This is the Revision D version of the <u>Servo4 module</u>. The status of this project is <u>finished</u>.

Servo4 Module (Revision D)

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
- <u>5. Issues</u>

1. Introduction

The Servo4 module allows for the control of up to 4 hobby grade servos.

2. Programming

The Servo4 module can independently control up to 4 servos. Each servo has 1) an enable bit and 2) a current position. The position is represented as an 8-bit number. Some experimentation is needed to determine how the 8-bit numbers correspond to actual servo positions. All servos are initialized to have the enable flags *off*.

The Servo4 commands are summarized in the table below:

Command	Send/ Receive	Byte Value								Discussion
		7	6	5	4	3	2	1	0	Discussion
Set High	Send	0	0	h	h	h	h	s	s	Set high order 4 bits of servo ss to hhhh and set the remaining 4 low order bits to zero.
Set Low	Send	0	1	l	l	l	l	S	S	Set the low order 4 bits of servo ss position to <i>IIII</i> .
Increment	Send	1	0	0	i	i	i	s	s	Add <i>iii</i> to the position of servo ss.
Decrement	Send	1	0	1	d	d	d	s	s	Subtract <i>ddd</i> from the position of servo <i>ss</i> .
Set	Send	1	1	0	0	0	e	S	S	Select servo ss and set its position to ppppppp
Position/Enable	Send	p	p	p	p	p	p	p	p	and enable flag to e .
Set Enable Flag	Send	ı								Select servo ss and set its enable flag to e.
Read Position	Send	1	1	0	1	0	0	s	s	Return the current position pppppppp for servo
	Receive	p	p	p	p	p	p	p	p	SS.
Read Enable	Send			0					c	Return the enable bit e for servo ss .
	Receive	0	0	0	0	0	0	0	e	
Read Enables	Send	1	1	0	1	1	0	0	0	Return the enable flags <i>eeee</i> for all four servos.
	Receive	0	0	0	0	e	e	e	e	

Servo4 Module (Revision D)

Set Enables	Send Send	1	1	0	1	1 e	0 e	$\frac{0}{e}$	1 e	Set enable flags for all four servos to eeee.
Shared Commands	Send									Execute shared command ccc.

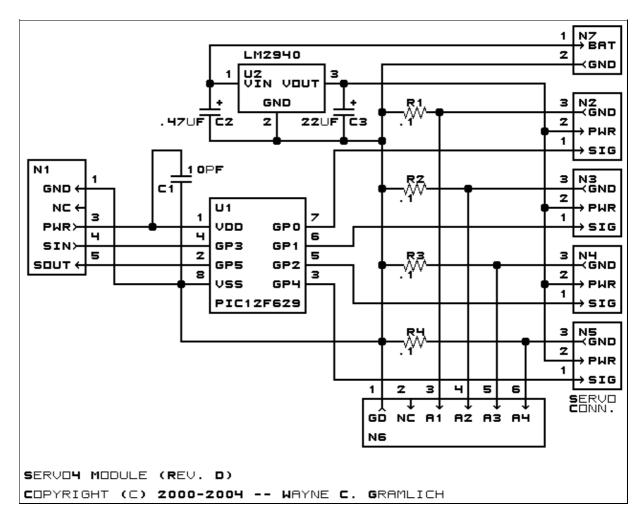
The Servo4 module does *not* know the minimum and maximum extent for each servo. This has to be determined by experimentation.

3. Hardware

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

3.1 Circuit Schematic

The schematic for the Servo4 module is shown below:



The parts list kept in a separate file — <u>servo4.ptl</u>.

3. Hardware 2

3.2 Printed Circuit Board

The printed circuit board files are listed below:

servo4 back.png

The solder side layer is shown below:

servo4 front.png

The component side layer is shown below:

servo4 artwork.png

The artwork layer is shown below

servo4.gbl

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

servo4.gtl

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

servo4.gal

The RS-274X "Gerber" artwork layer.

servo4.drl

The "Excellon" NC drill file.

servo4.tol

The "Excellon" tool rack file.

4. Software

The Servo4 software is available as one of:

servo4.ucl

The µCL source file.

<u>servo4.asm</u>

The resulting human readable PIC assembly file.

<u>servo4.lst</u>

The resulting human readable PIC listing file.

servo4.hex

The resulting Intel[®] Hex file.

5. Issues

The following software issues have came up:

- There is a request for enhancement from William Hubbard for the ability to set "set points" and a command to "return to set point". Reasonable request; it might even fit.
- William Hubbard is requesting the ability to delay servo changes until a single command is sent. Reasonable request; it might be a tight fit.

Copyright (c) 2000–2004 by Wayne C. Gramlich. All rights reserved.