debugging and communication facilities that Dynamic C needs. Once the BIOS has been compiled, the user can compile his own program and test it. If the user program stops running, a new cold boot and BIOS compile can be done at any time.

Dynamic C does not use **`include`** files, rather it has libraries that are used for the same purpose, that is, to supply functions and function prototypes to programs before they are compiled.

Dynamic C supports assembly language, either as separate functions or as fragments embedded in C programs. Interrupt routines may be written in Dynamic C or in assembly language.

3.1.1 Operating System Framework

Dynamic C does not include an operating system in the usual sense of a complex software system that is resident in memory. The user has complete control of what is loaded as a part of his program, other than those routines that support loading and debugging (which are inactive at embedded run time). However, certain routines are very basic and normally should always be present and active.

- Periodic interrupt routine. This interrupt routine is driven by the Rabbit periodic interrupt facility, and when enabled creates an interrupt every 16 ticks of the 32.768 kHz oscillator, or every 488 µs. This routine drives three long global variables that keep track of the time: **`SEC_TIMER`**, **`MS_TIMER`**, and **`TICK_TIMER`** that respectively count seconds, milliseconds, and 488 µs ticks. These variables are needed by some functions that measure time. The **`SEC_TIMER`** is set to seconds elapsed since 1 Jan 1980, and thus also keeps track of the time and date. The periodic interrupt routine must be disabled when the microprocessor enters sleepy mode and the processor clock is operating at 32.768 kHz. The interrupt routine cannot complete at this slow speed before the next tick of the periodic interrupt. In this situation, the hardware real-time clock can be read directly to provide the time.

  The periodic interrupt function also hits the hardware watchdog timer. Software or "virtual" watchdog timers are available in Dynamic C. See the ***Dynamic C Premier User's Manual*** for more information.

3.1.2 Using Dynamic C

You have a choice of doing your software development in the flash memory or in the static RAM. There are 256K bytes of flash memory and 128K bytes of SRAM. The advantage of working in RAM is to save wear on the flash, which is limited to about 100,000 writes.

> **NOTE:** Note that an application can be developed in RAM, but cannot run standalone from RAM after the programming cable is disconnected. All applications can only run from flash memory.

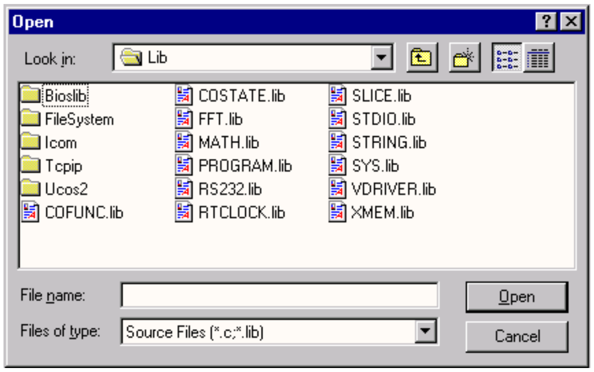
> **NOTE:** Do not depend on the flash memory sector size or type. Due to the volatility of the flash memory market, the RabbitCore RCM2200 and Dynamic C were designed to accommodate flash devices with various sector sizes.

The disadvantage of using flash memory when debugging a program is that interrupts must be disabled for approximately 5 ms to 20 ms whenever a break point is set in the program. This can crash fast interrupt routines that are running while you stop at a break point or single-step the program.

Flash memory or RAM is selected with the Dynamic C **Options** > **Compiler** menu.

3.2 Dynamic C Libraries

With Dynamic C running, click **File > Open**, and select **Lib**. The following list of Dynamic C libraries will be displayed.



There is no unique library that is specific to the RabbitCore RCM2200. The functions in the above libraries are described in the ***Dynamic C Premier User's Manual***.

3.2.1 I/O

The RabbitCore RCM2200 was designed to interface with other systems, and so there are no drivers written specifically for the I/O. The general Dynamic C read and write functions allow you to customize the parallel I/O to meet your specific needs. For example, use

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

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

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

to set all the port E bits as outputs.

The sample programs in the Dynamic C **SAMPLES/RCM2200** directory provide further examples.

3.2.2 Serial Communication Drivers

The Prototyping Board has room for an RS-232 chip for which the Rabbit serial library, **RS232.LIB**, provides a set of functions that send and receive entire blocks of data without yielding to other tasks, a set of single-user cofunctions that send and receive data but yield to other tasks, and a set of circular buffer functions.

The naming convention is **serXfn**:

ser—serial

X—the port being used: A, B, C, or D

fn - the function being implemented

For example, **serBgetc()** is the serial port B function **getc()**, which returns a character.

The Rabbit serial functions are listed in the following groups.

Open and Close Functions

Non-Cofunction Blocking Input Functions

Non-Cofunction Blocking Output Functions

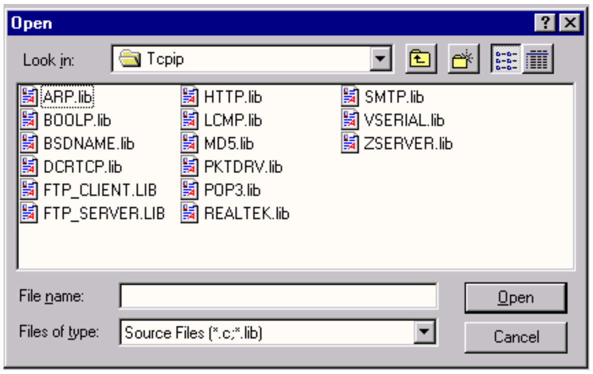
Single-User Cofunction Input Functions

Single-User Cofunction Output Functions

Circular Buffer Functions

3.2.3 TCP/IP Drivers

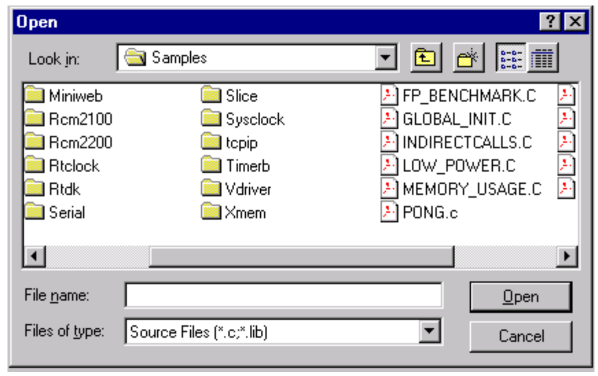
The TCP/IP drivers are located in the **TCPIP** directory.



Complete information on these libraries and the TCP/IP functions is provided in the Dynamic ***C Premier TCP/IP Function Reference Manual***.

3.3 Sample Programs

Sample programs are provided in the Dynamic C **Samples** folder, which is shown below.



The various folders contain specific sample programs that illustrate the use of the corresponding Dynamic C libraries. For example, the sample program **PONG.C** demonstrates the output to the Dynamic C **STDIO** window.

Two folders contain sample programs that illustrate features unique to the RabbitCore RCM2200.

- **RCM2200**—Demonstrates the basic operation and the Ethernet functionality of the RabbitCore RCM2200.
- **TCPIP**—Demonstrates more advanced TCP/IP programming for Z-World's Ethernet-enabled Rabbit-based boards.

Follow the instructions included with the sample program to connect the RabbitCore RCM2200 and the other hardware identified in the instructions.

To run a sample program, open it with the **File** menu (if it is not still open), compile it using the **Compile** menu, and then run it by selecting **Run** in the **Run** menu. The RabbitCore RCM2200 must be in Program Mode (see Section 2.4, "Programming Cable") and must be connected to a PC using the programming cable.

More complete information on Dynamic C is provided in the ***Dynamic C Premier User's Manual***.

3.4 Upgrading Dynamic C

Dynamic C patches that focus on bug fixes are available from time to time. Check the Web sites

- www.zworld.com/support/support-center.html

or

- www.rabbitsemiconductor.com/support.html

for the latest patches, workarounds, and bug fixes.

The default installation of a patch or bug fix is to install the file in a directory (folder) different from that of the original Dynamic C installation. Z-World recommends using a different directory so that you can verify the operation of the patch without overwriting the existing Dynamic C installation. If you have made any changes to the BIOS or to libraries, or if you have programs in the old directory (folder), make these same changes to the BIOS or libraries in the new directory containing the patch. Do ***not*** simply copy over an entire file since you may overwrite a bug fix; of course, you may copy over any programs you have written. Once you are sure the new patch works entirely to your satisfaction, you may retire the existing installation, but keep it available to handle legacy applications.

3.4.1 Upgrades

A special edition of Dynamic C, Dynamic C SE, is included on the CD that comes with the RabbitCore RCM2200 Development Kit, and has been customized with all the libraries and features needed to develop and run an application on the RabbitCore RCM2200.

More advanced users who may need upgrades and additional capabilities for other Z-World products in the future are encouraged to consider the standard edition of Dynamic C Premier, which Z-World plans to fully supported with upgrades now and into the future.

RabbitCore RCM2200 Specifications A

Appendix A provides the specifications for the RabbitCore RCM2200, and describes the conformal coating.

A.1 Electrical and Mechanical Characteristics

Figure A−1 shows the mechanical dimensions for the RabbitCore RCM2200.

