

This is the Revision B version of the IRDistance8 Module. The status of this project is finished.

IRDistance8 Module (Revision B)

Table of Contents

This document is also available in PDF format.

- 1. Introduction
- 2. Programming
- 3. Hardware
 - ◆ 3.1 Circuit Schematic
 - ◆ 3.2 Printed Circuit Board
- 4. Software
- 5. Issues

1. Introduction

The IRDistance8 Module is used to connect up to 8 Sharp[®] GP2D12 IROD (InfraRed Optical Distance) measuring sensors. The GP2D12 module provides an analog voltage that is proportional to the distance (although not linearly.)

2. Programming

The IRDistance8 Module can enable zero, one or more of the AIROD's. For the ones that are enabled, it continuously reads the distance values. To conserve power, only one AIROD is powered up at a time.

The IRDistance8 Module supports Module Interrupt Protocol for those lines that are being used as inputs. 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.

IRDistance8 Module (Revision B)

In addition to the common shared commands and the shared interrupt commands, the AnalogIn4 Module supports following commands:

Command	Send/ Receive	Byte Value								Discussion
		7	6	5	4	3	2	1	0	
Read 8-bit Analog	Send	0	0	0	0	0	<i>c</i>	<i>c</i>	<i>c</i>	Return the raw 8-bit analog voltage <i>aaaa aaaa</i> for channel <i>ccc</i> .
	Receive	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>	
Read 10-bit Analog	Send	0	0	0	0	1	<i>c</i>	<i>c</i>	<i>c</i>	Return the raw 10-bit analog voltage <i>aaaa aaaa bb</i> for channel <i>ccc</i> .
	Receive	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>	
	Receive	<i>b</i>	<i>b</i>	0	0	0	0	0	0	
Read Low Threshold	Send	0	0	0	1	0	<i>c</i>	<i>c</i>	<i>c</i>	Return the low threshold distance <i>llll llll</i> for channel <i>ccc</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 Threshold	Send	0	0	0	1	1	<i>c</i>	<i>c</i>	<i>c</i>	Return the high threshold distance <i>hhhh hhhh</i> for channel <i>ccc</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 High Mask	Send	0	0	1	0	0	0	0	0	Return and return the high mask <i>hhhh hhhh</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 Low Mask	Send	0	0	1	0	0	0	0	1	Return and return the low mask <i>llll llll</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 Raising Mask	Send	0	0	1	0	0	0	1	0	Return and return the raising mask <i>rrrr rrrr</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	1	0	0	0	1	1	Return and return the falling mask <i>ffff ffff</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 Servo Steps	Send	0	0	1	0	1	0	0	0	Return the number of servo steps <i>sssss</i> .
	Receive	0	0	0	<i>s</i>	<i>s</i>	<i>s</i>	<i>s</i>	<i>s</i>	
Read Servo Base	Send	0	0	1	0	1	0	0	1	Return the 16-bit <i>hhhh hhhh llll lll</i> Servo base pulse width (in μ S).
	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>	
	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 Servo Increment	Send	0	0	1	0	1	0	1	0	Return the 16-bit <i>hhhh hhhh llll lll</i> Servo increment pulse width (in μ S).
	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>	
	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 Servo Step Delay	Send	0	0	1	0	1	0	1	1	Return the servo step delay <i>dddd dddd</i> measured in milliseconds (0 disables servo.)
	Receive	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	
Read Minimum Distance	Send	0	0	1	1	0	1	0	0	Return minimum distance from the scan <i>ddddddd</i> .
	Receive	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	
Read Minimum Distance Index	Send	0	0	1	1	0	1	0	1	Return minimum distance index <i>iiiiii</i> .
	Receive	0	0	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	
Read Maximum Distance	Send	0	0	1	1	0	1	1	0	Return maximum distance from the scan <i>ddddddd</i> .
	Receive	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	
Read Maximum Distance Index	Send	0	0	1	1	0	1	1	1	Return minimum distance index from the scan <i>iiiiii</i> .
	Receive	0	0	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	
Set Low Threshold	Send	0	1	0	1	0	<i>c</i>	<i>c</i>	<i>c</i>	

IRDistance8 Module (Revision B)

