Digi - 101-0673 Datasheet

Welcome to E-XFL.COM

Understanding Embedded - Microcontroller, Microprocessor, FPGA Modules

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

Applications of Embedded - Microcontroller,

Details	
Product Status	Obsolete
Module/Board Type	MPU Core
Core Processor	Rabbit 3000
Co-Processor	-
Speed	22.1MHz
Flash Size	256KB
RAM Size	128KB
Connector Type	IDC Header 2x20
Size / Dimension	1.23" x 2.11" (31mm x 54mm)
Operating Temperature	-40°C ~ 85°C
Purchase URL	https://www.e-xfl.com/product-detail/digi-international/101-0673

1.2 Advantages of the RCM3600

- 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
- Rabbit Field Utility to download compiled Dynamic C .bin files, 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.

2.5 Where Do I Go From Here?

If the sample program ran fine, you are now ready to go on to other sample programs and to develop your own applications. The source code for the sample programs is provided to allow you to modify them for your own use. The ***RCM3600 User's Manual*** also provides complete hardware reference information and describes the software function calls for the RCM3600, the Prototyping Board, and the optional LCD/keypad module.

For advanced development topics, refer to the ***Dynamic C User's Manual***, which is available in the online documentation set.

2.5.1 Technical Support

> **NOTE:** If you purchased your RCM3600 through a distributor or through a Rabbit Semiconductor partner, contact the distributor or partner first for technical support.

If there are any problems at this point:

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

- **AD_SAMPLE.C**—Demonstrates how to use a low-level driver on single-ended inputs. The program will continuously display the voltage (average of 10 samples) that is present on the A/D channels.

  Before running this program, make sure that pins 3–5 are connected on headers JP5, JP6, and JP7 on the Prototyping Board. No pins are connected on header JP8.

- **ANAINCONFIG.C**—Demonstrates how to use the Register Mode method to read single-ended analog input values for display as voltages. The sample program uses the function call **anaInConfig()** and the ADS7870 CONVERT line to accomplish this task.

  Before running this program, make sure that pins 3–5 are connected on headers JP5, JP6, and JP7 on the Prototyping Board. No pins are connected on header JP8. Also connect PE4 on header J3 on the Prototyping Board to the CNVRT terminal on header J8; if you are using this sample program as a template for your own program, be aware that PE4 is also used as the IrDA FIR_SEL pin.

- **THERMISTOR.C**—Demonstrates how to use analog input THERM_IN7 to calculate temperature for display to the **STDIO** window. This sample program assumes that the thermistor is the one included in the Development Kit whose values for beta, series resistance, and resistance at standard temperature are given in the part specification.

  Before running this program, install the thermistor into the AIN7 and AGND holes at location J7 on the Prototyping Board.

Before running the next two sample programs, **DNLOADCALIB.C** or **UPLOADCALIB.C**, connect your PC serial COM port to header J2 on the Prototyping Board as follows.

- Tx to RxE
- Rx to TxE
- GND to GND

Then connect pins 1–3 and 2–4 on header JP2 on the Prototyping Board.

Now start Tera Term on your PC. Once Tera Term is running, configure the serial parameters as follows:

- Baud rate 19200, 8 bits, no parity, and 1 stop bit.
- Enable the "Local Echo" option.
- Set the line feed options to Receive = CR and Transmit = CR + LF.

Now press **F9** to compile and run this program. Verify that the message "Waiting, Please Send Data file" is being display in Tera Term display window before proceeding. From within Tera Term, select **File > Send File > Path and filename**, then select the OPEN option within the dialog box. Once the data file has been downloaded, it will indicate whether the calibration data were written successfully.

- **DNLOADCALIB.C**—Demonstrates how to retrieve analog calibration data to rewrite it back to simulated EEPROM in flash with using a serial utility such as Tera Term.

4. HARDWARE REFERENCE

Chapter 4 describes the hardware components and principal hardware subsystems of the RCM3600. Appendix A, “RCM3600 Specifications,” provides complete physical and electrical specifications.