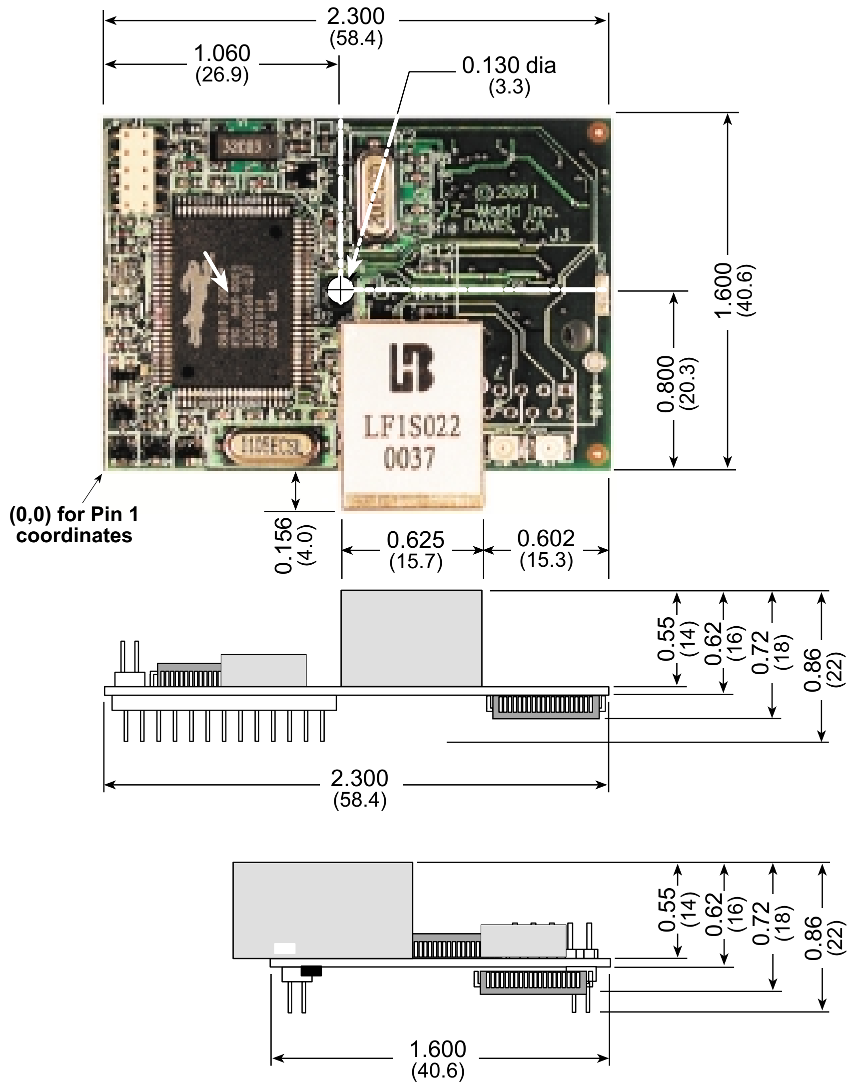


Figure A−1: RabbitCore RCM2200 Dimensions

Table A–1 provides the pin 1 locations for the RabbitCore RCM2200 headers viewed from the top side (as in Figure A–1).

Table A–1: RabbitCore RCM2200 Header Pin 1 Locations

Header	Description	Pin 1 (x,y) Coordinates (Inches)
J4	RabbitCore RCM2200 user board interface	(0.100, 1.445)
J5	RabbitCore RCM2200 user board interface	(0.100, 0.195)
J1	Programming header (top side)	(0.125, 1.515)
DS1	LNK LED	(1.815, 0.105)
DS2	ACT LED	(2.015, 0.105)

Table A–2 lists the electrical, mechanical, and environmental specifications for the RabbitCore RCM2200.

Table A–2: RabbitCore RCM2200 Specifications

Parameter	Specification
Board Size	1.60" × 2.30" × 0.86" (41 mm × 59 mm × 22 mm)
Operating Temperature	–40°C to +70°C
Humidity	5% to 95%, noncondensing
Input Voltage	4.75 V to 5.25 V DC
Current	134 mA at 22.1 MHz, 5 V DC; 10 mA additional with programming cable attached
General-Purpose I/O	26 parallel I/0 lines grouped in five 8-bit ports (shared with serial ports): 16 configurable for I/O, 7 fixed inputs, 3 fixed outputs
Memory, I/O Interface	4 address lines, 8 data lines, I/O read/write
Additional Digital Inputs	Startup mode (2), reset in, Serial Port A (1)
Additional Digital Outputs	Status, reset out, Serial Port A (1)
Ethernet Interface	10base-T
Microprocessor	Rabbit 2000™
Clock	22.1 MHz
SRAM	128K × 8, surface mount
Flash Memory	One 256K × 8, surface mount
Timers	Five 8-bit timers cascadable in pairs, one 10-bit timer with 2 match registers that each have an interrupt
Serial Ports	Three CMOS-compatible ports. One port is configurable as a clocked port, a fourth clocked pin is available on the programming port.
Serial Rate	CMOS: maximum asynchronous 345,600 bps maximum synchronous 138,240 bps
Slave Interface	A slave port allows the RabbitCore RCM2200 to be used as an intelligent peripheral device slaved to a master processor, which may either be another Rabbit 2000 or any other type of processor
Watchdog/Supervisor	Yes
Time/Date Clock	Yes
Socket Strip (for connection to headers J4 and J5)	2x13, 2 mm pitch
Backup Battery	Provision for user-supplied backup battery (2.85 V to 3.15 V) via connections on header J5

A.1.1 Headers

The RabbitCore 2000 uses headers at J4 and J5 for physical connection to other boards. J4 and J5 are 2 × 13 SMT headers with a 2 mm pin spacing. J1, the programming port, is a 2 × 5 header with a 2 mm pin spacing.

Figure A–2 shows the layout of another board for the RabbitCore RCM2200 to be plugged into. These values are relative to the header connectors.

A.1.2 Physical Mounting

A 9/32” (7 mm) standoff with a 6-32 screw is recommended to attach RabbitCore RCM2200 to a user board at the hole position shown in Figure A–2.

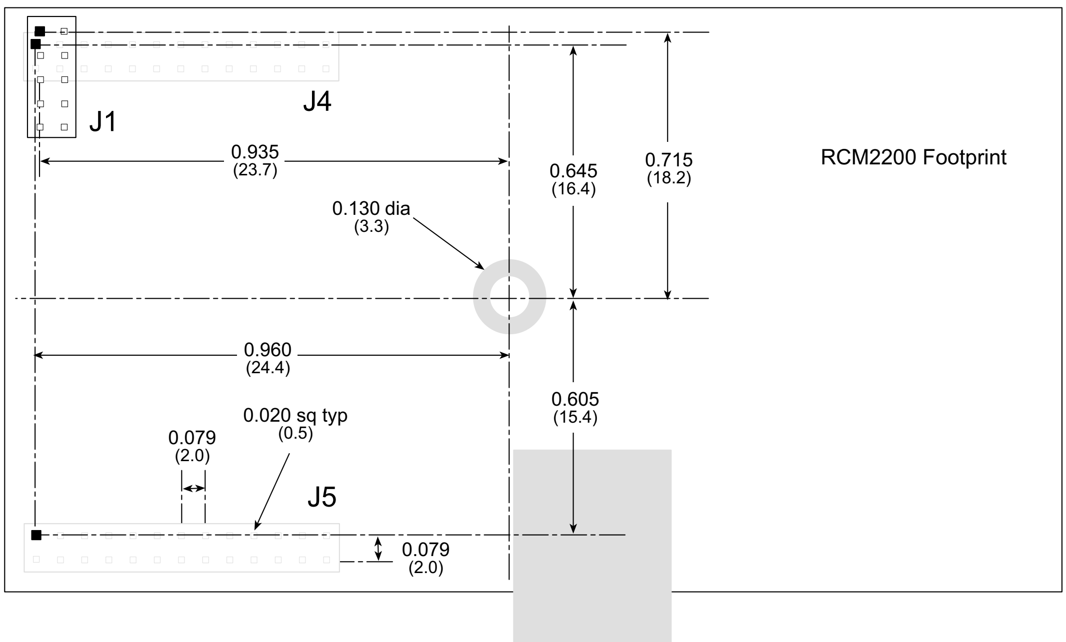


Figure A–2: User Board Footprint for RabbitCore RCM2200

A.2 Bus Loading

You must pay careful attention to bus loading when designing an interface to the RabbitCore RCM2200.

This section provides bus loading information for external devices.

Table A–3 lists the capacitance for the various RabbitCore 2000 I/O ports.

Table A–3: Capacitance of Rabbit 2000 I/O Ports

I/O Ports	Input Capacitance (pF)	Output Capacitance (pF)
Parallel Ports A to E	12	14
Data Lines BD0–BD7	10	12
Address Lines BA0–BA12	4	8

Figure A−3 shows a typical timing diagram for the Rabbit 2000 microprocessor external memory read and write cycles.

