

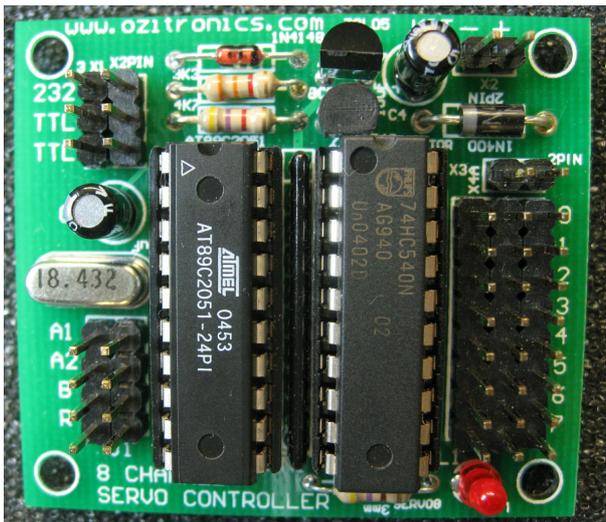
K1001 – 8 Channel Serial Servo Controller

INTRODUCTION

K1001 is an 8-Channel Serial Servo Controller based on the Atmel 89C2051 microcontroller. It accepts RS232 or TTL serial data from a computer or microcontroller and outputs PWM (pulse width modulated) signals to control up to eight RC servo motors (as used in radio-controlled model airplanes, cars, etc.). Unused servo pins can be reconfigured for digital output to drive on/off devices.

Up to four controllers can be multi-dropped on a single serial port for controlling up to 32 servo motors – more on request.

K1001 is supplied fully assembled and tested.



FEATURES

- Independent control of 8 servo motors
- Up to 4 controllers can be multi-dropped to control up to 32 servos (more on request)
- Unused pins can be configured as digital outputs
- Controlled via serial port – RS232 or TTL
- Selectable baud rates – 2400 or 9600
- Selectable pulse width range
- Simple 3-byte commands
- Compatible with industry standard **Mini SSC II** servo controller by Scott Edwards Electronics
- Power on LED - blinks when valid command received

SPECIFICATIONS

Power supply	8 to 24VDC @ 17mA
Serial input	RS232 or inverted TTL
Baud rate.....	2400 or 9600, 8N1
Pulse frequency.....	~ 50 Hz
Pulse width range	Normal - 1.0 to 2.0 mS Extended - 0.5 to 2.5 mS
Pulse width (startup)	1.5 mS
I/O voltage.....	0V and 5V
I/O current	High (source) – 6 mA Low (sink) – 20 mA
Size.....	4.9cm x 4.3cm (1.9" x 1.7")

CONNECTIONS (Refer to the wiring diagram on page 3)

Power – Controller Board

A **8 to 24VDC** power supply is required to power the controller board. This is connected to the 2-pin header labelled 'X2' (top right). The board is protected against reverse polarity power connection.

Power – Servo Motors

The servos require their own power supply with a nominal range of **4.8 to 6.0 volts DC**. This supply is connected to the 2-pin header labelled 'X3' (just below diode D2). The servo supply should be a high current supply, capable of delivering several amps.

If multiple servos are moving together then this could be as high as 10 amps (refer to servo motor specs). In this case the power and ground pins for each servo should be wired directly to the supply and only use the signal pins on the controller board.

Take care when connecting this supply as servos can be destroyed if the voltage is too high or reversed.

Servo Motors

Servo motors are connected to the 3-pin headers labelled '0 – 7'. The signal wire, usually yellow or white, should be toward the inside of the board (next to the IC). The red wire should be in the middle pin and the black wire to the outside of the board (closest to the edge).

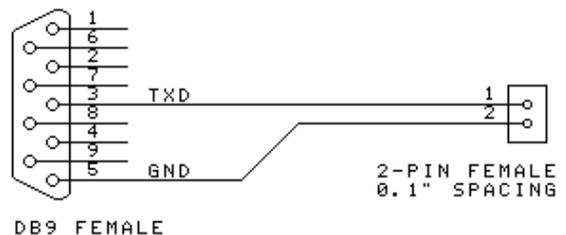
It is possible to plug the servo in backwards. Take care to connect it correctly to avoid damage.

Serial connections

The controller only requires two connections to a computer or microcontroller – serial data and ground.

