This is the Revision A version of the Threshold4 RoboBrick. The status of this project is that it has been replaced by the Light4 RoboBrick.

Threshold4 Robobrick (Revision A)

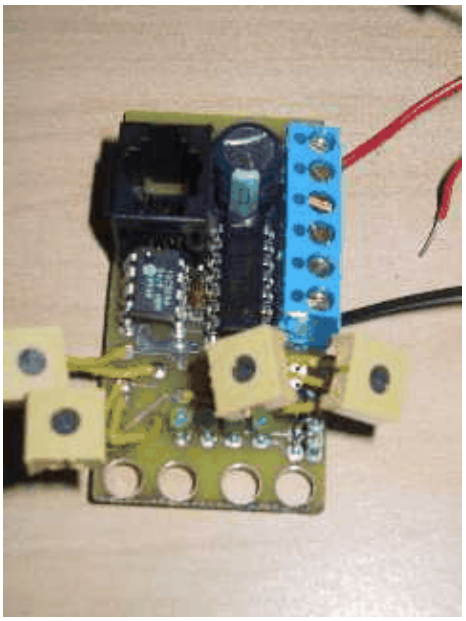
Table of Contents

This document is also available as a PDF document.

1. Introduction

The Threshold4 RoboBrick provides four voltage comparators to convert 4 analog input voltages into four bits of 1 or 0. There are four trim potentiometers that are used to set a threshold voltage between 0 and 5 volts for each of the four voltage comparators. To aid in setting the trim potentiometers, the outputs of the voltage comparators are sent to four LED's. The resulting 4 binary bits of data are available for querying.

The picture below shows the Threshold4A RoboBrick:



2. Programming

The basic Threshold4 RoboBrick operation is to send a query to the module to read the 4 bits of data. The programmer can download a complement mask to cause any of the bits to be complemented prior to reading.

The Threshold4 RoboBrick supports RoboBrick Interrupt Protocol. The interrupt pending bit is set whenever the the formula:

L&(~I) | H&I | R&(~P)&I | F&P&(~I)]

is non−zero, where:

- I is the current input bits XOR'ed with the complement mask (C)
- P is the previous value of I
- L is the low mask
- H is the high mask
- R is the raising mask
- F is the falling mask

and

- ~ is bit−wise complement
- | is bit−wise OR
- & is bit−wise AND

Once the interrupt pending bit is set, it must be explicitly cleared by the user.

The Threshold4 commands are summarized in the table below:

Command	Send/ Receive	Byte Value								Discussion
		7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
Read Inputs	Send	0	0	0	0	0	0	0	0	Return input values *abcd* (after XOR'ing with complement mask)
	Receive	0	0	0	0	*a*	*b*	*c*	*d*	
Read Complement Mask	Send	0	0	0	0	0	0	0	1	Return complement mask *cccc*
	Receive	0	0	0	0	*c*	*c*	*c*	*c*	
Read High Mask	Send	0	0	0	0	0	0	1	0	Return high mask *hhhh*
	Receive	0	0	0	0	*h*	*h*	*h*	*h*	
Read Low Mask	Send	0	0	0	0	0	0	1	1	Return low mask *llll*
	Receive	0	0	0	0	*l*	*l*	*l*	*l*	
Read Raising Mask	Send	0	0	0	0	0	1	0	0	Return raising mask *rrrr*
	Receive	0	0	0	0	*r*	*r*	*r*	*r*	
Read Falling Mask	Send	0	0	0	0	0	1	0	1	Return falling mask *ffff*
	Receive	0	0	0	0	*f*	*f*	*f*	*f*	
Read Raw	Send	0	0	0	0	0	1	1	0	Return raw data *abcd* (without XOR'ing with complement mask)
	Receive	0	0	0	0	*a*	*b*	*c*	*d*	
Set Complement Mask	Send	0	0	0	1	*c*	*c*	*c*	*c*	Set compliment mask to *cccc*
Set High Mask	Send	0	0	1	0	*h*	*h*	*h*	*h*	Set high mask to *hhhh*
Set Low Mask	Send	0	0	1	1	*l*	*l*	*l*	*l*	Set low mask to *llll*
Set Raising Mask	Send	0	1	0	0	*r*	*r*	*r*	*r*	Set raising mask to *rrrr*
Set Falling Mask	Send	0	1	0	1	*f*	*f*	*f*	*f*	Set falling mask to *ffff*

