

This is the Revision A version of the [In8 RoboBrick](#). The status of this project is that it has been [replaced](#) by the [InOut10 RoboBrick](#).

In8 Robobrick (Revision A)

Table of Contents

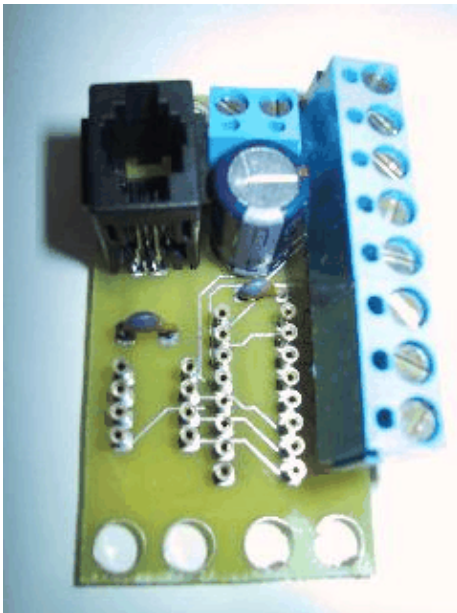
This document is also available as a [PDF](#) document.

- [1. Introduction](#)
- [2. Programming](#)
- [3. Hardware](#)
 - ◆ [3.1 Circuit Schematic](#)
 - ◆ [3.2 Printed Circuit Board](#)
- [4. Software](#)
- [5. Issues](#)

1. Introduction

The In8 RoboBrick allows you to read up to 8 digital inputs. An interrupt can be generated on the states of selected inputs.

A picture of the In8-A RoboBrick is shown below:



2. Programming

The basic operation is to send a query to the In8 RoboBrick 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.

In8 RoboBrick (Revision A)

The In8 RoboBrick supports [RoboBrick Interrupt Protocol](#). The interrupt pending bit is set whenever the the formula:

$$L \& (\sim I) \mid H \& I \mid R \& (\sim P) \& I \mid F \& P \& (\sim I)$$

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 In8 RoboBrick supports both the standard [shared commands](#) and the [shared interrupt commands](#) in addition to the following commands:

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 <i>abcdefgh</i> (after XOR'ing with complement mask)
	Receive	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	
Read Complement Mask	Send	0	0	0	0	0	0	0	1	Return complement mask <i>cccccccc</i>
	Receive	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	
Read Low Mask	Send	0	0	0	0	0	0	1	0	Return low mask <i>llllllll</i>
	Receive	<i>l</i>	<i>l</i>	<i>l</i>	<i>l</i>	<i>l</i>	<i>l</i>	<i>l</i>	<i>l</i>	
Read High Mask	Send	0	0	0	0	0	1	1	1	Return high mask <i>hhhhhhhh</i>
	Receive	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	
Read Raising Mask	Send	0	0	0	0	1	0	0	0	Return raising mask <i>rrrrrrrr</i>
	Receive	<i>r</i>	<i>r</i>	<i>r</i>	<i>r</i>	<i>r</i>	<i>r</i>	<i>r</i>	<i>r</i>	
Read Falling Mask	Send	0	0	0	0	1	0	1	1	Return falling mask <i>ffffff</i>
	Receive	<i>f</i>	<i>f</i>	<i>f</i>	<i>f</i>	<i>f</i>	<i>f</i>	<i>f</i>	<i>f</i>	
Read Raw	Send	0	0	0	0	1	0	0	0	Return raw data <i>abcd</i> (without XOR'ing with complement mask)
	Receive	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	
Set Complement Mask	Send	0	0	0	0	1	0	0	1	Set complement mask to <i>cccccccc</i>
	Send	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	
Set Low Mask	Send	0	0	0	0	1	0	1	0	Set low mask to <i>llllllll</i>
	Send	<i>l</i>	<i>l</i>	<i>l</i>	<i>l</i>	<i>l</i>	<i>l</i>	<i>l</i>	<i>l</i>	
Set High Mask	Send	0	0	0	0	1	0	1	1	Set high mask to <i>hhhhhhhh</i>

