This is the Revision A version of the Switch8 RoboBrick. The status of this project is that it has been replaced by the Revision B version.

Switch8 Robobrick (Revision A)

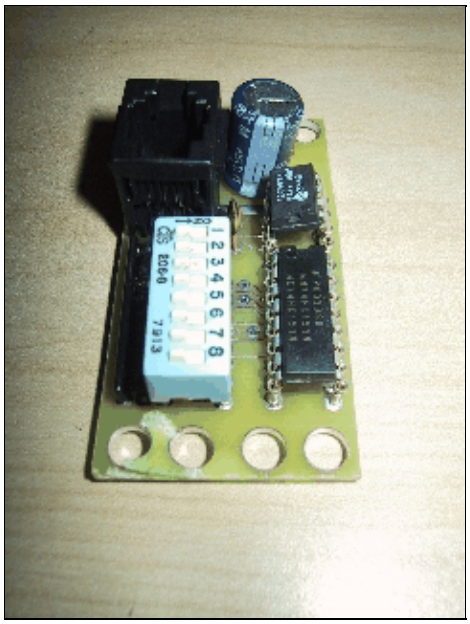
Table of Contents

This document is also available as a PDF document.

1. Introduction

The Switch8 RoboBrick allows you to read up to 8 digital inputs. An interrupt can be generated on the states of selected inptus.

A picture of the Switch8−A RoboBrick is shown below:



2. Programming

The basic operation is to send a query to the Switch8 RoboBrick to read the 8 bits of data. The programmer can download a complement mask to cause any of the bits to be complemented prior to reading.

The Switch8 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 Switch8 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 *abcdefgh* (after XOR'ing with complement mask)
	Receive	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	
Read Complement Mask	Send	0	0	0	0	0	0	0	1	Return complement mask *cccccccc*
	Receive	*c*	*c*	*c*	*c*	*c*	*c*	*c*	*c*	
Read Low Mask	Send	0	0	0	0	0	0	1	0	Return low mask *llllllll*
	Receive	*l*	*l*	*l*	*l*	*l*	*l*	*l*	*l*	
Read High Mask	Send	0	0	0	0	0	0	1	1	Return high mask *hhhhhhhh*
	Receive	*h*	*h*	*h*	*h*	*h*	*h*	*h*	*h*	
Read Raising Mask	Send	0	0	0	0	0	1	0	0	Return raising mask *rrrrrrrr*
	Receive	*r*	*r*	*r*	*r*	*r*	*r*	*r*	*r*	
Read Falling Mask	Send	0	0	0	0	0	1	0	1	Return falling mask *ffffffff*
	Receive	*f*	*f*	*f*	*f*	*f*	*f*	*f*	*f*	
Read Raw	Send	0	0	0	0	1	0	0	0	Return raw data *abcd* (without XOR'ing with complement mask)
	Receive	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	
Set Complement Mask	Send	0	0	0	0	1	0	0	1	Set complement mask to *cccccccc*
	Send	*c*	*c*	*c*	*c*	*c*	*c*	*c*	*c*	
Set Low Mask	Send	0	0	0	0	1	0	1	0	Set low mask to *llllllll*
	Send	*l*	*l*	*l*	*l*	*l*	*l*	*l*	*l*	
Set High Mask	Send	0	0	0	0	1	0	1	1	Set high mask to *hhhhhhhh*