	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 the low threshold distance for channel <i>ccc</i> to <i>llll llll</i> .
Set High Threshold	Send	0	1	0	1	1	<i>c</i>	<i>c</i>	<i>c</i>	Set the high threshold distance for channel <i>ccc</i> to <i>hhhh hhhh</i> .
	Send	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	
Set High Mask	Send	0	1	1	0	0	0	0	0	Set the high mask to <i>hhhh hhhh</i> .
	Send	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	
Set Low Mask	Send	0	1	1	0	0	0	0	1	Set the low mask to <i>llll llll</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 Raising Mask	Send	0	1	1	0	0	0	1	0	Set the raising mask to <i>rrrr rrrr</i> .
	Send	<i>r</i>	<i>r</i>	<i>r</i>	<i>r</i>	<i>r</i>	<i>r</i>	<i>r</i>	<i>r</i>	
Set Falling Mask	Send	0	1	1	0	0	0	1	1	Set the falling mask to <i>ffff ffff</i> .
	Send	<i>f</i>	<i>f</i>	<i>f</i>	<i>f</i>	<i>f</i>	<i>f</i>	<i>f</i>	<i>f</i>	
Set Servo Steps	Send	0	1	1	0	1	0	0	0	Set the number of servo steps to <i>ssss</i> .
	Send	0	0	0	<i>s</i>	<i>s</i>	<i>s</i>	<i>s</i>	<i>s</i>	
Set Servo Base	Send	0	1	1	0	1	0	0	1	Set the servo base pulse width to <i>hhhh hhhh llll</i> (a 16-bit number measured in μ s).
	Send	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</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 Servo Increment	Send	0	1	1	0	1	0	1	0	Set the servo increment pulse width to <i>hhhh hhhh llll lll</i> (a 16-bit number measured in μ s).
	Send	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</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 Servo Step Delay	Send	0	1	1	0	1	0	1	1	Set the servo step delay <i>dddd dddd</i> measured in milliseconds (0 disables servo.)
	Send	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	
Read Scan Distance	Send	1	0	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	Return the distance <i>dddd dddd</i> for scan position <i>iiii</i> .
	Receive	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</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</i>	<i>p</i>	
Set Interrupt Commands	Send	1	1	1	1	0	<i>c</i>	<i>c</i>	<i>c</i>	Set Interrupt Command <i>ccc</i> .
Shared Commands	Send	1	1	1	1	1	<i>c</i>	<i>c</i>	<i>c</i>	Execute common shared command <i>ccc</i>

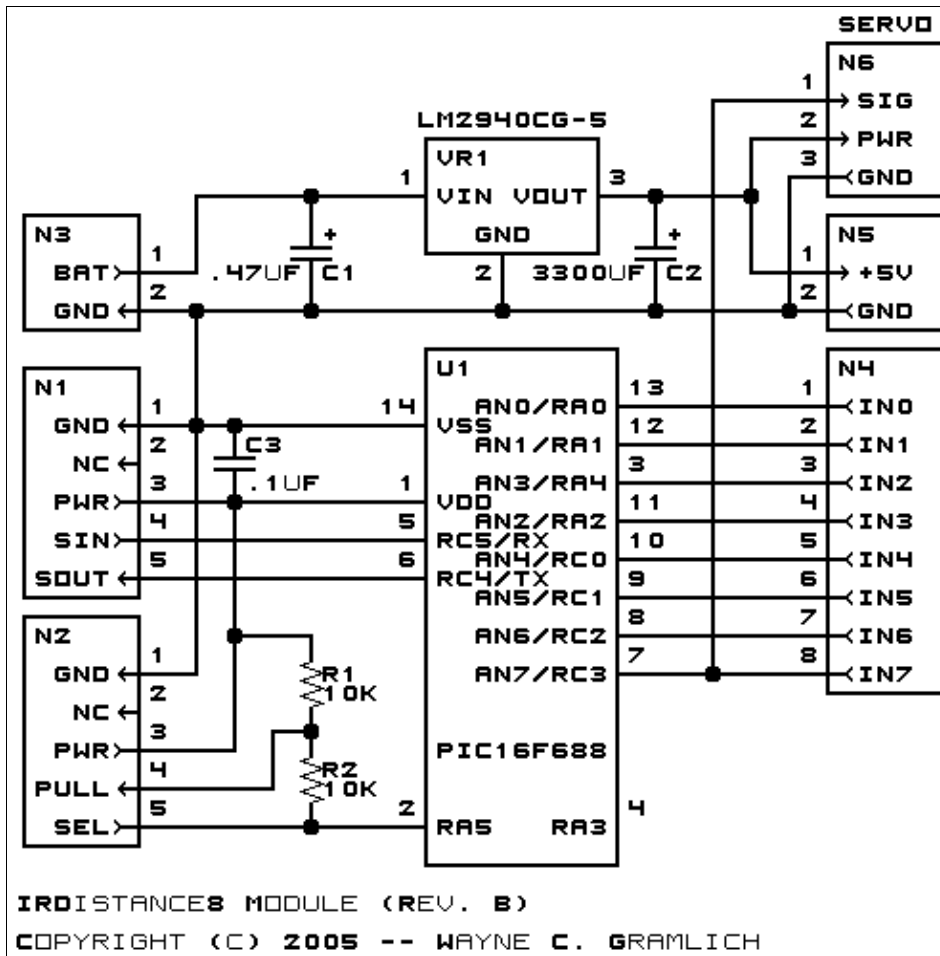
3. Hardware

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

3.1 Circuit Schematic

The schematic for the IRDistance8 Module is shown below:

IRDistance8 Module (Revision B)



The parts list kept in a separate file -- [irdistance8.ptl](#).