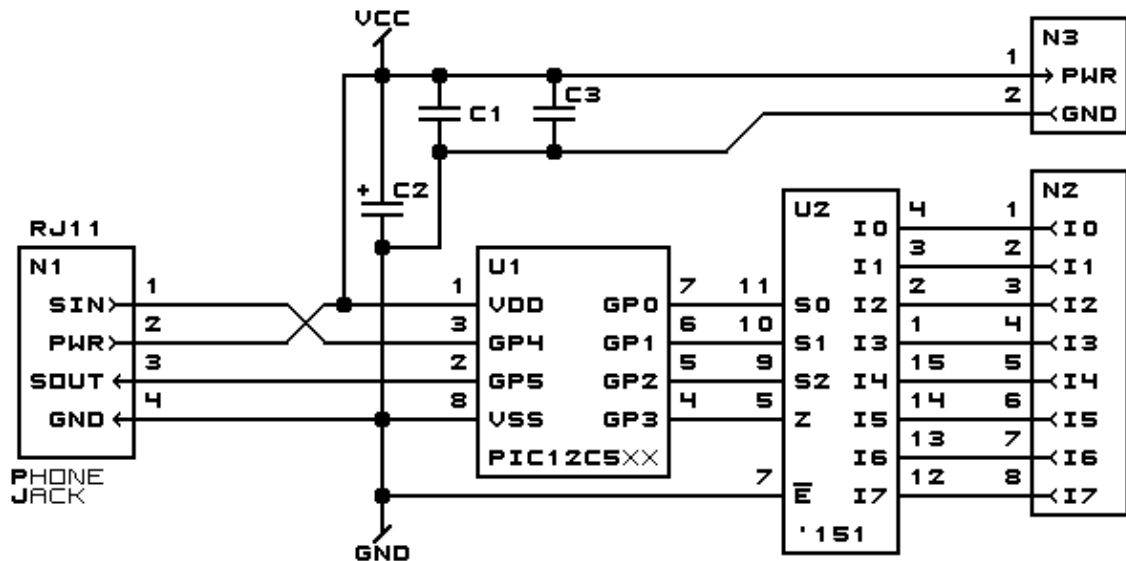
	Send	<i>h h h h h h h h</i>	
Set Raising Mask	Send	0 0 0 0 1 1 0 0	Set raising mask to <i>rrrrrrrr</i>
	Send	<i>r r r r r r r r</i>	
Set Falling Mask	Send	0 0 0 0 1 1 0 1	Set falling mask to <i>ffffff</i>
	Send	<i>f f f f f f f f</i>	
Read Interrupt Bits	Send	1 1 1 0 1 1 1 1	Return the interrupt pending bit <i>p</i> and the interrupt enable bit <i>e</i> .
	Receive	0 0 0 0 0 0 <i>e p</i>	
Set Interrupt Commands	Send	1 1 1 1 0 <i>c c c</i>	Set Interrupt Command <i>ccc</i> .
Shared Commands	Send	1 1 1 1 1 <i>c c c</i>	Execute Shared Command <i>ccc</i> .

3. Hardware

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

3.1 Circuit Schematic

The schematic for the In8 RoboBrick is shown below:



IN8 ROBOBRICK MODULE (REV. A)
 COPYRIGHT (C) 2000 -- WAYNE C. GRAMLICH

The parts list kept in a separate file -- [in8.ptl](#).

3.2 Printed Circuit Board

The printed circuit board files are listed below:

[in8_back.png](#)

The solder side layer.

[in8_front.png](#)

The component side layer.

[in8_artwork.png](#)

x The artwork layer.

[in8.gbl](#)

The RS-274X "Gerber" back (solder side) layer.

[in8.gtl](#)

The RS-274X "Gerber" top (component side) layer.

[in8.gal](#)

The RS-274X "Gerber" artwork layer.

[in8.drl](#)

The "Excellon" NC drill file.

[in8.tol](#)

The "Excellon" tool rack file.

4. Software

The In8 software is available as one of:

[in8.ucl](#)

The μ CL source file.

[in8.asm](#)

The resulting human readable PIC assembly file.

[in8.lst](#)

The resulting human readable PIC listing file.

[in8.hex](#)

The resulting Intel[®] Hex file that can be fed into a PIC12C5xx programmer.

In addition, there is a test suite available as:

[in8_test.ucl](#)

The μ CL source file.

[in8_test.asm](#)

The resulting human readable PIC assembly file.

[in8_test.lst](#)

The resulting human readable PIC listing file.