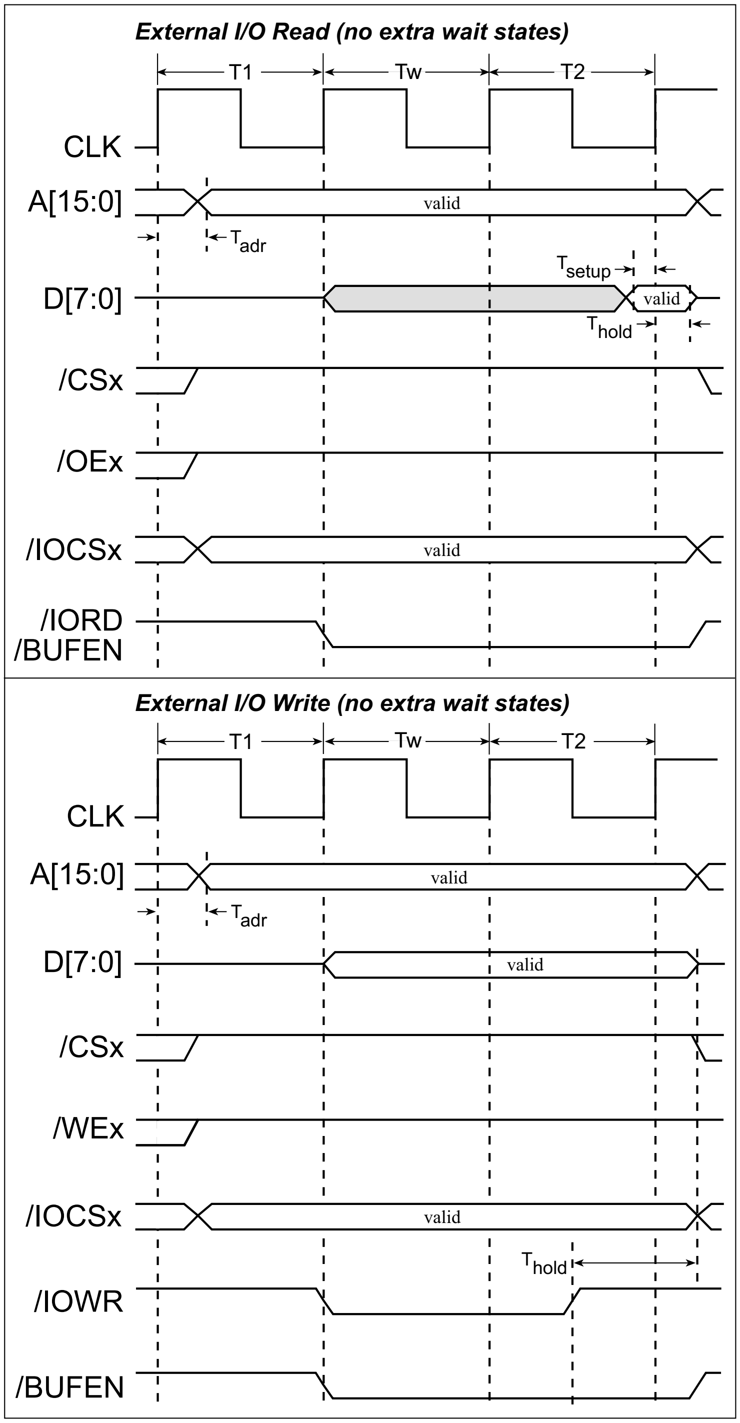


Figure A−3: Memory Read and Write Cycles

T_{adr} is the time required for the address output to reach 0.8 V. This time depends on the bus loading. T_{setup} is the data setup time relative to the clock. Tsetup is specified from 30%/70% of the V_{DD} voltage level.

Table A–4 lists the parameters shown in these figures and provides minimum or measured values.

Table A–4: Memory and External I/O Read/Write Parameters

Parameter		Description	Value	
Read Parameters	T_{adr}	Time from CPU clock rising edge to address valid	Max.	7 ns @ 20 pF, 5 V (10 ns @ 3.3 V) 14 ns @ 70 pF, 5 V (19 ns @ 3.3 V)
	T_{setup}	Data read setup time	Min.	2 ns @ 5 V (3 ns @ 3.3 V)
	T_{hold}	Data read hold time	Min.	0 ns
Write Parameters	T_{adr}	Time from CPU clock rising edge to address valid	Max.	7 ns @ 20 pF, 5 V (10 ns @ 3.3 V) 14 ns @ 70 pF, 5 V (19 ns @ 3.3 V)
	T_{hold}	Data write hold time from /WEx or /IOWR	Min.	½ CPU clock cycle

A.3 Rabbit 2000 DC Characteristics

Table A–5 outlines the DC characteristics for the Rabbit 2000 at 5.0 V over the recommended operating temperature range from T_a = –40°C to +85°C, V_{DD} = 4.5 V to 5.5 V.

Table A–5: 5.0 Volt DC Characteristics

Symbol	Parameter	Test Conditions	Min	Typ	Max	Units
I_{IH}	Input Leakage High	$V_{IN} = V_{DD}$, V_{DD} = 5.5 V			10	µA
I_{IL}	Input Leakage Low (no pull-up)	$V_{IN} = V_{SS}$, V_{DD} = 5.5 V	-10			µA
I_{OZ}	Output Leakage (no pull-up)	$V_{IN} = V_{DD}$ or V_{SS}, V_{DD} = 5.5 V	-10		10	µA
V_{IL}	CMOS Input Low Voltage				0.3 x V_{DD}	V
V_{IH}	CMOS Input High Voltage		0.7 x V_{DD}			V
V_T	CMOS Switching Threshold	V_{DD} = 5.0 V, 25°C		2.4		V
V_{OL}	CMOS Output Low Voltage	I_{OL} = See Table A–6 (sinking) V_{DD} = 4.5 V		0.2	0.4	V
V_{OH}	CMOS Output High Voltage	I_{OH} = See Table A–6 (sourcing) V_{DD} = 4.5 V	0.7 x V_{DD}	4.2		V

A.4 I/O Buffer Sourcing and Sinking Limit

Unless otherwise specified, the Rabbit I/O buffers are capable of sourcing and sinking 8 mA of current per pin at full AC switching speed. Full AC switching assumes a 25.8 MHz CPU clock and capacitive loading on address and data lines of less than 100 pF per pin. Address pin A0 and data pin D0 are rated at 16 mA each. Pins A1–A12 and D1–D7 are each rated at 8 mA. The absolute maximum operating voltage on all I/O is V_{DD} + 0.5 V, or 5.5 V.

Table A–6 shows the AC and DC output drive limits of the parallel I/O buffers when the Rabbit 2000 is used in the RabbitCore RCM2200.

Table A–6: I/O Buffer Sourcing and Sinking Capability

Pin Name	**Output Drive Sourcing*/Sinking† Limits (mA)**	
Output Port Name	Full AC Switching SRC/SNK	Maximum‡ DC Output Drive SRC/SNK
PA [7:0]	8/8	12/12
PB [7, 1, 0]	8/8	12/12
PC [6, 2, 0]	8/8	12/12
PD [5::4]	8/8	12/12
PD3**	16/16	25/25
PE [7, 5, 4, 1, 0]	8/8	12/12

* The maximum DC sourcing current for I/O buffers between V_{DD} pins is 112 mA.

† The maximum DC sinking current for I/O buffers between V_{SS} pins is 150 mA.

‡ The maximum DC output drive on I/O buffers must be adjusted to take into consideration the current demands made my AC switching outputs, capacitive loading on switching outputs, and switching voltage.

The current drawn by all switching and nonswitching I/O must not exceed the limits specified in the first two footnotes.

** The combined sourcing from Port D [7:0] may need to be adjusted so as not to exceed the 112 mA sourcing limit requirement specified in the first footnote.

A.5 Conformal Coating

The areas around the crystal oscillator has had the Dow Corning silicone-based 1-2620 conformal coating applied. The conformally coated area is shown in Figure A–4. The conformal coating protects these high-impedance circuits from the effects of moisture and contaminants over time.

Any components in the conformally coated area may be replaced using standard soldering procedures for surface-mounted components. A new conformal coating should then be applied to offer continuing protection against the effects of moisture and contaminants.

> **NOTE:** For more information on conformal coatings, refer to Rabbit Semiconductor Technical Note 303, ***Conformal Coatings***.

Figure A–4: RabbitCore RCM2200 Areas Receiving Conformal Coating

Power Supply B

Appendix B provides information on the current requirements of the RabbitCore RCM2200, and some background on the chip select circuit used in power management.

B.1 Power Supplies

The RabbitCore RCM2200 requires a regulated 5 V ± 0.25 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 RabbitCore board through headers J4 and J5.

A RabbitCore RCM2200 with no loading at the outputs operating at 22.1 MHz typically draws 134 mA. The RabbitCore RCM2200 will consume an additional 10 mA when the programming cable is used to connect J1 to a PC.

B.2 Battery Backup Circuits

As explained in Section 2.3.2, the RabbitCore RCM2200 has provision for battery backup, which kicks in to keep VRAM from dropping below 2 V.

The current drain on the battery in a battery-backed circuit must be kept to a minimum. When the RabbitCore RCM2200 is not powered, the battery keeps the SRAM memory contents and the real-time clock (RTC) going. The SRAM has a powerdown mode that greatly reduces power consumption. This powerdown mode is activated by raising the chip select (CS) signal line. Normally the SRAM requires Vcc to operate. However, only 2 V is required for data retention in powerdown mode. Thus, when power is removed from the circuit, the battery voltage needs to be provided to both the SRAM power pin and to the CS signal line. The CS control switch accomplishes this task for the CS signal line.

Figure B–1 shows a schematic of the chip select control switch.