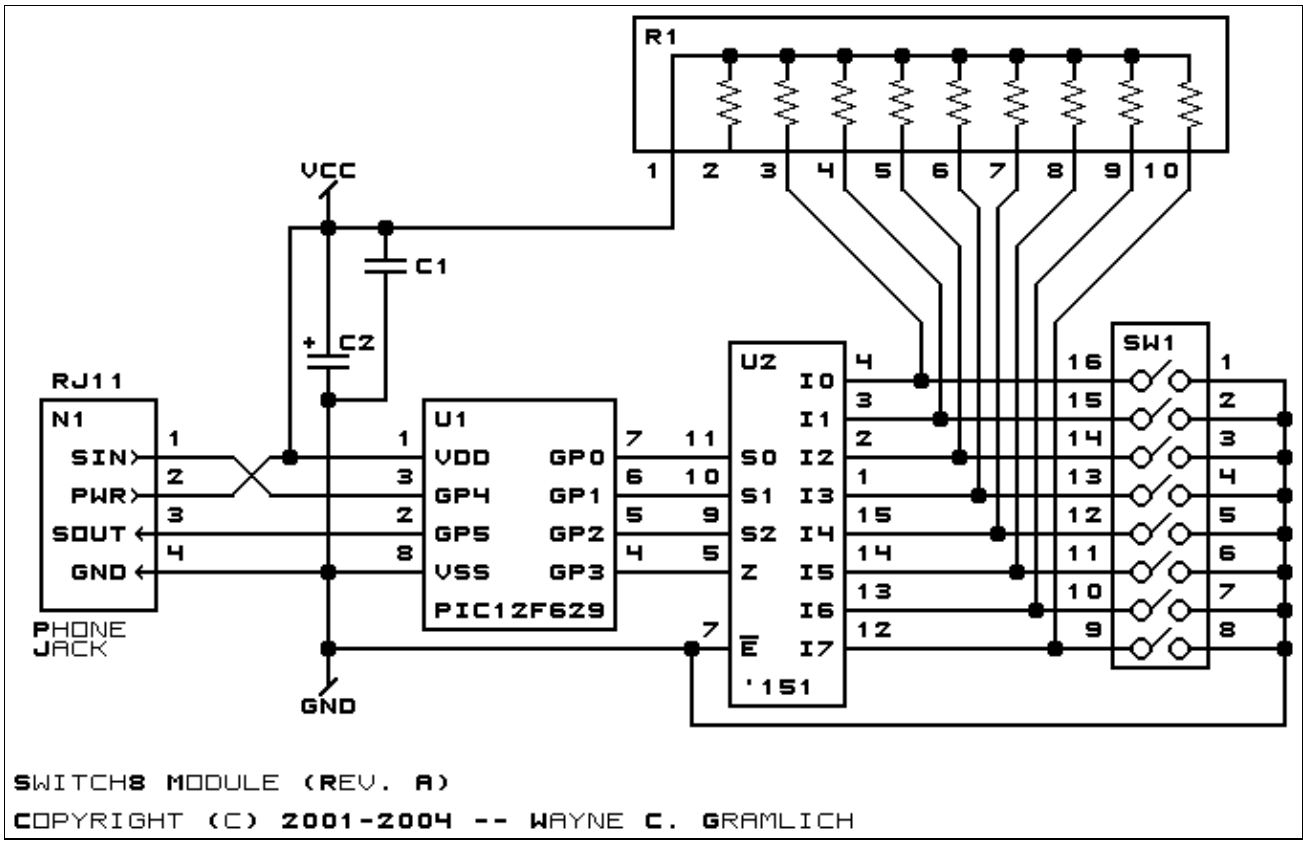
	Send	*h*	*h*	*h*	*h*	*h*	*h*	*h*	
Set Raising Mask	Send	0	0	0	0	1	1	0	0	Set raising mask to *rrrrrrrr*
	Send	*r*	*r*	*r*	*r*	*r*	*r*	*r*	*r*	
Set Falling Mask	Send	0	0	0	0	1	1	0	1	Set falling mask to *ffffffff*
	Send	*f*	*f*	*f*	*f*	*f*	*f*	*f*	*f*	
Read Interrupt Bits	Send	1	1	1	0	1	1	1	1	Return the 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*	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 Switch8 RoboBrick is shown below:



The parts list kept in a separate file −− switch8.ptl.

3.2 Printed Circuit Board

The printed circuit board files are listed below:

switch8_back.png
: The solder side layer.

switch8_front.png
: The component side layer.

switch8_artwork.png
: The artwork layer.

switch8.gbl
: The RS−274X "Gerber" back (solder side) layer.

switch8.gtl
: The RS−274X "Gerber" top (component side) layer.

switch8.gal
: The RS−274X "Gerber" artwork layer.

switch8.drl
: The "Excellon" NC drill file.

switch8.tol
: The "Excellon" tool rack file.

4. Software

The Switch8 software is available as one of:

switch8.ucl
: The µCL source file.

switch8.asm
: The resulting human readable PIC assembly file.

switch8.lst
: The resulting human readable PIC listing file.

switch8.hex
: The resulting Intel® Hex file.

5. Issues

The following issues have come up:

- The 2200 µF 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.