[in8_test.hex](#)

The resulting Intel[®] Hex file that can be fed into a PIC12C5xx programmer.

5. Issues

The following issues have come up:

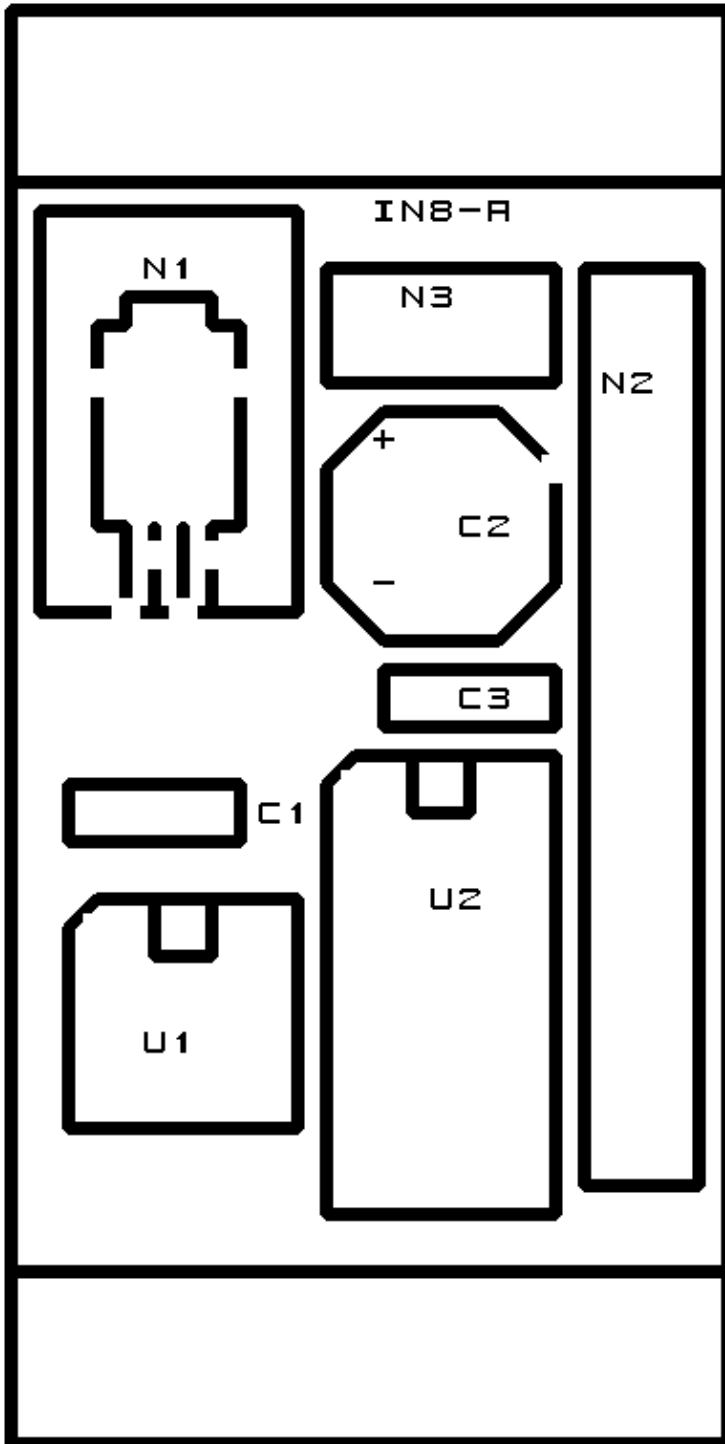
- The 2200 μ F capacitor does not fit between the RJ11 and the terminal strip.
- One of the traces has an unnecessary kink in it.
- The 8-pin terminal strip is too close to the the 74LS151.
- The terminal strip holes are too small.
- The Lego holes are not right.
- The RJ11 holes are not right.
- We need to switch over to a 6-wire RJ11 connector.

Copyright (c) 2000–2002 by Wayne C. Gramlich. All rights reserved.

