Ocean Controls RC Servo Motor Controller

RC Servo Motors:
RC Servo motors are used in radio-controlled model cars and planes, robotics, special effects, test equipment and industrial automation. At the hobbyist end of the market they are small, compact and relatively inexpensive at around $US20. The motors themselves are black boxes which contain a motor, gearbox and decoder electronics. Three wires go into the box; 5V, ground and signal (usually read, black and white wires respectively). A short shaft comes out of the motor which usually has a circular interface plate attached to it. Most servos will rotate through about 100 degrees in less than a second according to the signal input. The signal input is a Pulse Code Modulation system. The signal is a 5V pulse between 1 and 2 msec long repeated 50 times per second. That is, a 20msec frame rate. The width of the pulse determines the position of the server. Most servos will move to the center of their travel when they receive a 1.5msec pulse. One extreme of motion generally equates to a pulse width of 1.0msec; the other extreme to 2.0msec with a smooth variation throughout the range, and neutral at 1.5msec. The period between the pulses is used to synchronise the receiver.

RC Servo Motor Controller:
The Ocean Controls RC Servo Motor Controller will control up to 8 servo motors. 4 of the servo motors are controlled by ASCII RS232 serial commands. This can be increased to 8 when the configuration is set to 1 (see command WRCF.)
The form of the command is @xx CMND yyy where xx is the address of the motor, CMND is the command and yyy is a parameter value. **Please note the command must terminate in a carriage return**

For example @01 WRSP 200 sets the pulse width of motor 1 connected to terminal D1. 200 is the width of the pulse in increments of 10usec.
For example @02 WRSP 100 will output a pulse width of 1msec to motor 2 connected to terminal D2.

The other 4 servo motors are controlled by the position of a switch. A switch is connected between the Lx and COM inputs. The user then writes a series of commands to the controller to tell the controller what positions the motor is to move to when the switch is in the OPEN and CLOSED states. The user can also set the rate of movement between the OPEN and CLOSED positions. These variables are automatically saved in EEPROM so that they will not be lost on loss of power.

For example @01 WRON 200 will produce a 2 msec pulse when the switch 1 is ON or CLOSED, @01 WROF 100 will produce a 1 msec pulse when the switch 1 is OFF or OPEN

Initial Setup:
All connections and descriptions are outlined in Figure 1 and Table 1.
Connect 12 Volts across the Vs and COM terminals, with positive on the Vs terminal.
Connect the signal output (D1 to D4 serial data command control) (S1 to S4 switch control) to the signal inputs on the servo motors.
Using a **D9 Male to Female straight through cable** connect the Female D9 connector on the Serial Controller to the RS232 Serial Port on the computer**.**
Connect additional controller boards to the K3 RS232 connector using a 9-pin RS232 serial cable or simply align K3 with K2 of the next board and push the boards together.
Finally, address the controller boards using the DIP switches S1 and S2. This determines the numbers used when communicating via the computer. Details are provided in Table 2.

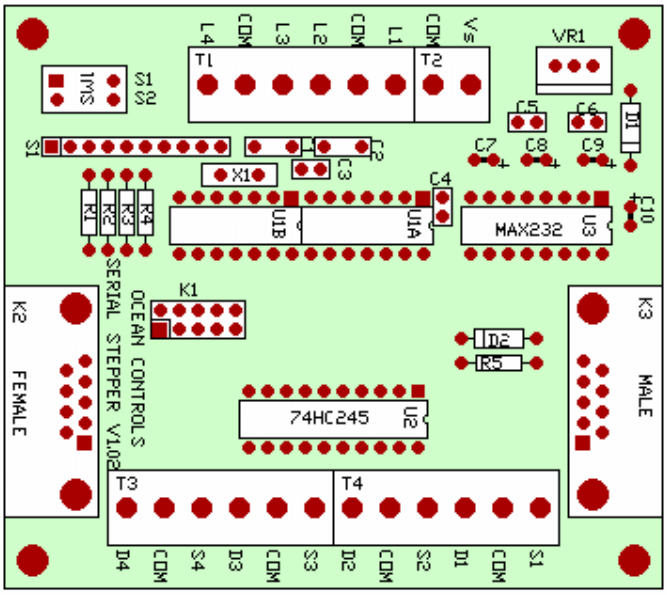

Figure 1 – Ocean Controls RC Servo Motor Controller

Table 1- Serial Stepper Motor Controller Connections

Abbreviation	Description
Vs	12V input
COM	Common/Ground
Lx	Switch input to control servo motor connected to Dx
Sx	Switch Signal output for servo motor x
Dx	Serial Data output for servo motor x
K1	ICSP Programming Connection (Not Used)
K2	Female RS232 Connector to computer
K3	Male RS232 Connector to additional controller boards