There are three pairs of header pins for making the serial connection. These are labelled 'X1' and are in the top left corner of the board. In each case the 'ground' pin of each pair is the leftmost pin closest to the edge of the board.

When connecting to a computer or other RS232 port use the 2-pin pair labelled 'RS232'.



When connecting to a microcontroller or single-board computer such as the Basic Stamp® use one of the 2-pin pairs labelled 'TTL'.

Note: There are **two** pairs of pins labelled 'TTL'. This enables multiple controller boards to be multi dropped on the same serial input. Refer to '**Connecting Multiple Controllers**' on page 3 for details.

K1001 – 8 Channel Serial Servo Controller

CONFIGURATION (JUMPER SETTINGS)

The default configuration (no jumpers) is:

2400 baud, servos 0 to 7, pulse width = 1 to 2 mS

To change a setting, insert a jumper across the appropriate pair of header pins. Changing the baud rate takes effect the next time the controller is powered up. The other settings take effect immediately.

Addresses (A1, A2)

The following table shows the range of servo numbers that the controller board responds to when jumpers are installed onto these pins.

A1	A2	Servos
Off	Off	0 – 7
On	Off	8 – 15
Off	On	16 – 23
On	On	24 - 31

Up to four servo controllers can be connected to the same serial port and each servo to be individually addressed.

It is possible to have more than four controllers connected together. It requires reprogramming of the microcontroller – please ask for details.

Baud rate (B)

B	Baudrate
Off	2400
On	9600

The data format is fixed at 8N1 – 8 data bits, no parity bit, 1 stop bit.

Changing this jumper setting takes effect next time the controller is powered up.

Pulse width range (R)

This sets the range of the output pulse width. All the servo outputs are affected.

R	Range	Step size (Range / 254)
Off	1.0 – 2.0 mS	3.937 uS
On	0.5 – 2.5 mS	7.874 uS

Servo positions are expressed in units from 0 to 254, with '0' being the minimum pulse width and '254' the maximum. In either range a value of '127' gives a 1.5mS pulse width (middle position).

The 'standard' range of motion of a servo is 90° when using pulse widths from 1 – 2 mS. However many servos are capable of more than 90° of motion, with some as much as 180°. The extended range of pulse width (jumper on) allows the controller to drive the servo over its entire range.

Caution when using extended range!

Some servos cannot move through a 180° range. It is possible that using the extended range of pulse widths could drive the servo past its physical limits at each end, causing the servo to stall or even be damaged.

When using extended range, start at the middle (127) position then gradually work out the upper and lower limits that the servo can take before it stalls. You can then restrict the servo position values to this safe range.

Note: The middle position is the default when the controller is powered up.

SENDING COMMANDS TO THE CONTROLLER

There are two types of commands that can be sent to the controller:

- Servo Control command
- Digital Output Control command

In both cases the command is a three byte sequence:

<header byte> <servo number> <position>

Note: These are HEX byte values, not text strings representing the numbers.

The **onboard LED** is normally on and blinks off when a valid command is received. Handy when debugging as it indicates that the controller is receiving valid data.

Servo Control command

Three byte command as follows:

1. header byte = 255 (FFh)
2. servo number = 0 to 255 (00h to FFh)
3. position = 0 to 254 (00h to FEh)

By default all outputs are configured for servo control on power up. A position value of '255' is invalid and ignored.

Example: Set servo number 5 to position 100.

<255><5><100>

Unused servo outputs can be configured to be digital outputs by sending a 'Digital Output Control command', as follows:

Digital Output Control command

Three byte command as follows:

1. header byte = 254 (FEh)
2. servo number = 0 to 255 (00h to FFh)
3. position (0 = low, 1 = high)

Any servo signal on that output is immediately stopped (if running) and the output set high or low accordingly. Position values other than 0 or 1 are ignored.

To configure the output for servo control again simply send it a valid 'Servo Control command' (see above).

Example: Configure servo number 7 as a digital output and make it high.

<254><7><1>

K1001 – 8 Channel Serial Servo Controller

CONNECTING MULTIPLE CONTROLLERS

When connecting multiple controllers together each controller **must** be set to the same baud rate.

The first controller in the chain is connected to the serial line using either the RS232 connection OR one of the TTL connections. Then connect a spare TTL pair on the first controller to a TTL pair on the second, then second to third and so on. See the wiring diagram below.

The connections between controllers use TTL level signals – keep wire leads should be kept as short as possible.