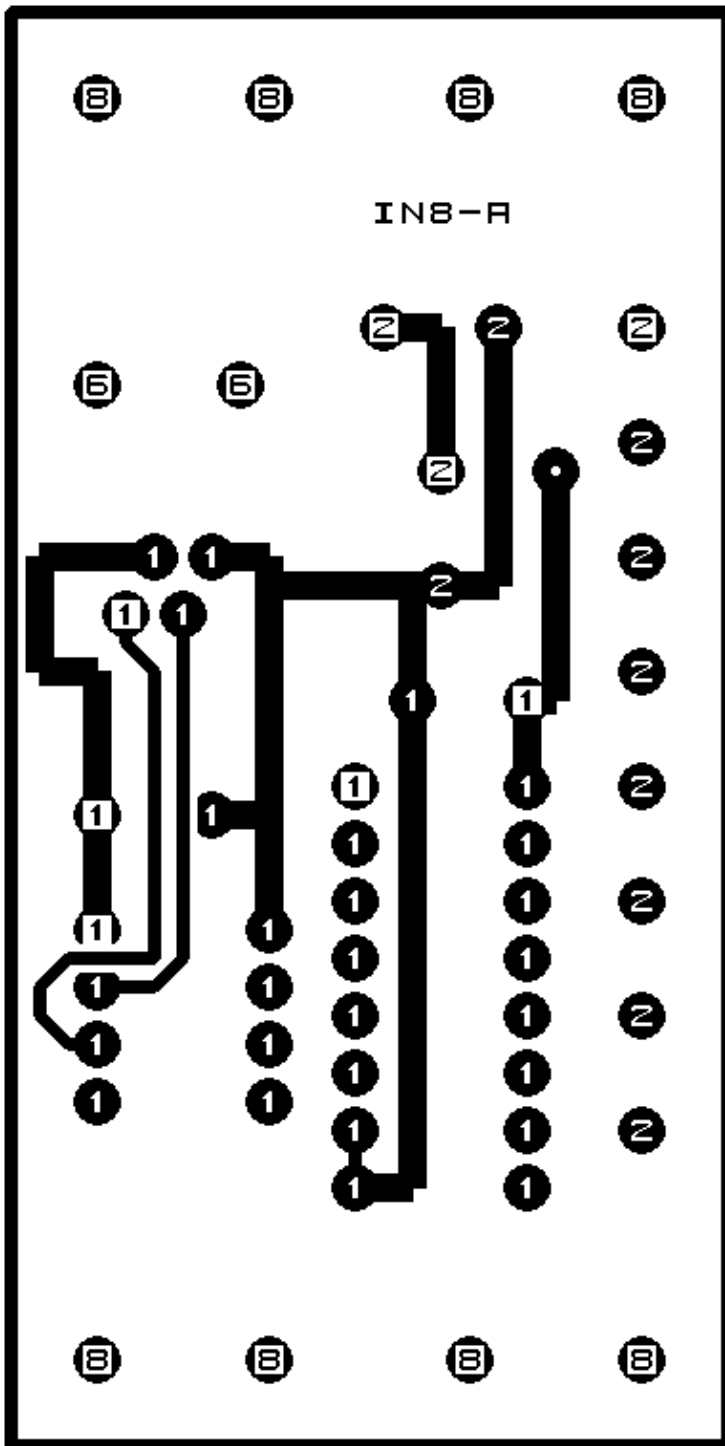
A. Appendix A: Parts List

```
# Parts list for In8 RoboBrick (Rev. A)
#
C1: Capacitor10pF - 10 pF Ceramic Capacitor [Jameco: 15333]
C2: Capacitor2200uF - 2200 uF 6.3V Electrolytic Capacitor [Jameco: 133145]
C3: Capacitor10pF - 10 pF Ceramic Capacitor [Jameco: 15333]
N1: RJ11Female4_4.RBSlave - Female RJ11 (4-4) Phone Jack [Digikey: A9071-ND]
N2: TerminalStrip8_In8 - 8 Junction Terminal Strip [4 Jameco: 189675]
N3: TerminalStrip2.In8 - 2 Junction Terminal Strip [Jameco: 189675]
U1: PIC12C509.In8 - Microchip PIC12C509 [Digikey: PIC12C509A-04/P-ND]
U2: 74HCT151 - 8 to 1 Selector [Digikey: CD74HCT151E]
```

B. Appendix B: Artwork Layer



C. Appendix C: Back (Solder Side) Layer



D. Appendix D: Front (Component Side) Layer