Table 2 – Addressing

S2	S1	Motor Numbers
OFF	OFF	01-04
OFF	ON	05-08
ON	OFF	09-12
ON	ON	13-16

Using the Controller:

The Ocean Controls RC Servo Motor Controller is controlled via the serial port of a personal computer, using any serial terminal software or custom software, set at a baud rate of 9600baud, No Parity and 1 Stop bit.

The commands for the controller are in the form:

@AA CMND XXX

Where AA is the 2 digit number of the motor being addressed, between 01 and 16 (see Table 2), CMND is the 4 letter command, refer to Table 3 for available commands, and XXX is a numeric value associated with the command, refer to Table 3 for detail.

Table 3 – Commands

Command	Description
WRON	Set the switch ON position where XXX is between 0 and 100
WROF	Set the switch OFF position where XXX is between 0 and 100
WRRA	Is the rate of position change every 25msec when the servo motor moves from the ON angle to the OFF angle and vice versa.
RDON	Reads the switch ON Position

RDOF	Reads the switch OFF Position
RDRA	Reads the rate of change
WRDP	Writes the motor position of the switch controlled servos S1 to S4
RDDP	Reads the motor position of the switch controlled servos S1 to S4
WRSP	Writes the motor position of the serial controlled servos D1 to D4
RDSP	Reads the motor position of the serial controlled servos D1 to D4
WRCF	Write Configuration Bit 1=0 results in 4 Switch controlled servos, 4 Serial controlled servos Bit 1=1 results in 8 Serial controlled servos
RDIP	Read the status Bits 0-3 = the 4 inputs L1 to L4

When a valid command is received by the unit it will respond with the address preceded by a hash symbol, ie. **#AA** and this will be followed by a value if it is requested.

WRCF Command
If the command @01 WRCF 0 is issued -
The position of the servos connected to S1,S2,S3 and S4 is controlled by the switches connected to terminals L1, L2, L3 and L4
The position of the servos connected to D1,D2,D3 and D4 is controlled by the serial command #AA WRSP YYY

If the command @01 WRCF 1 is issued -
The position of the servos connected to S1,S2,S3 and S4 is controlled by the serial command #AA WRDP YYY
The position of the servos connected to D1,D2,D3 and D4 is controlled by the serial command #AA WRSP YYY