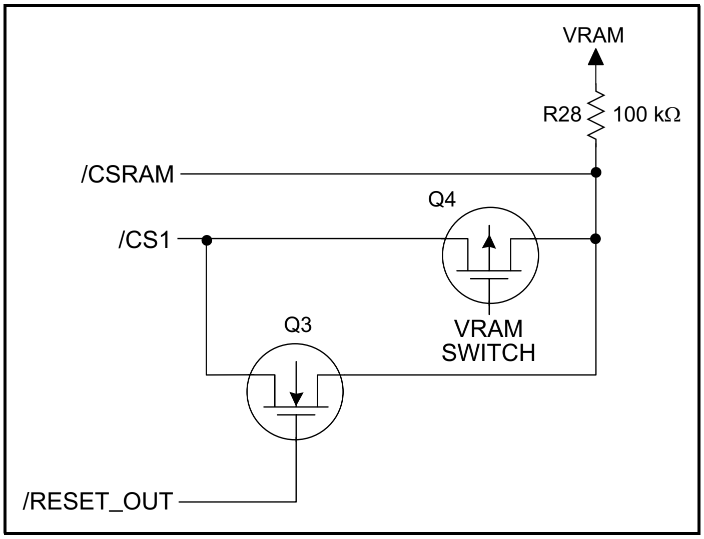


Figure B-1: Chip Select Control Switch

In a powered-up condition, the CS control switch must allow the processor's chip select signal /CS1 to control the SRAM's CS signal /CSRAM. So, with power applied, /CSRAM must be the same signal as /CS1, and with power removed, /CSRAM must be held high (but only needs to be as high as the battery voltage). Q3 and Q4 are MOSFET transistors with opposing polarity. They are both turned on when power is applied to the circuit. They allow the CS signal to pass from the processor to the SRAM so that the processor can periodically access the SRAM. When power is removed from the circuit, the transistors will turn off and isolate /CSRAM from the processor. The isolated /CSRAM line has a 100 kΩ pullup resistor to VRAM (R28). This pullup resistor keeps /CSRAM at the VRAM voltage level (which under no power condition is the backup battery's regulated voltage at a little more than 2 V).

Transistors Q3 and Q4 are of opposite polarity so that a rail-to-rail voltages can be passed. When the /CS1 voltage is low, Q3 will conduct. When the /CS1 voltage is high, Q4 will conduct. It takes time for the transistors to turn on, creating a propagation delay. This delay is typically very small, about 10 ns to 15ns.

Programming Cable C

Appendix C provides additional theoretical information for the Rabbit 2000™ microprocessor when using the **DIAG** and **PROG** connectors on the programming cable. The **PROG** connector is used only when the programming cable is attached to the programming connector (header J5) while a new application is being developed. Otherwise, the **DIAG** connector on the programming cable allows the programming cable to be used as an RS-232 to CMOS level converter for serial communication, which is appropriate for monitoring or debugging a RabbitCore system while it is running.

The programming port, which is shown in Figure C−1, can serve as a convenient communications port for field setup or other occasional communication need (for example, as a diagnostic port). There are several ways that the port can be automatically integrated into software. If the port is simply to perform a setup function, that is, write setup information to flash memory, then the controller can be reset through the programming port and a cold boot performed to start execution of a special program dedicated to this functionality.

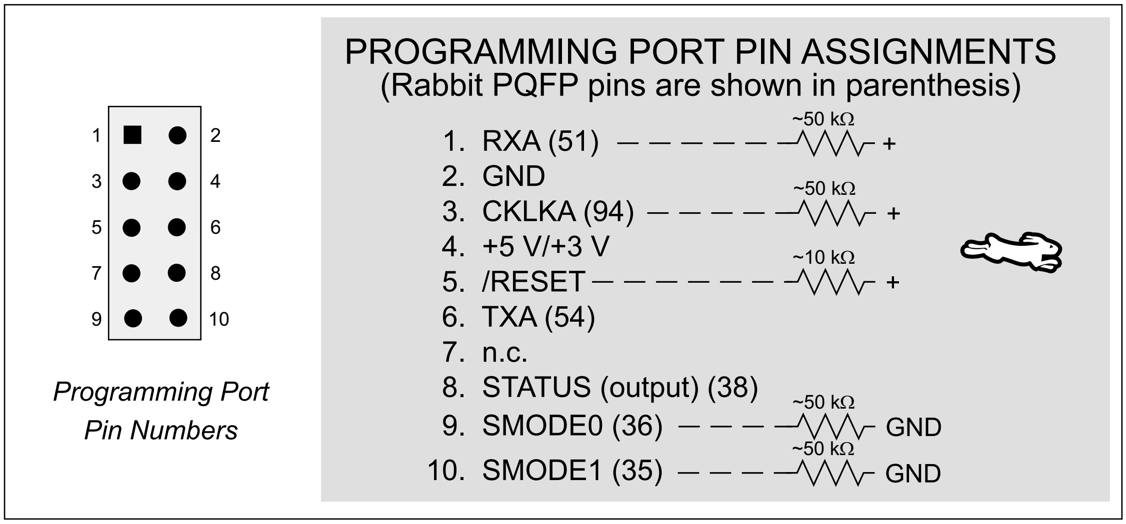


Figure C−1: Programming Port Pin Assignments

When the **PROG** connector is used, the /RESET line can be asserted by manipulating DTR and the STATUS line can be read as DSR on the serial port. The target can be restarted by pulsing reset and then, after a short delay, sending a special character string at 2400 bps. To simply restart the BIOS, the string 80h, 24h, 80h can be sent. When the BIOS is started, it can tell whether the programming cable is connected because the SMODE1 and SMODE0 pins are sensed as being high. This will cause the Rabbit 2000 to enter the bootstrap mode. The Dynamic C programming mode then can have an escape message that will enable the diagnostic serial port function.

Alternatively, the **DIAG** connector can be used to connect the programming port. The /RESET line and the SMODE1 and SMODE0 pins are not connected to this connector. The programming port is then enabled as a diagnostic port by polling the port periodically to see if communication needs to begin or to enable the port and wait for interrupts. The pull-up resistors on RXA and CLKA prevent spurious data reception that might take place if the pins floated.

If the clocked serial mode is used, the serial port can be driven by having two toggling lines that can be driven and one line that can be sensed. This allows a conversation with a device that does not have an asynchronous serial port but that has two output signal lines and one input signal line.

The line TXA (also called PC6) is zero after reset if the cold-boot mode is not enabled. A possible way to detect the presence of a cable on the programming port is for the cable to connect TXA to one of the SMODE pins and then test for the connection by raising PC6 and reading the SMODE pin after the cold-boot mode has been disabled.

Once you establish that the programming port will never again be needed for programming, it is possible to use the programming port for additional I/O lines. Table C−1 lists the pins available for this alternate configuration.

Table C−1: RabbitCore RCM2200 Programming Port Pinout Configurations

Pin		Pin Name	Default Use	Alternate Use	Notes
Header J1	1	RXA	Serial Port A	PC6—Input	
	2	GND			
	3	CLKA		PB1—Bitwise or parallel programmable input	
	4	VCC			
	5	RESET			Connected to reset generator U1
	6	TXA	Serial Port A	PC7—Output	
	8	STATUS		Output	
	9	SMODE0		Input	Must be low when RCM2200 boots up
	10	SMODE1		Input	Must be low when RCM2200 boots up

Sample Circuits D

This appendix details several basic sample circuits that can be used with the RabbitCore RCM2200 modules.

- RS-232/RS-485 Serial Communication
- Keypad and LCD Connections
- Keypad and LCD Connections
- D/A Converter

D.1 RS-232/RS-485 Serial Communication

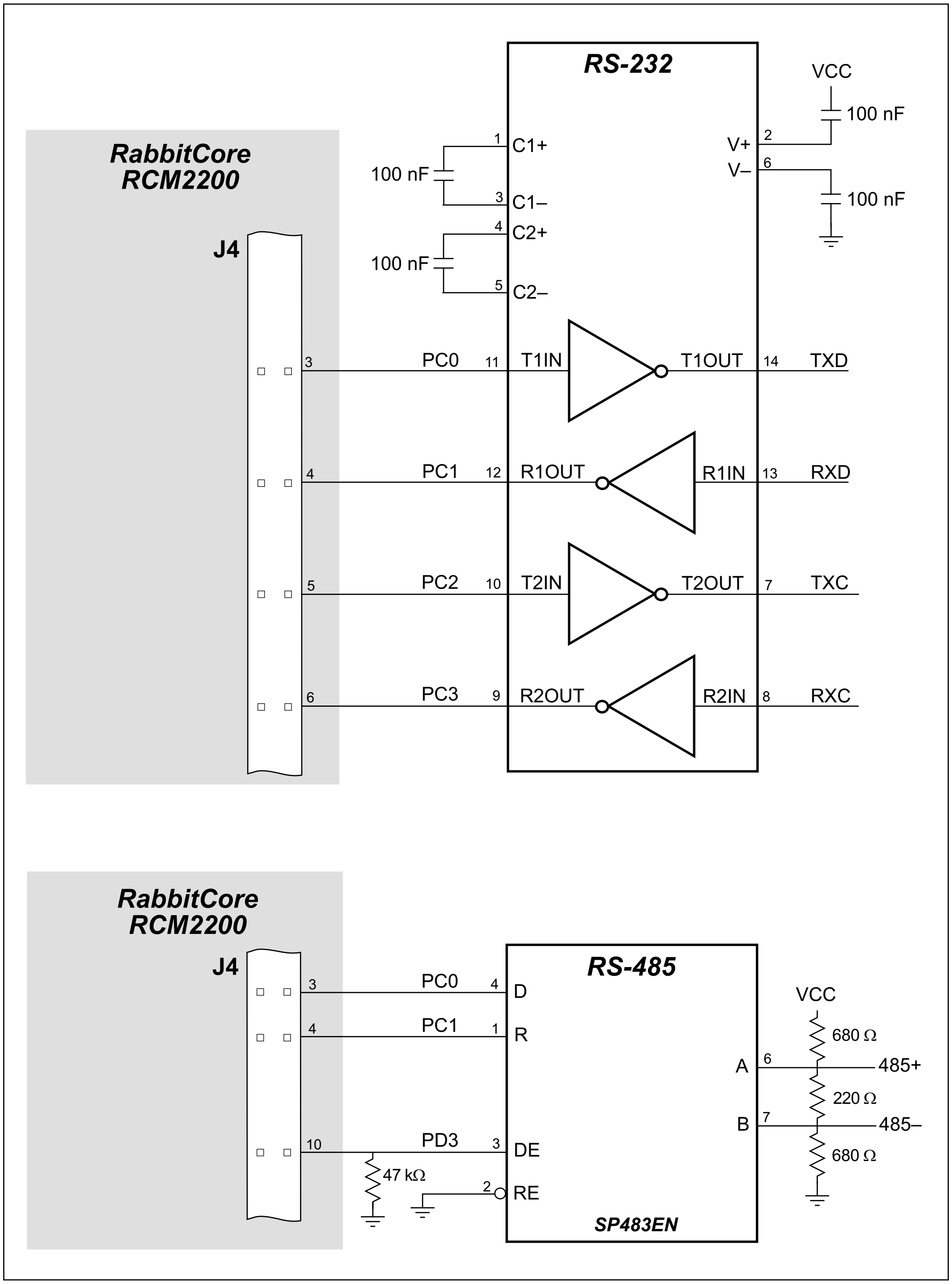


Figure D–1: Sample RS-232 and RS-485 Circuits

Sample Program: **`PUTS.C`** in **`SAMPLES/RCM2200`**.