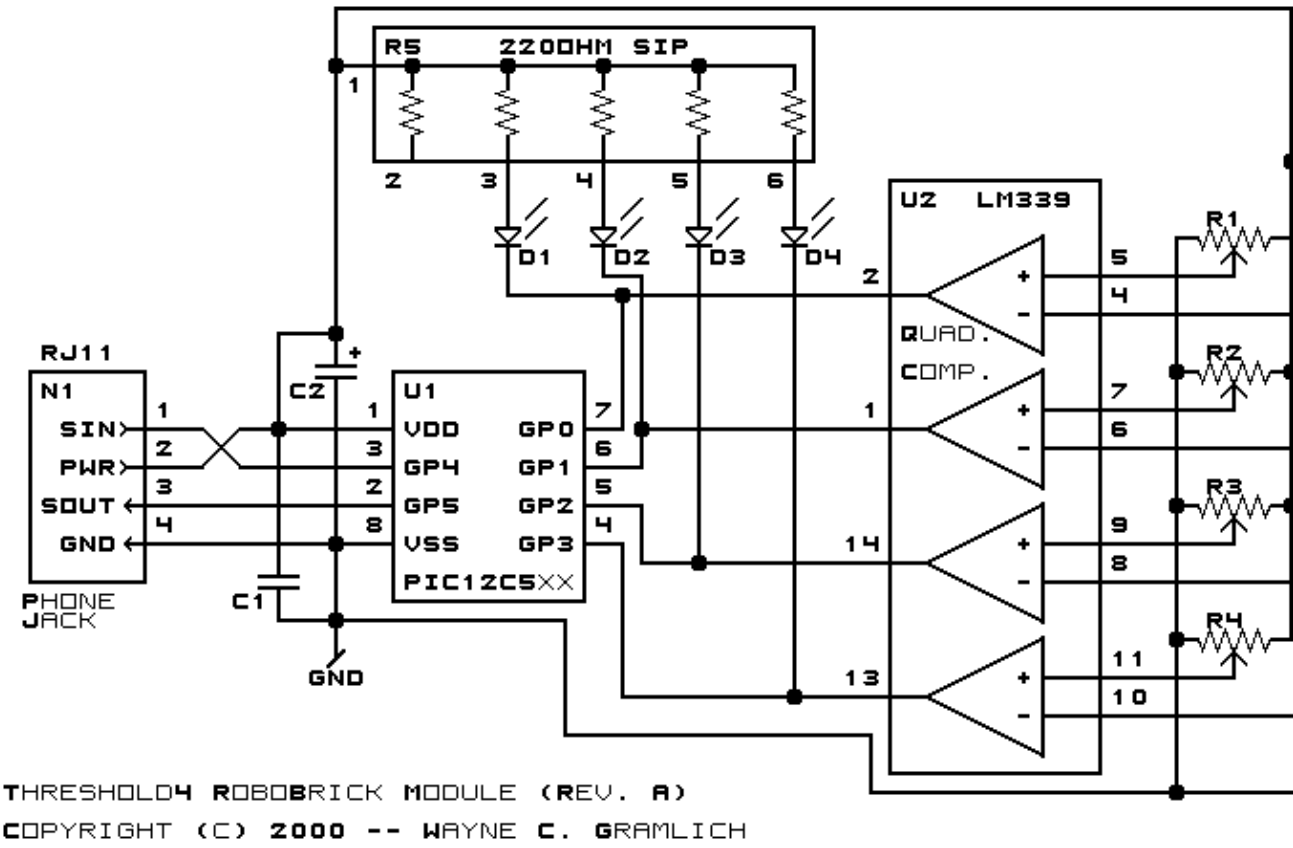
Read Interrupt Bits	Send	1	1	1	0	1	1	1	1	Return the interrupt pending bit *p* and the interrupt enable bit *e*.
	Receive	0	0	0	0	0	0	*e*	*p*	
Set Interrupt Commands	Send	1	1	1	1	0	*c*	*c*	*c*	Execute shared set interrupt command *ccc*.
Shared Commands	Send	1	1	1	1	1	*c*	*c*	*c*	Execute shared command *ccc*.