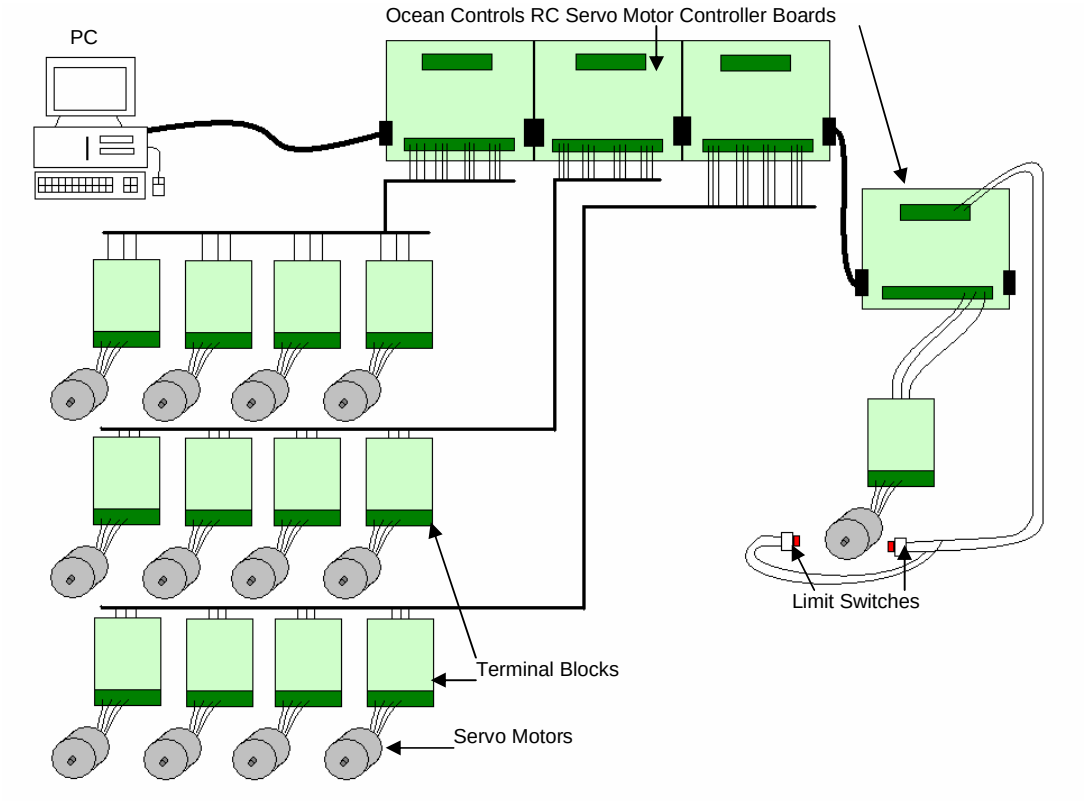

Figure 4 - Multiple Controller Connection Scheme

Multiple Controllers:
Figure 4 shows a schematic diagram of a possible connection scheme for connecting multiple controller boards.
The controller boards can be ganged together or they can be located in a different position by using a cable.
The first board in the chain is connected to the computer.
Each controller can have multiple servo motors.

Assembly & Testing the Kit:
All the components are mounted on the dual layer PCB as in Figure 1

Start by soldering the resistors and diodes and then the capacitors and crystal. Next add the IC sockets, voltage regulator and DIP switch. Finally add the terminals and 9 pin D connectors, noting the placement of the male and female connectors.
If you are going to gang two or more controller boards together you will need to unscrew the screw connectors on the right hand side of all the D9 connectors. This will allow the connectors to mate when pushed together.
See Figures 7a and 7b for the screw connectors.

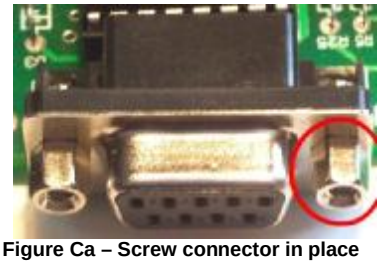
Figure Ca – Screw connector in place

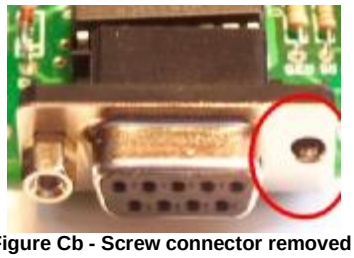
Figure Cb - Screw connector removed

To test the controller, connect power to the relevant terminals and measure the voltage across pins 10 and 20 ensuring that it is 5V. If this is ok remove the power and insert the microcontroller, max232 into their sockets. Solder the 74HC245 IC into position. Reconnect power and connect the board to the computer using an RS232 cable. Set the DIP switches so they address from 01 to 04 (both off) and then run a terminal program at 9600 baud and type **@01 RDIP**. This should return **#01 0** if the kit is working properly and nothing else is connected. If you have an oscilloscope you can give a move command such as **@01 WRSP 200** and view the pulses about 2msec wide at terminal D1 on the oscilloscope.

Parts List:
1 28-pin DIP sockets (U1A, U1B)
1 16-pin DIP socket (U3)
1 2-way DIP switch (SW1)
1 D9 Female right angle connector (K2)
1 D9 Male right angle connector (K3)
6 3-way terminal blocks (T1, T3-T4)
1 2-way terminal block (T2)
1 20MHz crystal (X1)

Semiconductors:
1 Atmel Mega168 microcontroller (U1)
1 74HC245 Octal Buffer (U2)
1 MAX232 RS232 to TTL Level Shifter (U3)
1 7805 5V Voltage Regulator (VR1)
1 1N4004 silicon diode (D1)
1 1N4148 silicon diode (D2)

Resistors:

1 10K SIL pull up network (S1)
4 10K (R1-R4)
1 18K (R5) maybe substituted with a 20K or 22K

Capacitors:
2 22pF ceramic (C1, C2)
4 0.1uF monolythic (C3-C6)
4 1uF electrolytic (C7-C10)