3.2 Printed Circuit Board

The printed circuit board files are listed below:

[irdistance8_back.png](#)

The solder side layer.

[irdistance8_front.png](#)

The component side layer.

[irdistance8_artwork.png](#)

The artwork layer.

[irdistance8.gbl](#)

The RS-272X "Gerber" back (solder side) layer.

[irdistance8.gtl](#)

The RS-272X "Gerber" top (component side) layer.

[irdistance8.gal](#)

The RS-272X "Gerber" artwork layer.

[irdistance8.drl](#)

The "Excellon" NC drill file.

[irdistance8.tol](#)

The "Excellon" tool rack file.

4. Software

The software for the IRDistance8 is listed below:

irdistance8.ucl

The μ CL file for IRDistance8.

irdistance8.asm

The assembly file for IRDistance8.

irdistance8.hex

The Intel[®] Hex file.

irdistance8.lst

The listing file for IRDistance8.

5. Issues

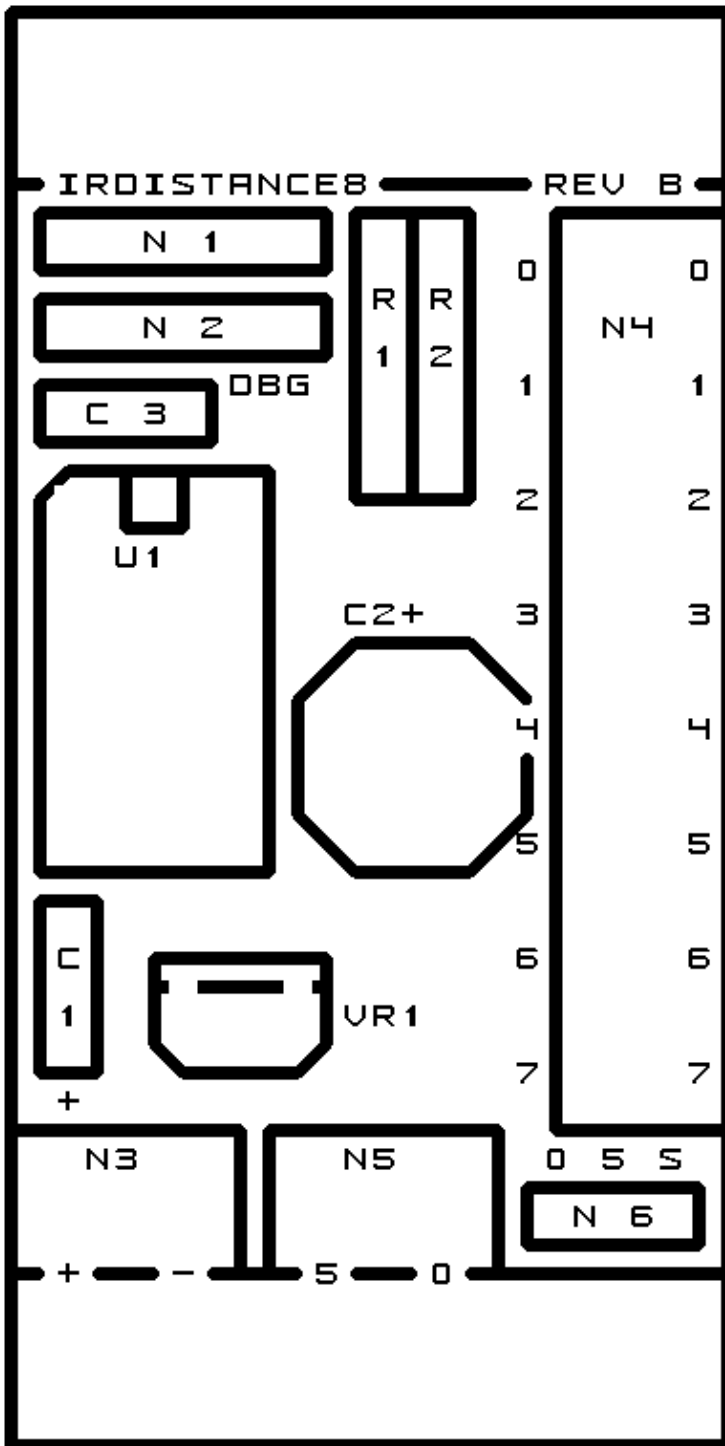
Any issues that come up will be discussed here.