3. Hardware

The hardware consists of a circuit schematic and a printed circuit board.

3.1 Circuit Schematic

The schematic for the Threshold4 RoboBrick is shown below:



The parts list kept in a separate file −− threshold4.ptl.

3.2 Printed Circuit Board

The printed circuit board files are listed below:

threshold4_back.png
: The solder side.

threshold4_front.png
: The component side.

threshold4_artwork.png
: The artwork.

threshold4.gbl
: The RS−274X "Gerber" back (solder side) layer.

threshold4.gtl
: The RS−274X "Gerber" top (component side) layer.

threshold4.gal
: The RS−274X "Gerber" artwork layer.

threshold4.drl
: The "Excellon" NC drill file.

threshold4.tol
: The "Excellon" NC drill rack file.

4. Software

Each Threshold4 RoboBrick has essentially the same program in it as the In4 Robobrick. The *only* difference is that the Robobrick Query command gives back a different number.

The Threshold4 software is available as one of:

threshold4.ucl
: The µCL source file.

threshold4.asm
: The resulting human readable PIC assembly file.

threshold4.lst
: The resulting human readable PIC listing file.

threshold4.hex
: The resulting Intel® Hex file that can be fed into a PIC12C5xx programmer.

In addition to the Threshold4 RoboBrick software, there is some testing software too:

threshold4_test.ucl
: The µCL source file.

threshold4_test.asm
: The resulting human readable PIC assembly file.

threshold4_test.lst
: The resulting human readable PIC listing file.

threshold4_test.hex
: The resulting Intel® Hex file.

The Threshold4 Testing software is loaded into a Harness RoboBrick. Upon power up, a prompt of the form:

```
Threshold4A?
```

is displayed. After plugging a Threshold4A into the other end the Harness, type any character.

A bunch of information from the identification string is printed. Something like:

```
123 123 123 123 123 123 123 123
123 123 123 123 123 123 123 123
```

```
Threshold4A
Gramlich
```

The first 16 numbers is the 128−bit random number burned into the RoboBrick. The next two lines are RoboBrick name and vendor strings.

Next the program will prompt for an input pattern with something like:

```
0*0*?
```

Make the LED's on the Threshold4 look like the pattern where `0' means the LED is off and `*' means the LED is on. Type any character to continue. Keep doing this until the tests end with the following message.

```
Done
Threshold4A?
```

Any errors that occur will look like:

name `Fail` *octal*

where

name
: is the test name, and

octal
: is the test number, in octal, that failed.

