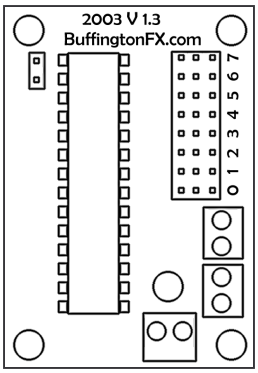
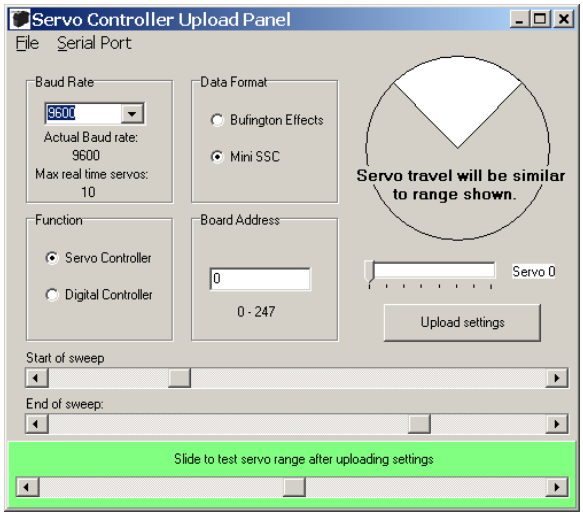
RC Servo Controller v 1.3



The Buffington Effects RC Servo controller is an electronic device that allows you to control eight remote control servos such as those used in model airplanes. Throughout its design and development it has been made with animatronics in mind.

Here are some features of the Buffington Effects RC Servo Controller:

- The Buffington Effects RC Servo Controller allows you to change the range of travel of your servo. This is an extremely useful function when making multiple animatronics that need to move in exactly the same way. You will never again need to fool around with your linkages!
- Internally this servo controller is controlling the servo at slightly less than 13-bit precision, which means that even if you scale the output range down, you will still be able to command it to 256 distinct positions.
- The baud rate can be adjusted to fit your custom application. Any standard baud rate from 4800 to 115200 baud can be chosen. Non-standard baud rates are also available.
- Supports the BFX (Buffington Effects) data format which allows you to drive up to 378 servos at 30 updates per second. If update speed is not important then you can drive up to 65536 servos total at a slower update rate.
- Supports the Mini SSC data format for compatibility with legacy programs.
- The Buffington Effects RC Servo Controller can also act as a digital controller. In this mode, each servo output becomes an on/off output instead. This allows you to control things such as lights, relays, motors, or simple pneumatics using a small amount of external circuitry.



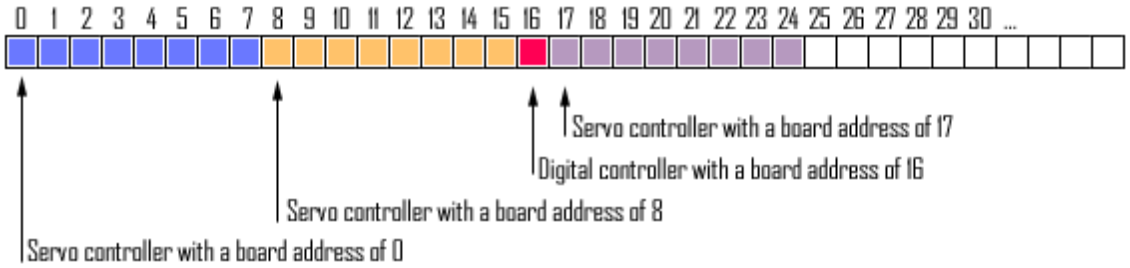
When you first receive your servo controller it will be configured to drive servos through a 90-degree arc. If you are using this in an animatronic that was previously driven by a remote control radio then you may be ready to go with no change to your servo controller.
There is a utility located on the Buffington Effects website that allows you to change the way that the servo controller acts. This software is located at: http://www.buffingtonfx.com/downloads/ConfigUtility.zip

This utility is simple to use. Select your desired baud using the drop-down list or by entering it in the drop-down list. Now select the data format and function using the radio buttons. Next you will need to choose your board address. The board address determines which servo numbers are driven by this board. This controller can drive eight servos. If you want to drive more than eight servos you will need to use one or more boards with different board addresses. For example, the first servo controller could have an address of 0 so it would drive servos 0 through 7. Setting the next controller's board address to 8 would allow it to drive servos 8 through 15. If you were to add another board, its address could be 16. You don't have to have the addresses be consecutive. You could have one board drive servos 0 through 8 and another driving servos 26 through 33 if you want. It would also be perfectly acceptable to have two boards drive servos 0 through 8 if you want. This would allow you to have two mechanisms that moved in exactly the same way. A quick way to figure out what the servo number for any given servo is, is to add the base address of the board to the number printed by the servo connector for that servo.

If you are confused about servo numbers and the board address, here is another way to think about it: Think of each servo number as a house number and a board address as the first house on a block. The houses are consecutively numbered. If you set a board's address to 10, the houses (servos) on that block (board) will be 10 through 17.

If you decide that you want to use this controller as a digital controller and the format is MiniSSC then the board will only take up one address. This address will be the board address. If you use it as a digital controller and you are using the BFX format then the board will take up two addresses. The reason for this is explained later.