Copyright (c) 2005 by Wayne C. Gramlich. All rights reserved.

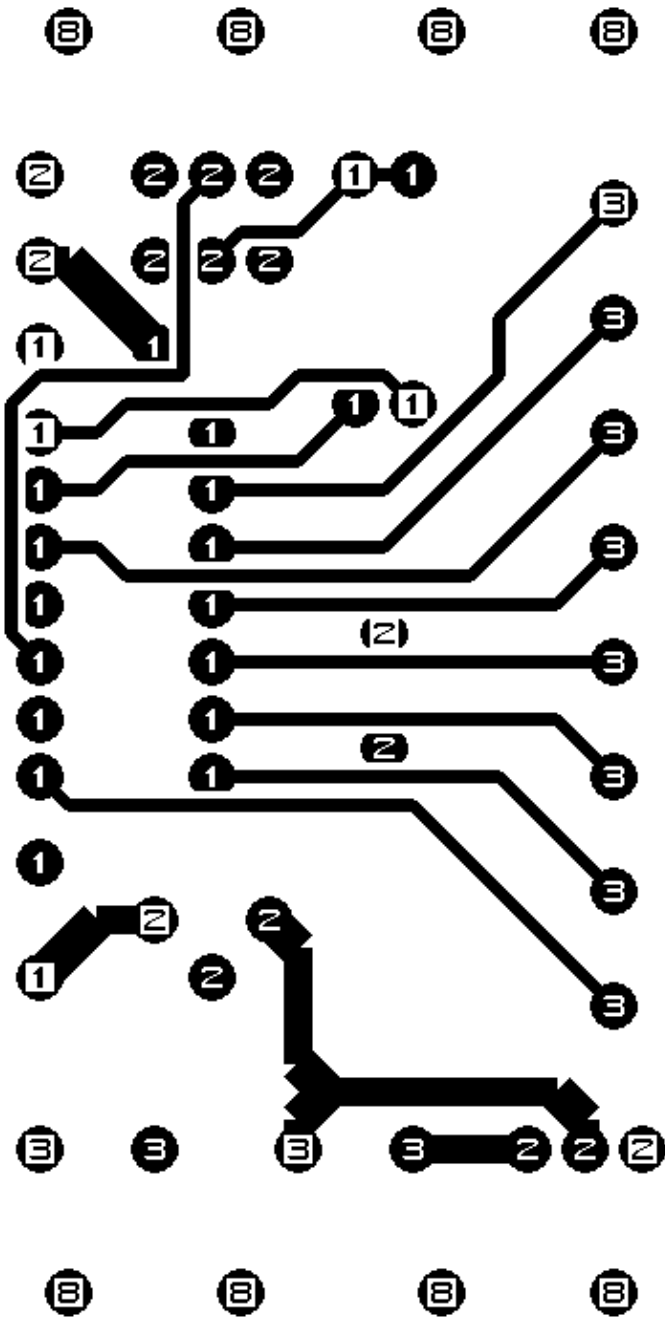
A. Appendix A: Parts List

```
# Parts list for IRDistance8 RoboBrix (Rev. B)
#
C1: Capacitor470nF - .47 uF Tantalum Capacitor [Jameco: 33531]
C2: Capacitor3300uF - 3300 uF Tantalum Capacitor [Jameco: 198897]
C3: Capacitor100nF - .1 uF Ceramic Capacitor [Jameco: 25524]
N1: Header1x5.Slave - 1x5 Male Header [5/40 Jameco: 160881]
N2: Header1x5.Debug2 - 1x5 Male Header [5/40 Jameco: 160881]
N3: TerminalStrip2.Power - 2 Junction Terminal Strip [Jameco: 189675]
N4: TerminalStrip8 - 8 Junction Terminal Strip [4 Jameco: 189675]
N5: TerminalStrip2.Power - 2 Junction Terminal Strip [Jameco: 189675]
N6: Header1x3.Servo - 1x3 Male Header [3/40 Jameco: 160881]
R1-2: Resistor10K - 10K Ohm 1/4 Watt Resistor [Jameco: 29911]
U1: PIC16F688.IRDistance8 - Microchip PIC16C688 [Digkey: PIC16F688-I/P-ND]
VR1: LM2940CG-5 - 5 Volt Low Dropout Voltage Regulator [Jameco: 107182]
```

B. Appendix B: Artwork Layer



C. Appendix C: Back (Solder Side) Layer



D. Appendix D: Front (Component Side) Layer