Figure 4 shows the Rabbit-based subsystems designed into the RCM3600.

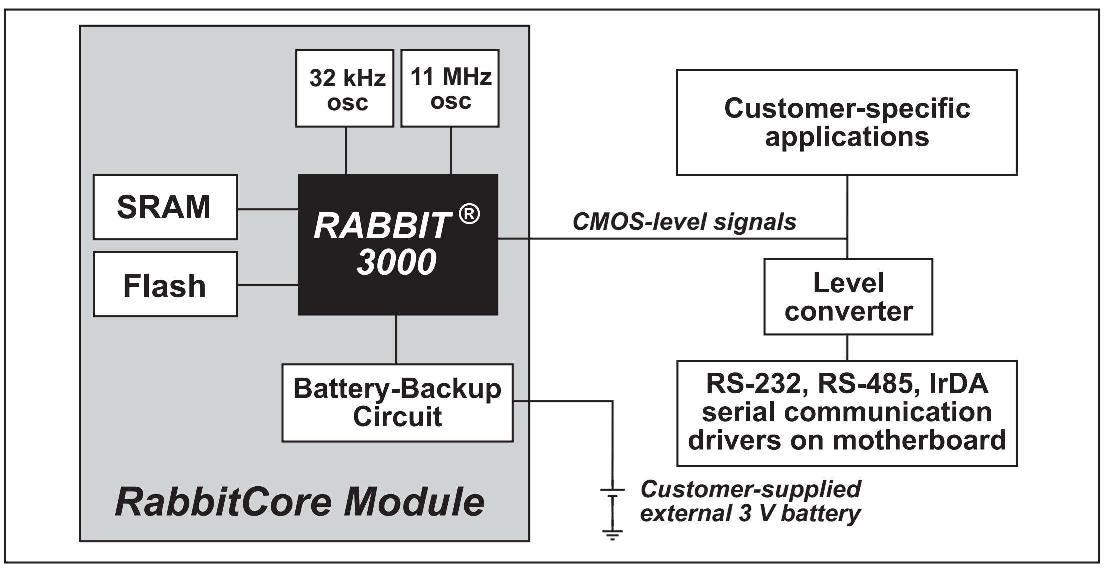

Figure 4. RCM3600 Subsystems

5. Software Reference

Dynamic C is an integrated development system for writing embedded software. It runs on an IBM-compatible PC and is designed for use with Rabbit Semiconductor single-board computers and other single-board computers based on the Rabbit microprocessor. Chapter 5 describes the libraries and function calls related to the RCM3600.

5.1 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. A complete reference guide to Dynamic C is contained in the ***Dynamic C User's Manual*** and in the ***Dynamic C Function Reference Manual***.

You have a choice of doing your software development in the flash memory or in the SRAM included on the RCM3600. The flash memory and SRAM options are selected with the **Options > Compiler** menu.

The advantage of working in RAM is to save wear on the flash memory, which is limited to about 100,000 write cycles. The disadvantage is that the code and data might not both fit in RAM.

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

> **NOTE:** Do not depend on the flash memory sector size or type in your program logic. The RCM3600 and Dynamic C were designed to accommodate flash devices with various sector sizes in response to the volatility of the flash-memory market.

Developing software with Dynamic C is simple. Users can write, compile, and test C and assembly code without leaving the Dynamic C development environment. Debugging occurs while the application runs on the target. Alternatively, users can compile a program to an image file for later loading. Dynamic C runs on PCs under Windows 95 and later. Programs can be downloaded at baud rates of up to 460,800 bps after the program compiles.

RETURN VALUE

A value corresponding to the voltage on the analog input channel:

0–2047 for 11-bit A/D conversions (signed 12th bit)
`ADOVERFLOW` (defined macro = -4096) if overflow or out of range
-4095 if conversion is incomplete or busy-bit timeout

SEE ALSO

`anaIn, anaInConfig, anaInDriver`

`value1` is the first A/D converter channel value (0–2047)

`volts1` is the voltage or current corresponding to the first A/D converter channel value (0 to +20 V or 4 to 20 mA)

`value2` is the second A/D converter channel value (0–2047)

`volts2` is the voltage or current corresponding to the first A/D converter channel value (0 to +20 V or 4 to 20 mA)

RETURN VALUE

0 if successful.