5. Issues

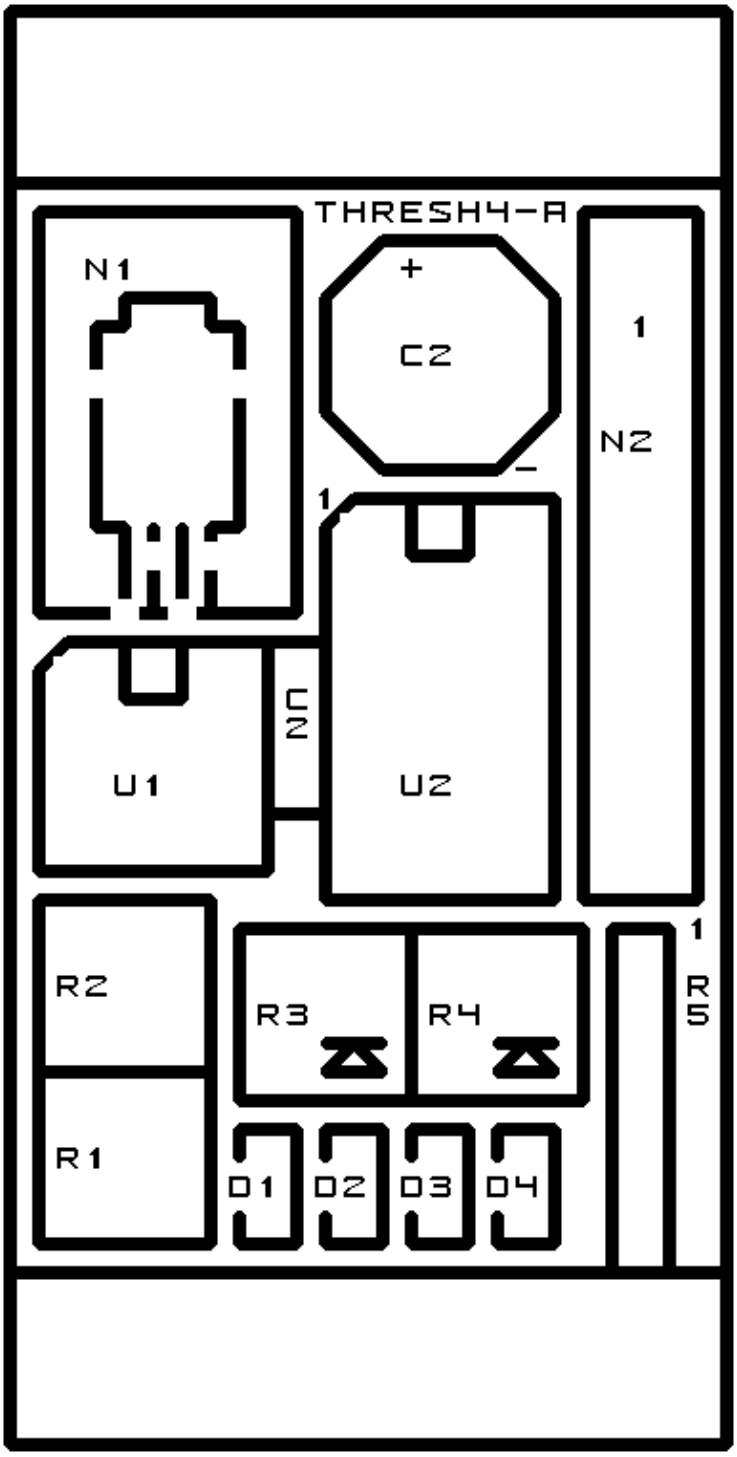
The following issues have come up:

- The terminal strip holes are too small.
- The potentiometer wires did not line up with holes.
- Think about adding pull−up resistors to the inputs.
- Think about providing a current limiting resistor for the power.
- The LED's should be swapped so that the LSB LED is to the right.
- Remove the 2200µF capacitor.
- Switch over to 6−wire connectors.
- The PIC is too close to the RJ11 socket.

