

This is the Revision A version of the Digital8 module. The status of this project is finished.

Digital8 Module (Revision A)

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 Digital8 module provides the ability to input and output 8 bits of digital data. The direction of each bit can be changed under program control.

2. Programming

The programmer can download a complement mask to cause any of the bits to be complemented prior to reading.

The Digital8 module supports the 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

- \sim is bit-wise complement
- \mid is bit-wise OR
- $\&$ is bit-wise AND

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

Digital8 Module (Revision A)

The Digital8 module 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 Low	Send	0	0	0	0	0	0	0	0	Return low order 5–bits of input <i>iiii</i> (after XOR'ing with complement mask)
	Receive	0	0	0	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	<i>i</i>	
Read Inputs High	Send	0	0	0	0	0	0	0	1	Return high order 5–bits of input <i>IIII</i> (after XOR'ing with complement mask)
	Receive	0	0	0	<i>I</i>	<i>I</i>	<i>I</i>	<i>I</i>	<i>I</i>	
Read Complement Mask Low	Send	0	0	0	0	0	0	1	0	Return low order 5–bits of complement mask <i>cccc</i>
	Receive	0	0	0	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	
Read Complement Mask High	Send	0	0	0	0	0	0	1	1	Return high order 5 bits of complement mask <i>CCCC</i>
	Receive	0	0	0	<i>C</i>	<i>C</i>	<i>C</i>	<i>C</i>	<i>C</i>	
Read Direction Mask Low	Send	0	0	0	0	0	1	0	0	Return low order 5–bits of direction mask <i>dddd</i>
	Receive	0	0	0	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	
Read Direction Mask High	Send	0	0	0	0	0	1	0	1	Return high order 5 bits of direction mask <i>DDDD</i>
	Receive	0	0	0	<i>D</i>	<i>D</i>	<i>D</i>	<i>D</i>	<i>D</i>	
Read Raw Low	Send	0	0	0	0	0	1	1	0	Return low order 5–bits of raw input data <i>rrrrr</i> (without XOR'ing with complement mask)
	Receive	0	0	0	<i>r</i>	<i>r</i>	<i>r</i>	<i>r</i>	<i>r</i>	
Read Raw High	Send	0	0	0	0	0	1	1	1	Return high order 5–bits of raw input data <i>RRRRR</i> (without XOR'ing with complement mask)
	Receive	0	0	0	<i>R</i>	<i>R</i>	<i>R</i>	<i>R</i>	<i>R</i>	
Read Low Mask Low	Send	0	0	0	0	1	0	0	0	Return low order 5–bits of low mask <i>llll</i>
	Receive	0	0	0	<i>l</i>	<i>l</i>	<i>l</i>	<i>l</i>	<i>l</i>	
Read Low Mask High	Send	0	0	0	0	1	0	0	1	Return high order 5–bits of low mask <i>LLLL</i>
	Receive	0	0	0	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	
Read High Mask Low	Send	0	0	0	0	1	0	1	0	Return low order 5–bits of the high mask <i>hhhhh</i>
	Receive	0	0	0	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	
Read High Mask High	Send	0	0	0	0	1	0	1	1	Return high order 5 bits of the high mask <i>HHHHH</i>
	Receive	0	0	0	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	
Read Raising Mask Low	Send	0	0	0	0	1	1	0	0	Return low order 5–bits of the raising mask <i>rrrrr</i>
	Receive	0	0	0	<i>r</i>	<i>r</i>	<i>r</i>	<i>r</i>	<i>r</i>	
Read Raising Mask High	Send	0	0	0	0	1	1	0	1	Return high order 5 bits of the raising mask <i>RRRRR</i>
	Receive	0	0	0	<i>R</i>	<i>R</i>	<i>R</i>	<i>R</i>	<i>R</i>	
Read Falling Mask Low	Send	0	0	0	0	1	1	1	0	Return low order 5–bits of the falling mask <i>ffff</i>
	Receive	0	0	0	<i>f</i>	<i>f</i>	<i>f</i>	<i>f</i>	<i>f</i>	
Read Falling Mask High	Send	0	0	0	0	1	1	1	1	Return high order 5–bits of the falling mask <i>FFFFF</i>
	Receive	0	0	0	<i>F</i>	<i>F</i>	<i>F</i>	<i>F</i>	<i>F</i>	
Read Outputs Low	Send	0	0	0	1	0	0	0	0	

Digital8 Module (Revision A)