-1 if not able to make calibration constants.

SEE ALSO

`anaIn, anaInVolts, anaInmAmps, anaInDiff, anaInCalib, brdInit`

`float anaInVolts(unsigned int channel, unsigned int gaincode);`

Reads the state of a single-ended analog input channel and uses the calibration constants previously set using **`anaInCalib`** to convert it to volts.

PARAMETERS

`channel` is the channel number (0–7)

Channel Code	Single-Ended Input Lines*	Voltage Range† (V)
0	+AIN0	0–20
1	+AIN1	0–20
2	+AIN2	0–20
3	+AIN3	0–20
4	+AIN4	0–20
5	+AIN5	0–20
6	+AIN6	0–20
7	+AIN7	0–2‡

* Negative input is ground.
† Applies to RCM3600 Prototyping Board.
‡ Used for thermistor in sample program.

`gaincode` is the gain code of 0 to 7

Gain Code	Multiplier	Voltage Range* (V)
0	×1	0–20
1	×2	0–10
2	×4	0–5
3	×5	0–4
4	×8	0–2.5
5	×10	0–2
6	×16	0–1.25
7	×20	0–1

* Applies to RCM3600 Prototyping Board.

RETURN VALUE

A voltage value corresponding to the voltage on the analog input channel.

`ADOVERFLOW` (defined macro = -4096) if overflow or out of range.

SEE ALSO

`anaInCalib, anaIn, anaInmAmps, brdInit`

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***.

A.1 Electrical and Mechanical Characteristics

Figure A-1 shows the mechanical dimensions for the RCM3600.