Sample Address map for SSC format:

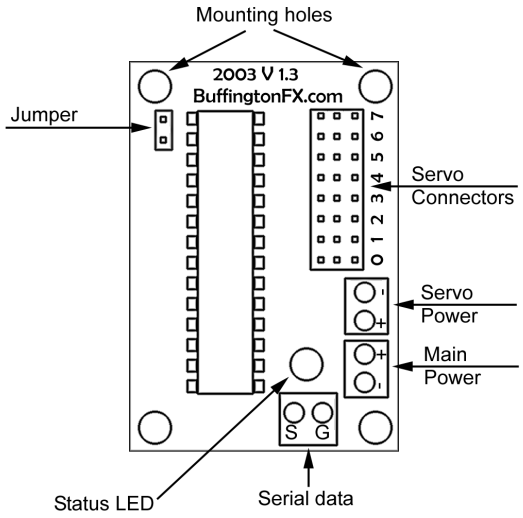


You are now ready to set the range of travel for your servos. There is a pie chart on the upper right of the upload utility. You can either manually move the edges of the white area, which represents the servo's travel, or you can move the two sweep sliders at the bottom of the screen. You can change which servo you are setting the travel for by using the track bar under the pie chart. The chart shows a representation of the travel of the servo but be aware that not all servos respond equally to the same signals. Your actual travel will most likely be a little different from what is shown. Your servo may not be able to respond to the full range of positions that you request of it. Near the ends of its travel your servo may come up against internal stops that prevent it from turning any farther. It is not recommended that you allow this to happen for very long. When a servo

tries to achieve a position that it is prevented from getting to it will draw a lot of current. There is also a small possibility that the servo will break or overheat.

If you place your controller into configuration mode you will be able to see the servos connected to your board move as you adjust the sliders or the travel graph. To place your servo controller into configuration mode, connect your servo controller to your computer, place a jumper over the two jumper pins on the left side of the board, then connect the power to your servo controller. The LED on the servo controller should go on and stay on. You will need to select the serial port that you are using in the upload utility before your servo controller will be able to show you what positions it is driving your servos to.

Once you have the travel of all of your servos as you would like them, press the "Upload Settings" button on the upload utility software to make these settings permanent. Your settings are now stored in the servo controller. They will remain there even if the power is disconnected. Now remove the configuration jumper. If you are using a standard baud rate you will now be able to test the ranges of your servos by using the slider in the green area at the bottom of the screen. If you don't like something about how the controller is set up, disconnect power from the servo controller and put the configuration jumper back on. Now apply power to the controller again. This will put it back into configuration mode. Change the settings that you have in the configuration software and upload them again.



Connections

The servo controller has three pairs of screw terminals. On the right side of the board you will find two of them. These are for the servo and logic power. For your servo power it is recommended that you do not go over about six volts. Higher voltages could damage your servos. For your logic or main power you can apply anything from 5 volts up to about 24 volts. It is recommended that you use a lower voltage such as power from a 9-volt battery. In some cases you may be able to get away with powering both the logic and main power from the same source. If the load on your servos is light and your power source can supply a lot of current then you should be OK. If you notice that under strain all of your servos start shaking then it is time to go to two power sources.

At the bottom of the board are the serial input screw terminals. The terminal on the left is the serial signal and the one on the right is serial ground.

On the upper left part of the board are the servo connectors. The pins nearest to the right edge of the board are the ground pins. On your servo's cable, this will usually correspond with a black wire but it may also be just the darkest wire.

Technical information

Current draw: 7mA @ 9Volts
Internal resolution: 7300 distinct positions
The BFX servo controller uses two data formats.

MiniSSC data format: To use this format you will need to send serial data in 8N1 format (8 bits, no parity, one stop bit). To command a servo to move you will need to send the following bytes:

- 255
- The servo number (0-255)
- What position you want it to go to (0-255)

That's it! It is a really easy format to use.

BFX data format: This format is more complex to use but allows you drive more servos in real time (30 updates per second) or more servos total (up to 65535). To send data in the BFX format, send the following in 8N1 format:

- 255
- 255
- Low byte of number of bytes to follow not including the 0 at the end
- High byte of number of bytes to follow not including the 0 at the end
- Data byte #1
- Data byte #2
- Data byte #3
- …….
- Data byte #N (the last data byte)
- 0 (zero)

There are some quirks to using the BFX format. The BFX controllers know when a packet of data has started by looking for two 255's in a row. If you have a sufficient number of data bytes there arises the possibility that two consecutive bytes could be 255. You should avoid sending 255's when controlling servos. Missing one position out of 256 possible positions is unlikely to be noticeable. If you absolutely cannot live without the position #255 then you can send it and the BFX controller will be able to drive your servo to that position but you should try to arrange your servos so that no two consecutive servos are sent a 255 at the same time. When it is in digital controller mode using the BFX format, it is likely that you will want to be able to use a 255. Because of this the digital controller mode takes up two bytes of data where the first byte is what is displayed on the output port and the second byte can be sent as anything other than 255.

The second and third bytes sent represent how many data bytes will follow. Internally this is a 16 bit number so it must be broken into two 8 bit bytes.