D.2 Keypad and LCD Connections

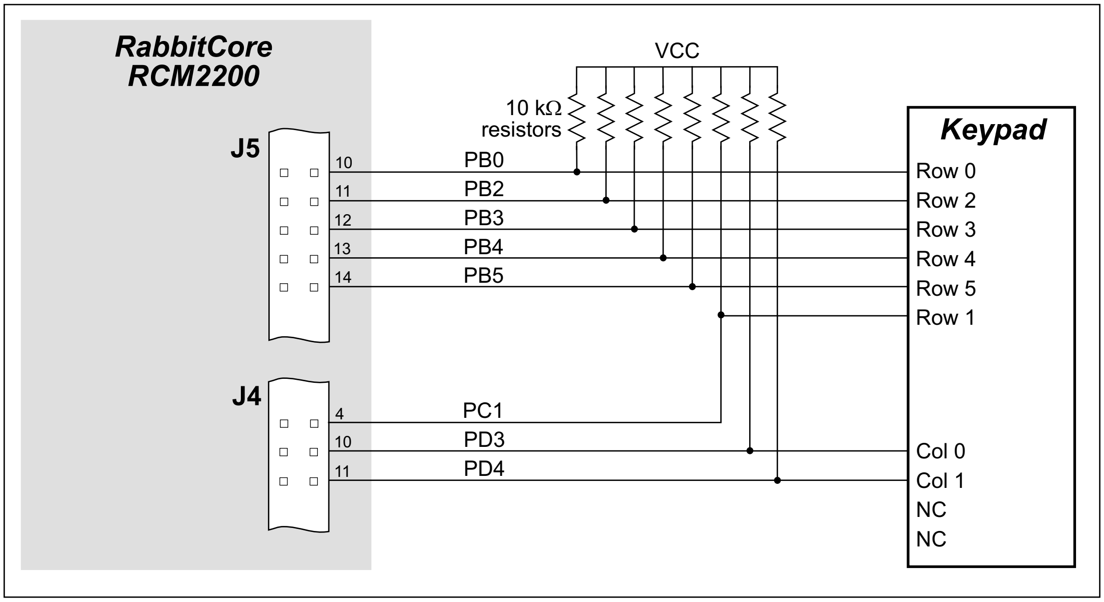


Figure D–2: Sample Keypad Connections

Sample Program: **`KEYLCD.C`** in **`SAMPLES/RCM2200`**.

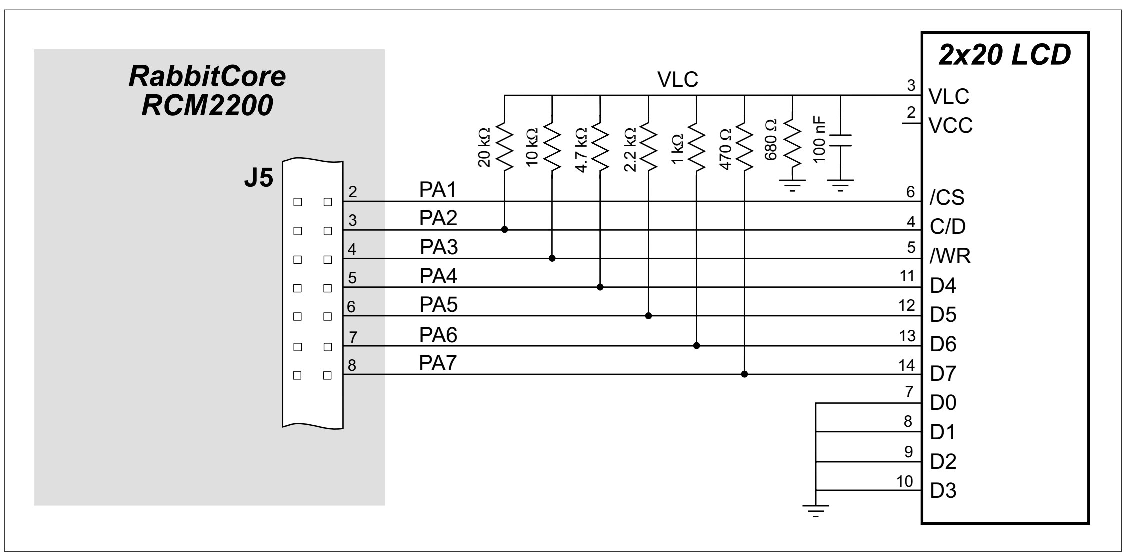


Figure D–3: Sample LCD Connections

Sample Program: **`KEYLCD.C`** in **`SAMPLES/RCM2200`**.

D.3 External Memory

The sample circuit can be used with an external 64K memory device. Larger SRAMs can be written to using this scheme by using other available Rabbit 2000 ports (parallel ports A to E) as address lines.

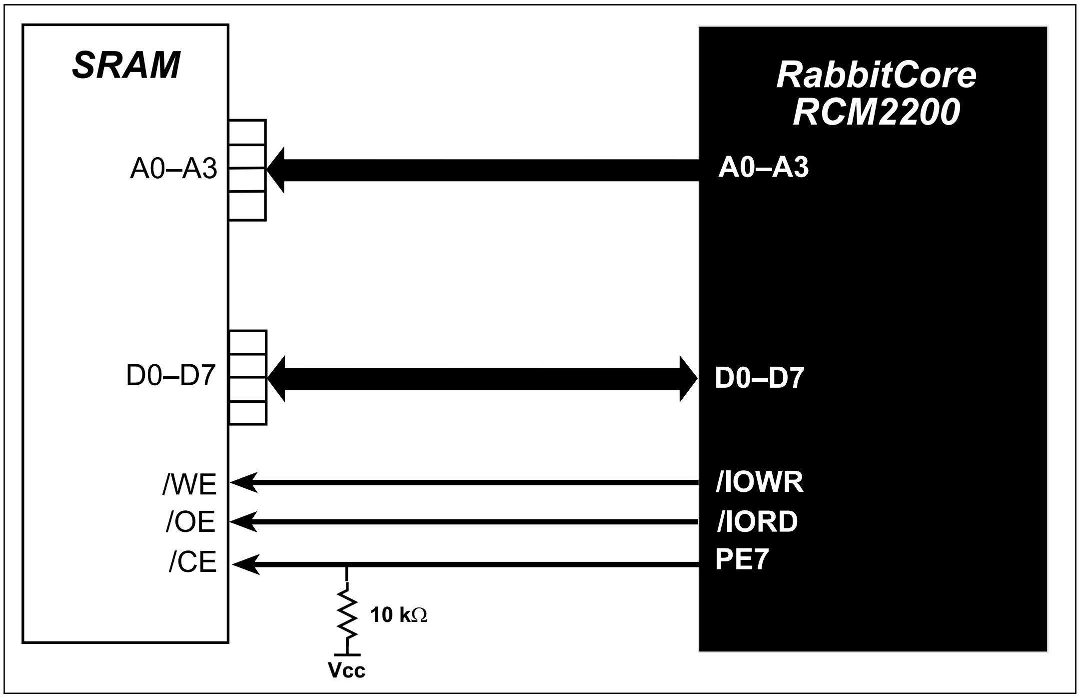


Figure D–4: Sample External Memory Connections

Sample Program: **`EXTSRAM.C`** in **`SAMPLES/RCM2200`**.

D.4 D/A Converter

The output will initially be 0 V to -10.05 V after the first inverting op-amp, and 0 V to +10.05 V after the second inverting op-amp. All lows produce 0 V out, FF produces 10 V out. The output can be scaled by changing the feedback resistors on the op-amps. For example, changing 5.11 kΩ to 2.5 kΩ will produce an output from 0 V to -5 V. Op-amps with a very low input offset voltage are recommended.