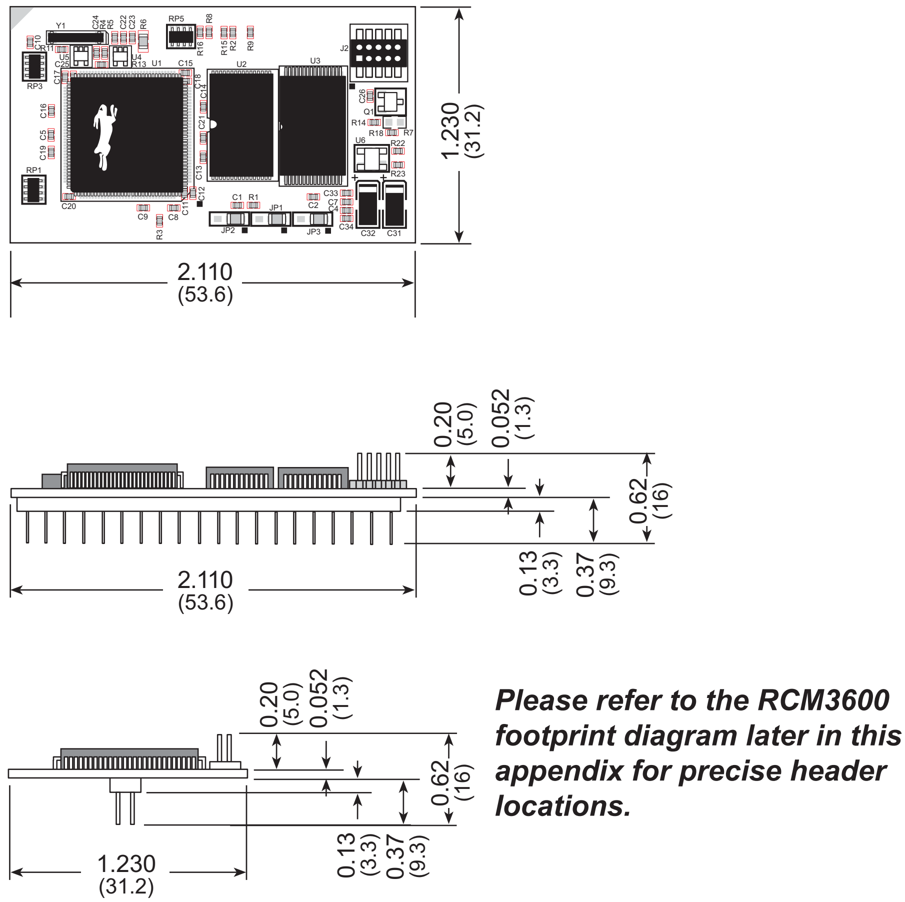

Figure A-1. RCM3600 Dimensions

NOTE: All measurements are in inches followed by millimeters enclosed in parentheses. All dimensions have a manufacturing tolerance of ±0.01" (0.25 mm).

It is recommended that you allow for an “exclusion zone” of 0.04" (1 mm) around the RCM3600 in all directions when the RCM3600 is incorporated into an assembly that includes other printed circuit boards. This “exclusion zone” that you keep free of other components and boards will allow for sufficient air flow, and will help to minimize any electrical or electromagnetic interference between adjacent boards. An “exclusion zone” of 0.08" (2 mm) is recommended below the RCM3600 when the RCM3600 is plugged into another assembly using the shortest connectors for header J1. Figure A-2 shows this “exclusion zone.”

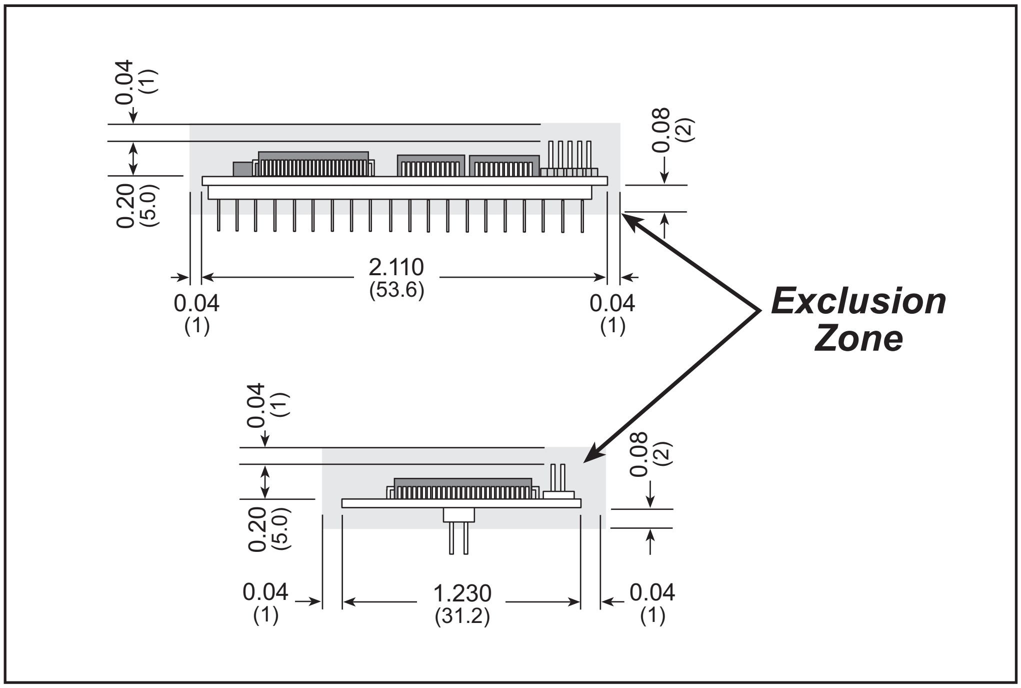

Figure A-2. RCM3600 “Exclusion Zone”

APPENDIX B. PROTOTYPING BOARD

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

B.4 Using the Prototyping Board

The Prototyping Board is actually both a demonstration board and a prototyping board. As a demonstration board, it can be used to demonstrate the functionality of the RCM3600 right out of the box without any modifications.