Note: The RS232 connection is only used on the first controller (if needed) but not on any of the others.

By default up to four controllers can be connected on the same serial line, controlling up to 32 servos. For controlling more than 32 servos you need to order special controllers that allow higher address ranges – please ask for details.

POWERING UP

When board power is applied the LED will turn on. The LED indicates that power is applied and that the controller is working properly and waiting for a command. When servo power is applied all connected servos will move to their centre position.

The LED will remain on until a valid command is received, when it will blink off momentarily.

TESTING

Enter the following **QBasic** program. Connect a servo to one of the outputs and run the program. You can then enter position data to move the servo from end to end. Use 'Ctrl-Break' to end the program.

```
OPEN "COM1:2400,N,8,1,CD0,CS0,DS0" FOR OUTPUT AS #1
CLS
DO
  PRINT: PRINT "K1001 Serial Servo Controller"
  INPUT "Servo Number :"; Servo
  INPUT "Servo Position :"; Position
  PRINT #1, CHR$(255); CHR$(Servo); CHR$(Position);
LOOP
```

Note the use of the CHR\$ function to convert a number to its corresponding ASCII character.

For information about QBasic or where to get it go to <http://www.tedfelix.com/qbasic>

SAMPLE CODE

Each code sample moves servo 3 to position 100. No jumpers installed.

BASIC Stamp 1

```
SEROUT 0,N2400,(255,3,100)
```

BASIC Stamp 2

```
SEROUT 0,$418D,(255,3,100)
```

Visual Basic

For an excellent article about writing your own Visual Basic code go to <http://www.rentron.com/Mini-ssc.htm>

INFORMATION ABOUT SERVO MOTORS

There are many websites containing information about servo motors and how they work. The following are just a few:

www.societyofrobots.com/actuators_servos.shtml

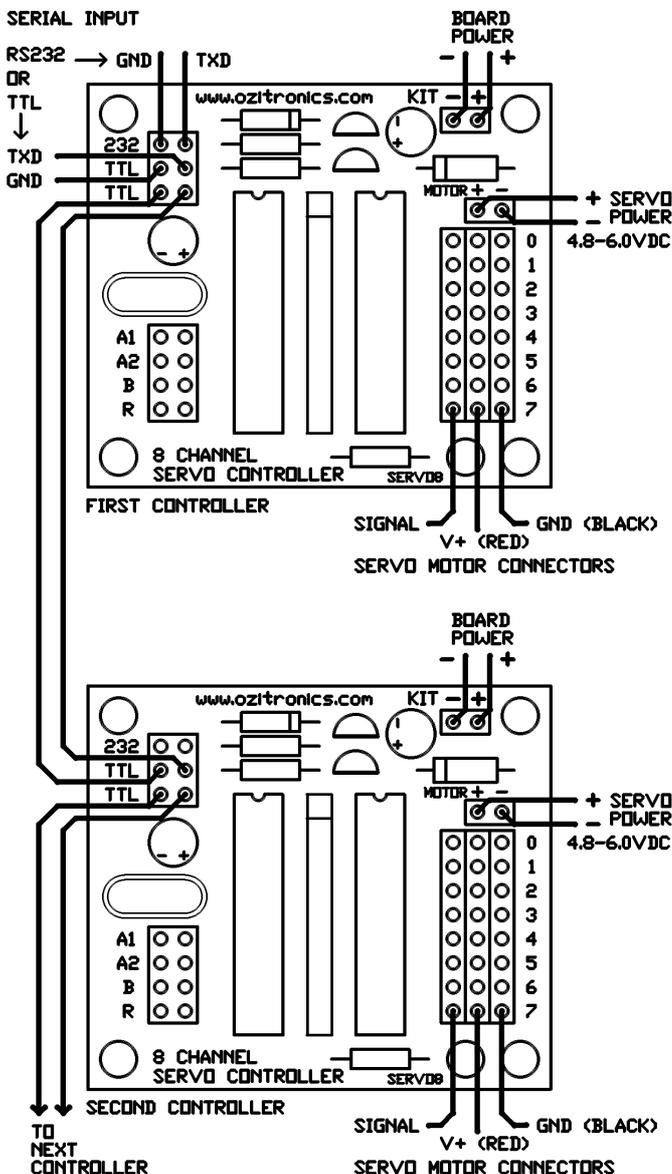
www.seattlerobotics.org/guide/servos.html