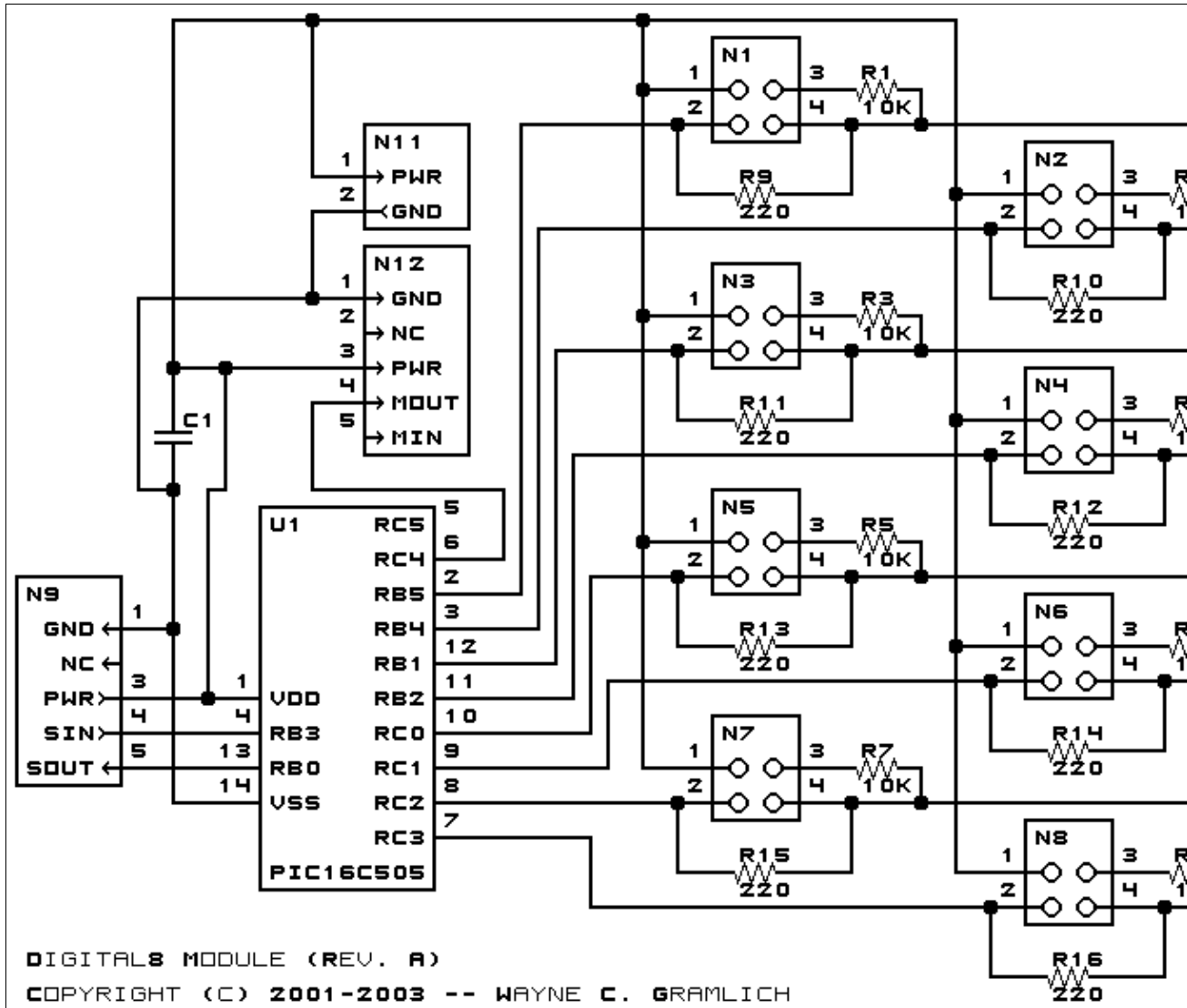
	Receive	0	0	0	<i>o</i>	<i>o</i>	<i>o</i>	<i>o</i>	<i>o</i>	Return low order 5–bits of the outputs <i>ooooo</i>
Read Outputs High	Send	0	0	0	1	0	0	0	1	Return high order 5–bits of the outputs <i>OOOOO</i>
	Receive	0	0	0	<i>O</i>	<i>O</i>	<i>O</i>	<i>O</i>	<i>O</i>	
Set Complement Mask Low	Send	0	0	0	1	0	0	1	0	Set low order 5–bits of complement mask to <i>cccc</i>
	Send	0	0	0	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	
Set Complement Mask High	Send	0	0	0	1	0	0	1	1	Set high order 5 bits of complement mask to <i>CCCC</i>
	Send	0	0	0	<i>C</i>	<i>C</i>	<i>C</i>	<i>C</i>	<i>C</i>	
Set Direction Mask Low	Send	0	0	0	1	0	1	0	0	Set low order 5–bits of direction mask to <i>dddd</i> 1=input; 0=output
	Send	0	0	0	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>d</i>	
Set Direction Mask High	Send	0	0	0	1	0	1	0	1	Set high order 5 bits of direction mask of <i>DDDDD</i> 1=input; 0=output
	Send	0	0	0	<i>D</i>	<i>D</i>	<i>D</i>	<i>D</i>	<i>D</i>	
Reset Outputs	Send	0	0	0	1	0	1	1	0	Set all 10 bits of outputs to 0
Reset Everything	Send	0	0	0	1	0	1	1	1	Reset all registers to 0 and set direction bits to 1 (input)
Set Low Mask Low	Send	0	0	0	1	1	0	0	0	Set low order 5–bits of low mask to <i>llll</i>
	Send	0	0	0	<i>l</i>	<i>l</i>	<i>l</i>	<i>l</i>	<i>l</i>	
Set Low Mask High	Send	0	0	0	1	1	0	0	1	Set high order 5–bits of low mask to <i>LLLL</i>
	Send	0	0	0	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	
Set High Mask Low	Send	0	0	0	1	1	0	1	0	Set low order 5–bits of the high mask to <i>hhhh</i>
	Send	0	0	0	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	<i>h</i>	
Set High Mask High	Send	0	0	0	1	1	0	1	1	Set high order 5 bits of the high mask to <i>HHHHH</i>
	Send	0	0	0	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	
Set Raising Mask Low	Send	0	0	0	1	1	1	0	0	Set low order 5–bits of the raising mask to <i>rrrr</i>
	Send	0	0	0	<i>r</i>	<i>r</i>	<i>r</i>	<i>r</i>	<i>r</i>	
Set Raising Mask High	Send	0	0	0	1	1	1	0	1	Set high order 5 bits of the raising mask to <i>RRRR</i>