The Prototyping Board pinouts are shown in Figure B-4.

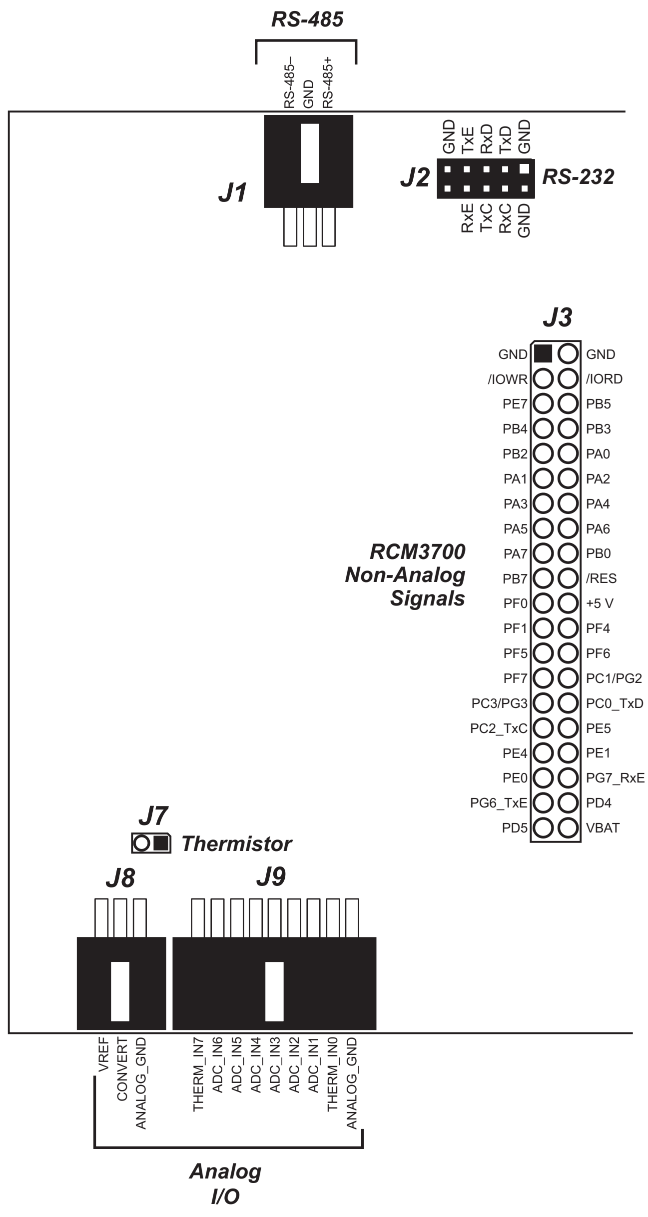

Figure B-4. Prototyping Board Pinout

Table B-5 lists the configuration options using jumpers.

Table B-5. RCM3600 Jumper Configurations

Header	Description	Pins Connected		Factory Default
JP1	RS-485 Bias and Termination Resistors	1–2 5–6	Bias and termination resistors connected	×
		1–3 4–6	Bias and termination resistors ***not*** connected (parking position for jumpers)	
JP2	RS-232/RS-485 on Serial Port E	1–3 2–4	RS-232	
		3–5 4–6	RS-485	×
JP4	A/D Converter Outputs	1	PIO_0	n.c.
		2	PIO_1	n.c.
		3	PIO_2	n.c.
		4	PIO_3	n.c.
JP5	ADC_IN4–ADC_IN5	1–2	Tied to VREF	
		2–3	Tied to analog ground	×
JP6	ADC_IN2–ADC_IN3	1–2	Tied to VREF	
		2–3	Tied to analog ground	×
JP7	ADC_IN0–ADC_IN1	1–2	Tied to VREF	
		2–3	Tied to analog ground	×
JP8	Analog Voltage/4–20 mA Options	1–2	Connect for 4–20 mA option on ADC_IN3	n.c.
		3–4	Connect for 4–20 mA option on ADC_IN4	n.c.
		5–6	Connect for 4–20 mA option on ADC_IN5	n.c.
		7–8	Connect for 4–20 mA option on ADC_IN6	n.c.