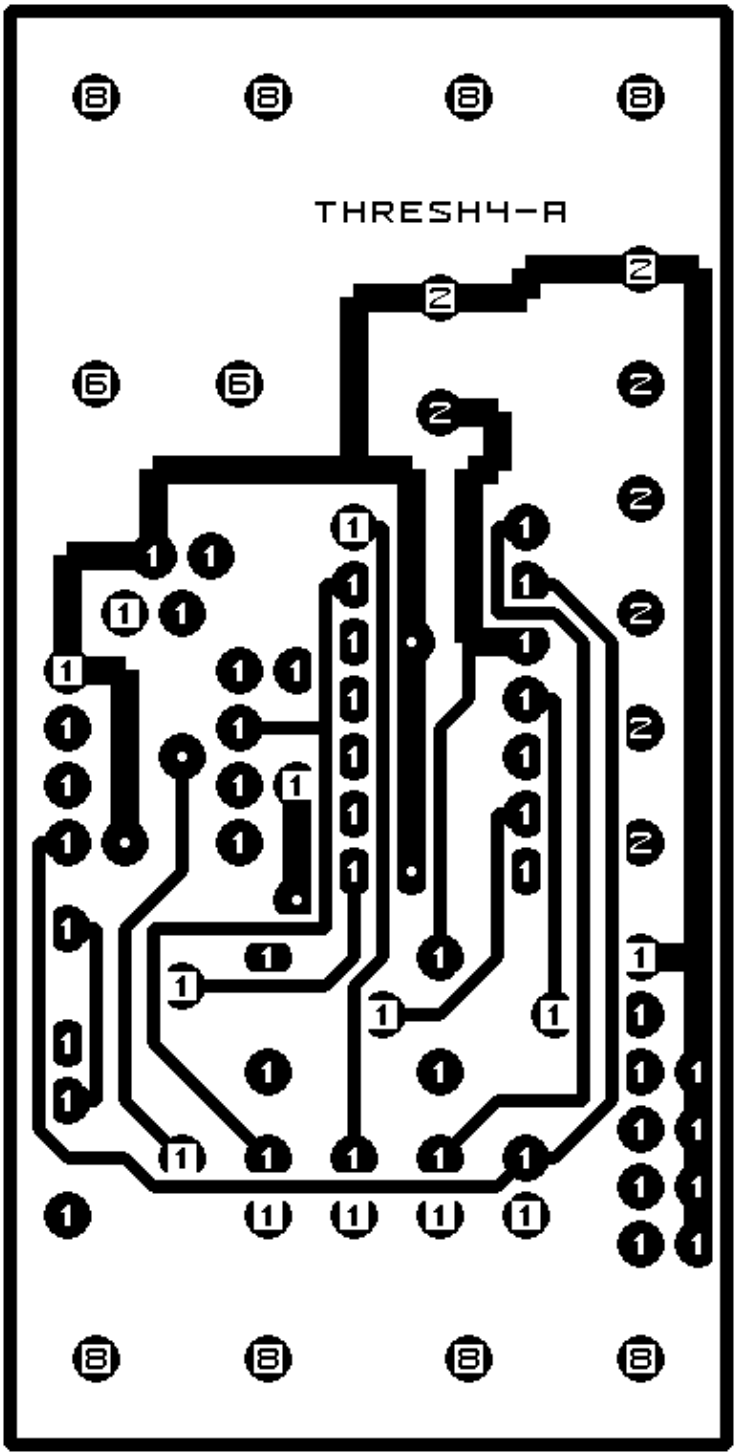
A. Appendix A: Parts List

```
# Parts list for Threshold4 RoboBrick (Rev. A)
#
C1: Capacitor10pF − 10 pF Ceramic Capacitor [Jameco: 15333]
C2: Capacitor2200uF − 2200 uF 6.3V Electrolytic Capacitor [Jameco: 133145]
D1−4: LEDGreen − Small Green LED [Jameco: 34606]
N1: RJ11Female4_4.RBSlave − Female RJ11 (4−4) Phone Jack [Digikey: A9071−ND]
N2: TerminalStrip6.Threshold4 − 6 Junction Terminal Strip [2 Jameco: 189667]
R1−4: ResistorTrimPot100K.Threshold4 − 100K Trim Potentionmeter [Jameco: 95484]
R5: Resistor5SIP220 − 5 220Ohm 1/4 resistors in a SIP package [Digikey: 770−61−R220−ND]
U1: PIC12C509.Threshold4 − Microchip PIC12C509 [Digikey: PIC12C509A−04/P−ND]
U2: LM339 − Quad Comparator [Jameco: 23851]
```

B. Appendix B: Artwork Layer



C. Appendix C: Back (Solder Side) Layer



D. Appendix D: Front (Component Side) Layer