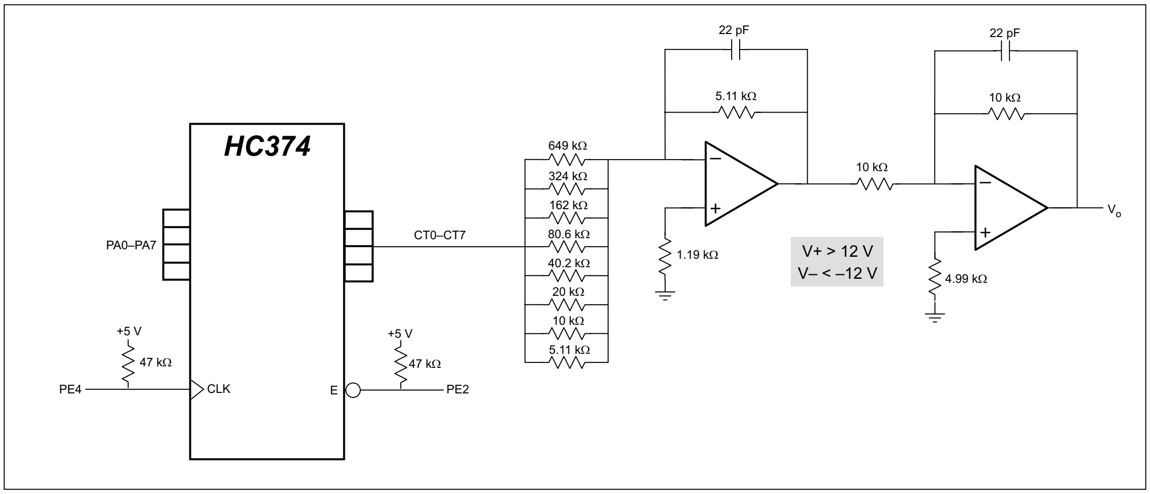


Figure D–5: Sample D/A Converter Connections

External Interrupts E

Appendix E provides information about using the RabbitCore RCM2200 external interrupts.

The Rabbit 2000 microprocessor has four external interrupt inputs on Parallel Port E, which is accessed through pins PE0, PE1, PE4, and PE5 on header J4.

Table E–1 lists the general-purpose Parallel Port E I/O pins that can be used for external interrupts.

Table E–1: Rabbit 2000 Parallel Port E Interrupts

Pin	Default Use	Alternate Use
PE0	General-Purpose I/O	INT0A input
PE1		INT1A input
PE4		INT0B input
PE5		INT1B input

Figure E–1 illustrates these pins.

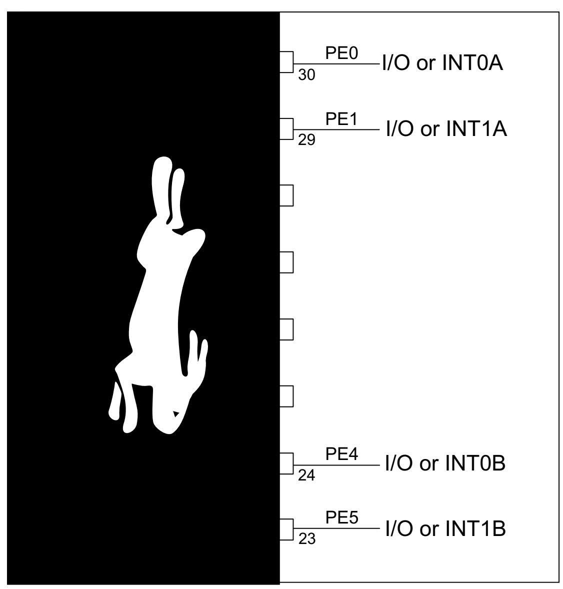


Figure E–1: Rabbit 2000 Interrupt Pins

E.1 Use of External Interrupts

Figure E−2 shows a block diagram of how the Rabbit 2000 external interrupt logic is used in general.

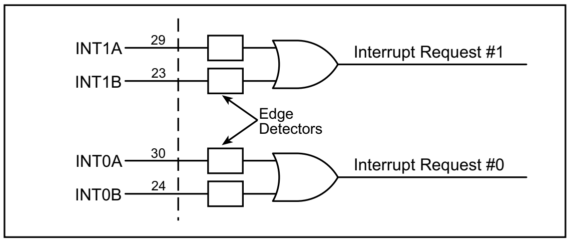


Figure E−2: Rabbit 2000 External Interrupt Logic

Interrupts on the Rabbit 2000 can take place at three priority levels from low to high priority, and are numbered 1, 2 and 3. Each on-chip device, including the two external interrupts, can be assigned a priority at which interrupts will take place. For interrupts that have been assigned the same programmed priority, there is an implicit priority with external interrupt #1 having the highest priority, external interrupt #0 the second highest, and the remaining on-chip devices having lower priorities in the order specified in Section 7.8, “Rabbit Interrupt Structure,” in the ***Rabbit 2000 User’s Manual***.

The two independent interrupts are generated by inputs to the four pins shown in Figure E−2. Each pin is connected to an edge detector that can be configured under program control to detect rising edges, falling edges, or both. These same pins, a part of parallel port E, support alternate functionality as reflected in Table E−1.

When the edge detector detects the rising or falling edge that it is programmed to detect, it sets a flip-flop that drives the output of the edge detector. The flip-flop should be cleared automatically when the interrupt takes place.

Instead, the flip-flop may be cleared spuriously because a different, *lower priority*, interrupt occurs nearly simultaneously (during an 8-clock window) with the occurrence of the edge that sets the flip-flop. This results in a lost interrupt.

Or the flip-flop might not be cleared when the interrupt takes place if a different, *higher priority*, interrupt is being requested nearly simultaneously (during an 8-clock window) with the occurrence of the external interrupt. This results in a spurious interrupt after the first interrupt because the interrupt request was not cleared.

In either case, precautions need to be taken if an interrupt request transitions during a short time period 8 clocks long. These sequences are shown schematically in Figure E−3.

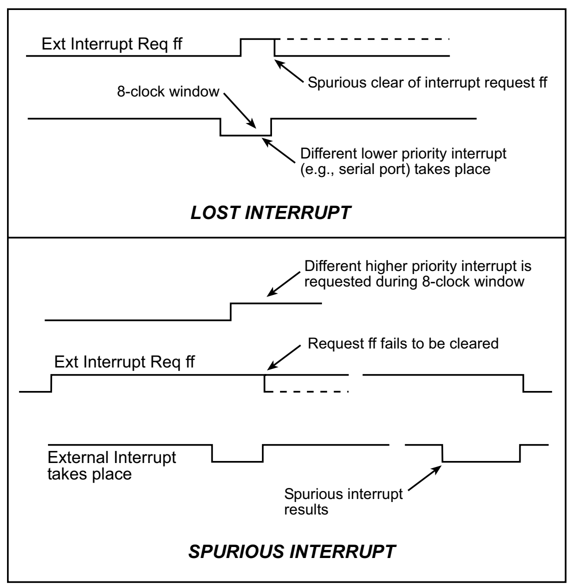


Figure E−3: Interrupt Sequences with Lost or Spurious Interrupts

E.2 Single-Interrupt Request

Tie the inputs for external interrupt #1 and #0 together by adding a 1 kΩ resistor between the two lines. Under this configuration, shown in Figure E–4, both interrupt #1 and #0 will be requested when an edge is detected. The #1 interrupt will take place first since it is of a higher priority.

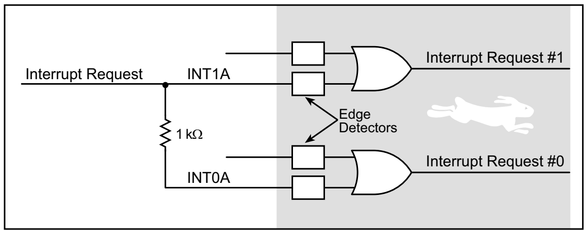


Figure E–4: RabbitCore RCM2200 Configuration for Single-Interrupt Request

The interrupt service routine for interrupt #1 should ignore the interrupt. The actual service routine will be the service routine for interrupt #0. If an interrupt is lost, it will always be #1 and never #0. The 1 kΩ resistor delays the edge slightly so that interrupt #1 is guaranteed to be latched earlier or simultaneously with interrupt #0. It is important that the programmed priority of interrupt #1 be higher than or equal to the programmed priority of interrupt #0. Normally they should be equal.

Spurious interrupts, which occur because of a failure to clear the request latch, are a possibility only if there are other interrupts of higher priority than external interrupt #1 and #0. These can only be the result of programming one of the on-chip peripheral interrupts to have a higher interrupt priority. This could be the case, for example, if the external interrupts are programmed to have priority 1, and one of the serial port interrupts is programmed to have priority 2. Spurious interrupts can always be eliminated by programming both external interrupts to have a priority equal to the highest priority used for another device. The priority can be reduced on entry to the service routine to avoid blocking the true high-priority interrupts. External interrupt #1 cannot cause interrupt #0 to have a spurious interrupt or vice versa. In some cases, spurious interrupts may not disturb function, but the fix is so simple that it is not usually worth the trouble to analyze this possibility.

E.3 OR'ed Interrupt Request

Tie the inputs for external interrupt #1 and #0 together by adding a 1 kΩ resistor. This configuration is shown in Figure E–5.