C.8 Sample Programs

Sample programs illustrating the use of the LCD/keypad module with the Prototyping Board are provided in the **SAMPLES\RCM3600\LCD_KEYPAD** folder.

These sample programs use the auxiliary I/O bus on the Rabbit 3000 chip, and so the **#define PORTA_AUX_IO** line is already included in the sample programs.

Each sample program has comments that describe the purpose and function of the program. Follow the instructions at the beginning of the sample program. To run a sample program, open it with the **File** menu (if it is not still open), then compile and run it by pressing **F9**. The RCM3600 must be connected to a PC using the programming cable as described in Chapter 2, "Getting Started."

Complete information on Dynamic C is provided in the ***Dynamic C User's Manual***.

- **KEYPADTOLED.C**—This program demonstrates the use of the external I/O bus. The program will light up an LED on the LCD/keypad module and will display a message on the LCD when a key press is detected. The DS1 and DS2 LEDs on the Prototyping Board will also light up.
- **LCDKEYFUN.C**—This program demonstrates how to draw primitive features from the graphic library (lines, circles, polygons), and also demonstrates the keypad with the key release option.
- **SWITCHTOLED.C**—This program demonstrates the use of the external I/O bus. The program will light up an LED on the LCD/keypad module and will display a message on the LCD when a switch press is detected. The DS1 and DS2 LEDs on the Prototyping Board will also light up.

Additional sample programs are available in the **122x32_1x7** folder in **SAMPLES\LCD_KEYPAD**.

void glBlock(int left, int top, int width, int height);

Draws a rectangular block in the page buffer and on the LCD if the buffer is unlocked. Any portion of the block that is outside the LCD display area will be clipped.

PARAMETERS

`left` is the *x* coordinate of the top left corner of the block.

`top` is the *y* coordinate of the top left corner of the block.

`width` is the width of the block.

`height` is the height of the block.

RETURN VALUE

None.

SEE ALSO

`glFillScreen, glBlankScreen, glPlotPolygon, glPlotCircle`

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`

A lithium battery with a nominal voltage of 3 V and a minimum capacity of 165 mA·h is recommended. A lithium battery is strongly recommended because of its nearly constant nominal voltage over most of its life.

The drain on the battery by the RCM3600 is typically 6 µA when no other power is supplied. If a 235 mA·h battery is used, the battery can last about 4.5 years:

$$\frac{235\ \text{mA}\cdot\text{h}}{6\ \mu\text{A}} = 4.5\ \text{years.}$$

The actual life in your application will depend on the current drawn by components not on the RCM3600 and the storage capacity of the battery. The RCM3600 does not drain the battery while it is powered up normally.

Cycle the main power off/on on the RCM3600 after you install a backup battery for the first time, and whenever you replace the battery. This step will minimize the current drawn by the real-time clock oscillator circuit from the backup battery should the RCM3600 experience a loss of main power.

> **NOTE:** Remember to cycle the main power off/on any time the RCM3600 is removed from the Protoyping Board or motherboard since that is where the backup battery would be located.

D.1.2 Reset Generator

The RCM3600 uses a reset generator to reset the Rabbit 3000 microprocessor when the voltage drops below the voltage necessary for reliable operation. The reset occurs between 2.55 V and 2.70 V, typically 2.63 V.

The RCM3600 has a reset pin, pin 36 on header J1. This pin provides access to the reset output from the reset generator, and is also connected to the reset input of the Rabbit 3000 to allow you to reset the microprocessor externally. A resistor divider consisting of R21 and R22 attenuates the signal associated with an externally applied reset to prevent it from affecting the reset generator.