	Send	0	0	0	<i>R</i>	<i>R</i>	<i>R</i>	<i>R</i>	<i>R</i>	
Set Falling Mask Low	Send	0	0	0	1	1	1	1	0	Set low order 5–bits of the falling mask to <i>ffff</i>
	Send	0	0	0	<i>f</i>	<i>f</i>	<i>f</i>	<i>f</i>	<i>f</i>	
Set Falling Mask High	Send	0	0	0	1	1	1	1	1	Set high order 5–bits of the falling mask to <i>FFFF</i>
	Send	0	0	0	<i>F</i>	<i>F</i>	<i>F</i>	<i>F</i>	<i>F</i>	
Set Outputs Low	Send	0	0	1	<i>o</i>	<i>o</i>	<i>o</i>	<i>o</i>	<i>o</i>	Set low order 5–bits to <i>oooo</i>
Set Outputs High	Send	0	1	0	<i>O</i>	<i>O</i>	<i>O</i>	<i>O</i>	<i>O</i>	Set high order 5–bits to <i>OOOO</i>
Set Output Bit	Send	0	1	1	<i>v</i>	<i>b</i>	<i>b</i>	<i>b</i>	<i>b</i>	Set output bit <i>bbbb</i> to <i>v</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>	
<u>Set Interrupt Commands</u>	Send	1	1	1	1	0	<i>c</i>	<i>c</i>	<i>c</i>	Set Interrupt Command <i>ccc</i> .
<u>Shared Commands</u>	Send	1	1	1	1	1	<i>c</i>	<i>c</i>	<i>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 Digital8 module is shown below:



The parts list kept in a separate file --- [digital8.ptl](#).

3.2 Printed Circuit Board

The printed circuit files are listed below:

[digital8_back.png](#)

The solder side layer.

[digital8_front.png](#)

The component side layer.

digital8_artwork.png

The artwork layer.

digital8.gbl

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

digital8.gtl

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

digital8.gal

The RS-274X "Gerber" artwork layer.

digital8.drl

The "Excellon" NC drill file.

digital8.tol

The "Excellon" tool rack file.

4. Software

The Digital8 software is available as one of:

digital8.ucl

The μ CL source file.

digital8.asm

The resulting human readable PIC assembly file.

digital8.lst

The resulting human readable PIC listing file.

digital8.hex

The resulting Intel[®] Hex file.

5. Issues

The following fabrication issues came up:

- N12 is too tight between N11 and U1.

Copyright (c) 2001–2004 by Wayne C. Gramlich. All rights reserved.