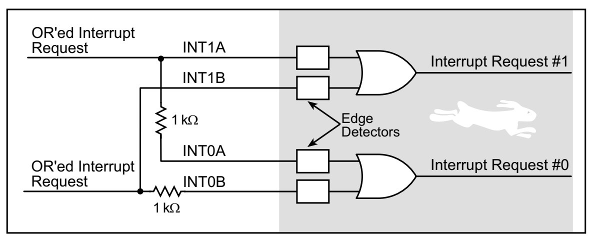


Figure E–5: ***RabbitCore RCM2200 Configuration for OR'ed Interrupt Request***

Index

R

S

Schematics

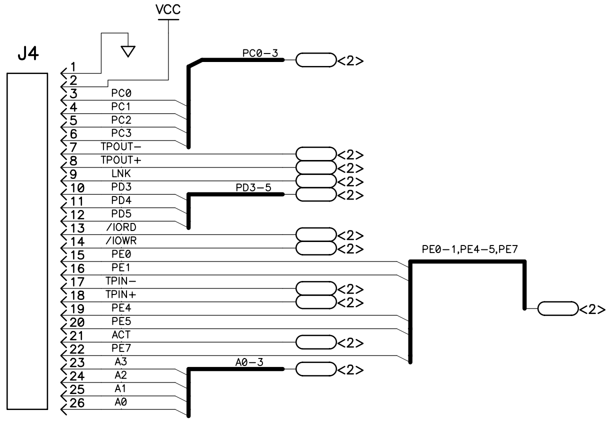
The following schematics are included for user reference:

090–0120	RabbitCore RCM2200 schematic
090–0122	RCM2200 Prototyping Board schematic
090–0085	Programming Cable

REVISION HISTORY

REV	ECO	DESCRIPTION OF CHANGE	PROJECT ENGINEER	APPROVAL DATE	DOCUMENT CONTROL	APPROVAL DATE
A	E11399	INITIAL RELEASE	RJH	N/A	---	---

(REVISION APPROVAL: PROJECT ENGINEER, APPROVAL DATE, DOCUMENT CONTROL, APPROVAL DATE)



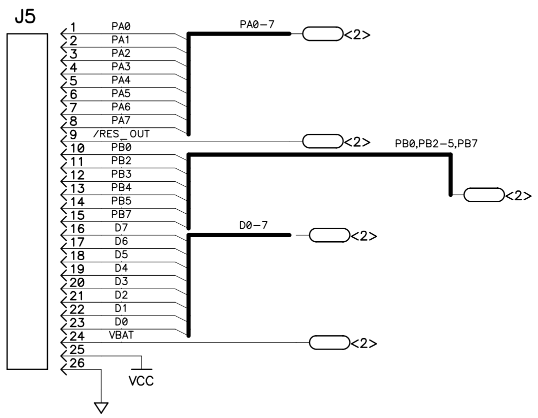
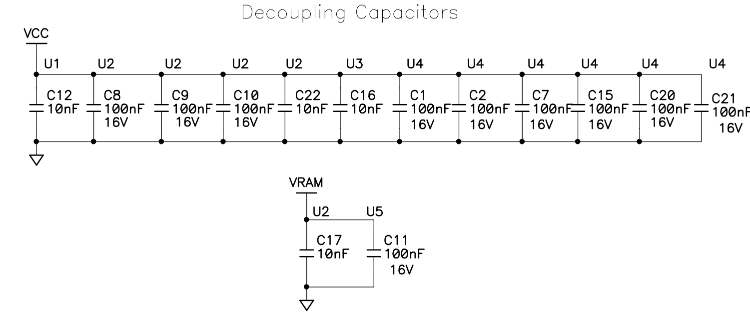


TABLE A

REF DES	DEVICE	DEVICE VOLTAGE INFORMATION	GND	VCC	VRAM	NO CONNECTS	DEVICE: FILTER CAP REF DES(s)
U1	ETC811L						C12
U2	RABBIT 2000		2,27,39 52,77,89	3,28,53, 78,92	42		C8,C10,C9,C22 C17 – PIN42 (VRAM)
U3	FLASH		24	8			C16
U4	RTL8019AS		14,28,44 52,83,86	6,17,47 57,70,89			C1,C2,C7,C15,C20,C21
U5	SRAM 128K X 8		24		8		C11

Decoupling Capacitors



STUFFING TABLE

	CIRCUIT	PART	RCM2200
POWER TO VRAM SWITCH	WITH BATTERY BACKUP CIRCUITRY	R33	NOT INSTALLED
CS CONTROL SWITCH	WITH BATTERY BACKUP CIRCUITRY	R27	NOT INSTALLED
FLASH	MAIN	U3	256K FLASH
	FLASH SELECT	JP1	ZERO ohm ACROSS PINS 1–2
	FLASH TYPE	JP2	ZERO ohm ACROSS PINS 1–2
ETHERNET	RJ–45 CONNECTOR WITH BUILT IN MAGNETICS	J3	NOT INSTALLED
		J2	INSTALLED

NOTES: UNLESS OTHERWISE SPECIFIED;

1. ALL RESISTOR VALUES ARE IN OHMS, 1/16W, 5%
2. ALL CAPACITORS ARE 50VDC OR HIGHER.
3. THE ORIGINATION SOURCE OF A VOLTAGE IS REPRESENTED BY (VCC ▲), AND ALL REFERENCES TO THAT VOLTAGE ARE REPRESENTED BY (VCC).
4. R27, R33, & J3 not normally stuffed.
5. COMPONENT VALUES SHOWN WITH AN ASTERISK (*) FOLLOWING THE VALUE, MAY HAVE DIFFERENT VALUES, OR MAY NOT BE STUFFED DEPENDING ON MODEL. SEE STUFFING CHART FOR CLARIFICATION..
6. JP1 AND JP2 ARE JUMPERED POSITION 1 TO POSITION 2 BY FACTORY DEFAULT.

 MARCH 21, 2001

APPEND THE FOLLOWING DOCUMENTS WHEN CHANGING THIS DOCUMENT:

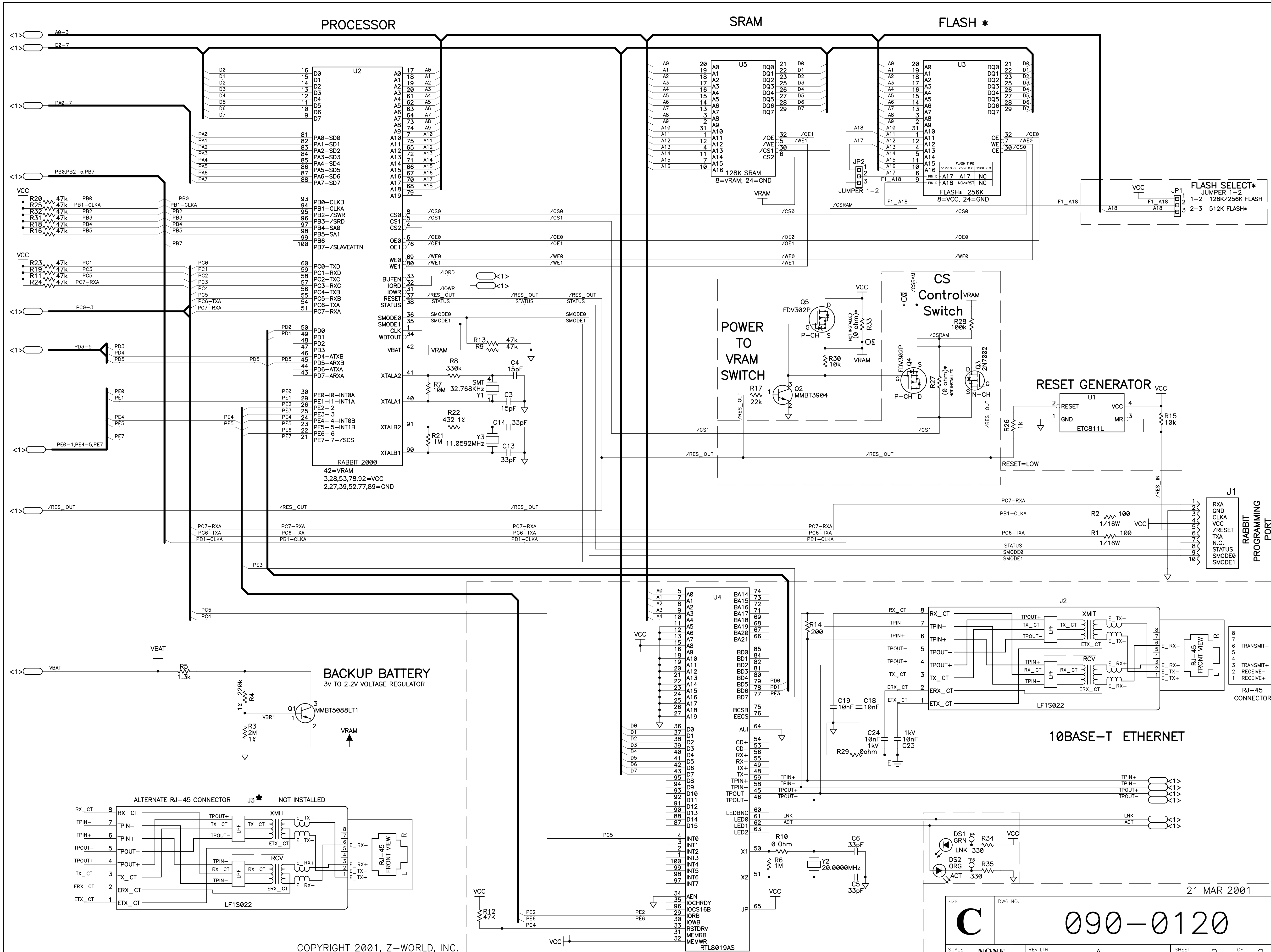
DRAWING CONTENT:	
DRAWN BY: (INITIAL RELEASE) RJH	3/21/01
REVISED BY: RJH	21MAR01
APPROVALS: INITIAL RELEASE	
PROJECT ENGINEER:	
ENGINEERING MANAGER:	
SIGNATURES	DATE