http://en.wikipedia.org/wiki/RC_Servo

www.repairfaq.org/filipg/RC/F_Servo101.html

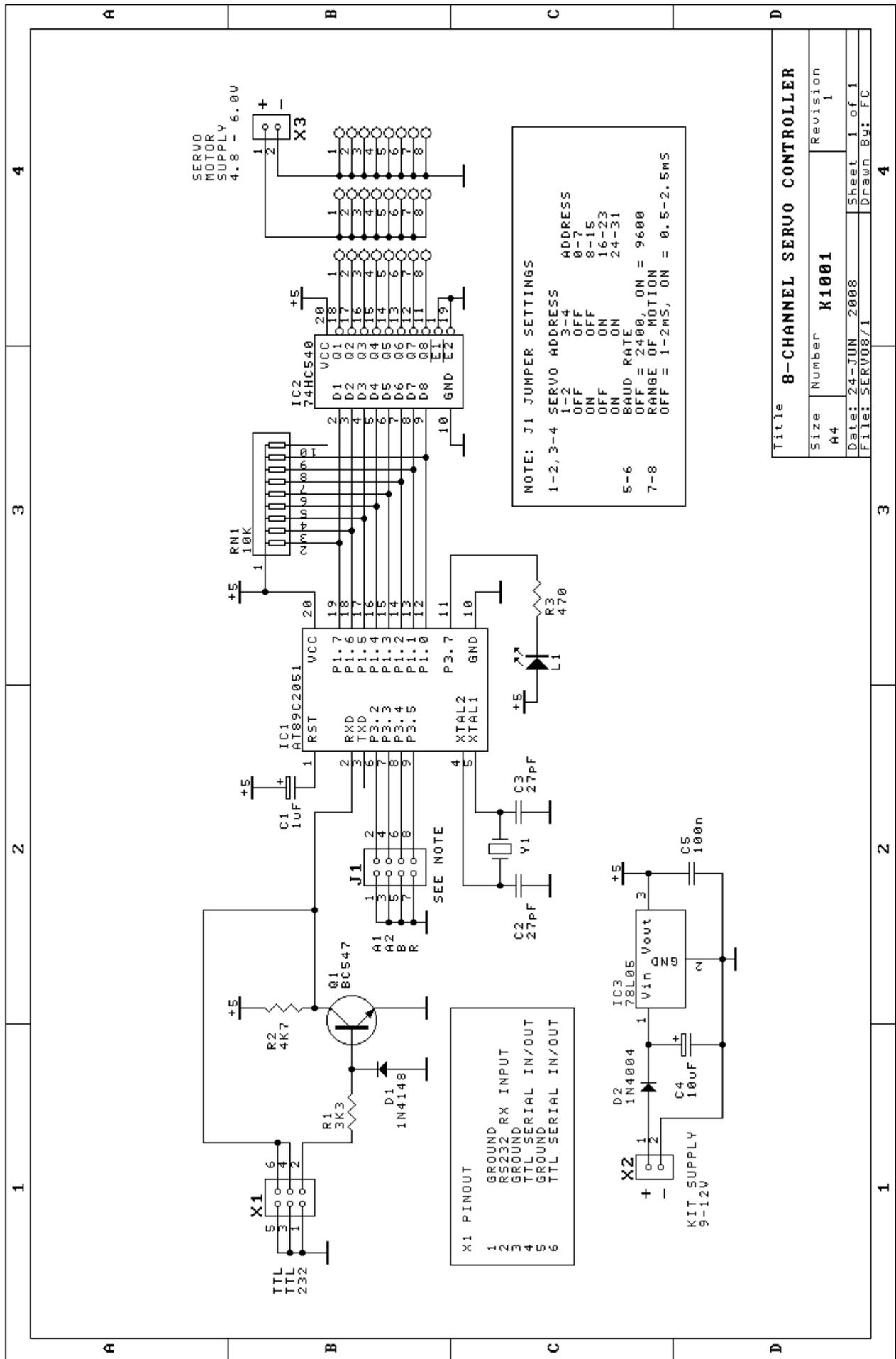
CONTACT DETAILS

You can contact the developer via the email link on our website at www.ozitronics.com



Wiring Diagram

K1001 – 8 Channel Serial Servo Controller



Title 8-CHANNEL SERVO CONTROLLER		
Size A4	Number K1001	Revision 1
Date: 24-JUN 2008	Sheet 1 of 1	Drawn By: FC
File: SERV08/1		