TITLE: SCHEMATIC DIAGRAM MICRO ETHERNET CORE MODULE RCM2200

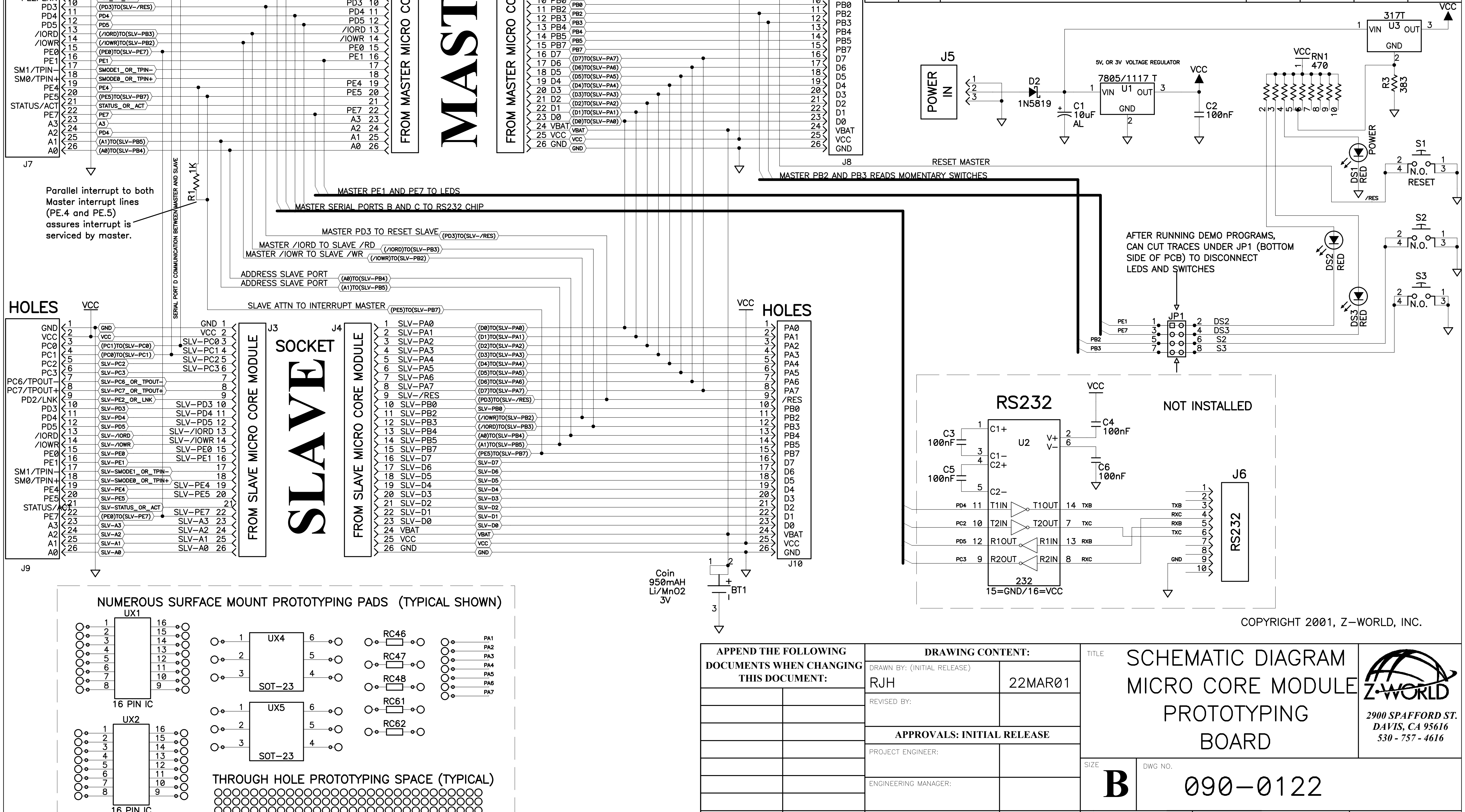
Z-WORLD
2900 SPAFFORD ST.
DAVIS, CA 95616
530 - 757-4616

SIZE: C DWG NO. 090–0120

SCALE: NONE RELEASE DATE: SHEET 1 OF 2


PROCESSOR
SRAM
FLASH *
U2
RABBIT 2000
42=VRAM
3,28,53,78,92=VCC
2,27,39,52,77,89=GND
U5
128K SRAM
8=VRAM; 24=GND
U3
FLASH* 256K
8=VCC, 24=GND
JP2
JUMPER 1-2
FLASH SELECT*
JUMPER 1-2
1-2 128K/256K FLASH
2-3 512K FLASH*
JP1
POWER TO VRAM SWITCH
CS Control Switch
Q5 FDV302P
Q4 FDV302P
Q3 2N7002
Q2 MMBT3904
RESET GENERATOR
U1 ETC811L
RESET=LOW
J1
RABBIT PROGRAMMING PORT
RXA
GND
CLKA
VCC
/RESET
TXA
N.C.
STATUS
SMODE0
SMODE1
BACKUP BATTERY
3V TO 2.2V VOLTAGE REGULATOR
Q1 MMBT5088LT1
ALTERNATE RJ-45 CONNECTOR
J3*
NOT INSTALLED
LF1S022
U4
RTL8019AS
6,17,47,57,70,89=VCC; 14,28,44,52,83,86=GND
J2
XMIT
RCV
RJ-45 CONNECTOR
10BASE-T ETHERNET
Y1 32.768KHz
Y2 20.0000MHz
Y3 11.0592MHz
COPYRIGHT 2001, Z-WORLD, INC.
21 MAR 2001
SIZE C
DWG NO. 090-0120
SCALE NONE
REV LTR A
SHEET 2 OF 2

REVISION HISTORY			REVISION APPROVAL			
REV	ECO	DESCRIPTION	PROJECT ENGINEER	APPROVAL DATE	DOCUMENT CONTROL	APPROVAL DATE
A	E11401	INITIAL RELEASE A/W @ REV-A				



APPEND THE FOLLOWING DOCUMENTS WHEN CHANGING THIS DOCUMENT:	DRAWING CONTENT:	
	DRAWN BY: (INITIAL RELEASE) RJH	22MAR01
	REVISED BY:	
	APPROVALS: INITIAL RELEASE	
	PROJECT ENGINEER:	
	ENGINEERING MANAGER:	
	SIGNATURES	DATE

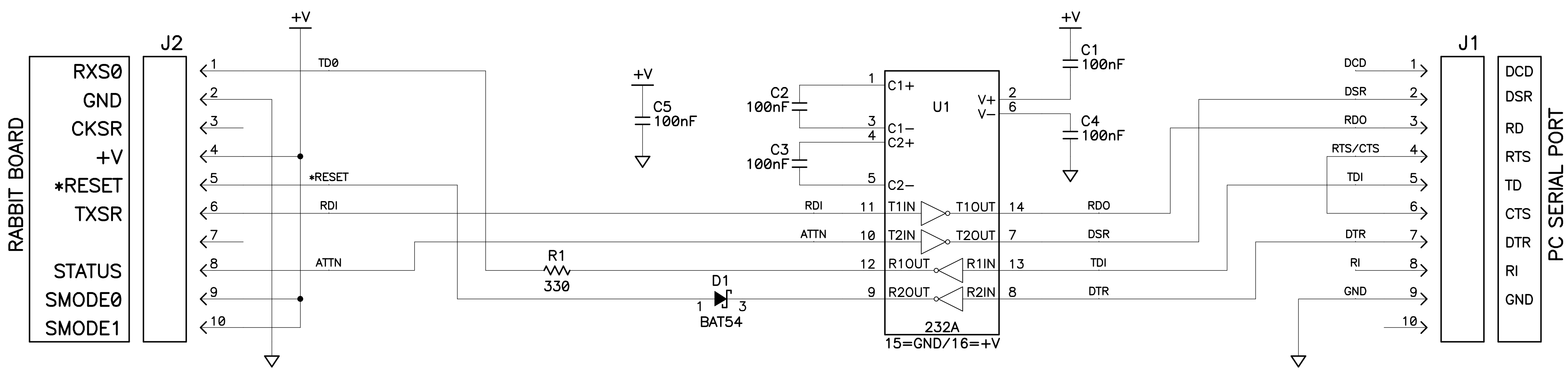
TITLE: SCHEMATIC DIAGRAM MICRO CORE MODULE PROTOTYPING BOARD

Z-WORLD
2900 SPAFFORD ST.
DAVIS, CA 95616
530 - 757 - 4616

SIZE	DWG NO.
B	090-0122

SCALE: NONE | RELEASE DATE: | SHEET 1 OF 1

REVISION HISTORY			REVISION APPROVAL			
REV	ECO	DESCRIPTION	PROJECT ENGINEER	APPROVAL DATE	DOCUMENT CONTROL	APPROVAL DATE
X1	----	Engineering Prototype Release A/W Rev-A	RH	----	----	----
A	E10680	INITIAL RELEASE OF SCHEMATIC, PCB A/W @ REV-B				



NOTES: UNLESS OTHERWISE SPECIFIED;
1. ALL RESISTOR VALUES ARE IN OHMS, 1/10W, 5%
2. ALL CAPACITORS ARE 50VDC OR HIGHER.
3. THE ORIGINATION SOURCE OF A VOLTAGE IS REPRESENTED BY (VCC ▲), AND ALL REFERENCES TO THAT VOLTAGE ARE REPRESENTED BY (VCC ⊤).

APPEND THE FOLLOWING DOCUMENTS WHEN CHANGING THIS DOCUMENT:	

DRAWING CONTENT:	
DRAWN BY: (INITIAL RELEASE) KAH	15MAR99
REVISED BY: KAH	13AUG99
APPROVALS: INITIAL RELEASE	
PROJECT ENGINEER:	
ENGINEERING MANAGER:	
SIGNATURES	DATE

TITLE
SCHEMATIC DIAGRAM
RABBIT SIB

Z-WORLD
2900 SPAFFORD ST.
DAVIS, CA 95616
530 - 757 - 4616

SIZE	DWG NO.
B	090-0085

SCALE	NONE	RELEASE DATE		SHEET 1 OF 1