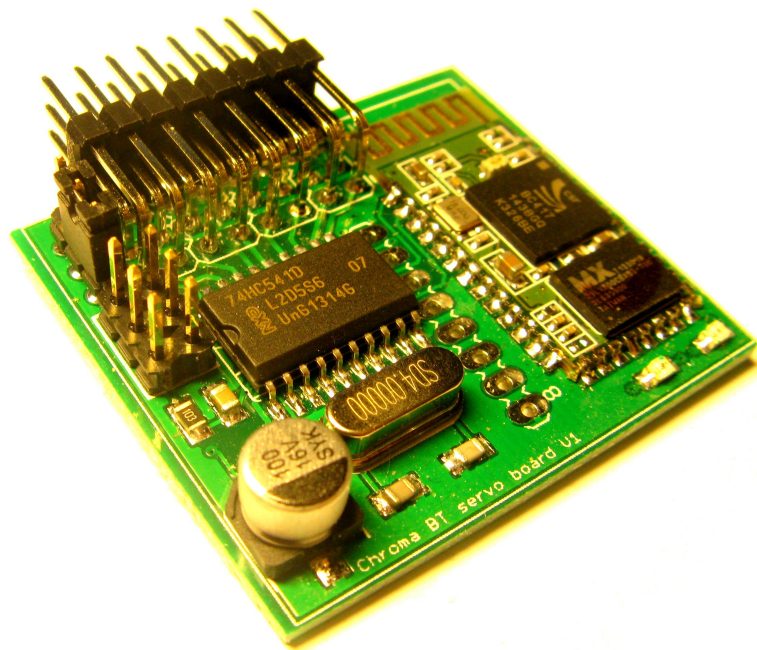


Chroma

Bluetooth Servo Board

(Firmware 0.1)

2015-02-08



Default Bluetooth name:
Default pin-code:

Chroma servo board
1234

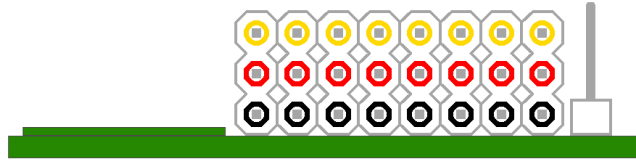
Content

Setup.....	3
Connecting servos.....	3
Power options.....	4
Getting started.....	6
Connecting to the board.....	6
Using shell.....	6
Using minicom.....	6
Using Python.....	7
The protocol.....	8
Overview.....	8
Servo Test.....	9
Servo All Velocity.....	9
Servo All.....	9
Servo N.....	10
Servo Initial position All.....	10
Servo Failsafe.....	11
Servo Bluetooth Pin.....	11
Servo Bluetooth Name.....	12
Servo versioN.....	12
Servo Enable.....	13
Servo Disable.....	13
(Servo) Output.....	13
Installing new firmware	15
Software changes.....	15
Specification.....	16
Size.....	16
Electrical	16
Pinout.....	17
Hints for hacking.....	17

Setup

Connecting servos

When connecting servos make sure ground wire of servo is closest to the edge of the servo board. See picture below (black – ground, red – 5V, orange – signal). Also make sure the servo connector is aligned.



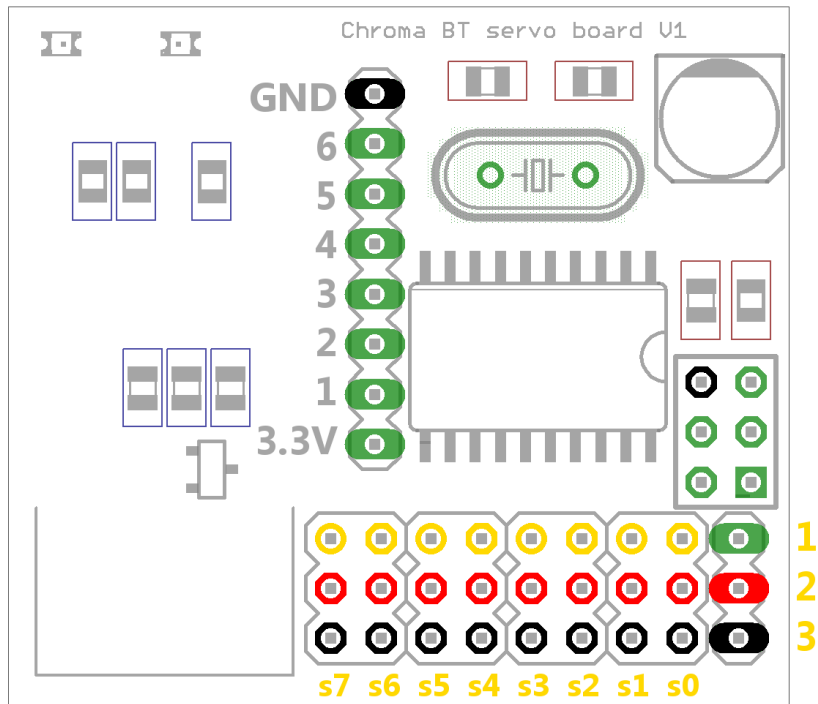
Power options

The board logic and servos can be powered separate or together.

WARNING:

Board logic voltage shall NOT exceed 5.5V. Make sure your ESC with built in BEC, BEC or battery is in the range 4.0V to 5.5V.

Power connector/jumper



The green-red-black 3-pin connector/jumper to the right of the servo-connection block on the above illustration is:

Pin 1. +5V for Board logic

Pin 2. +5V on servo rail

Pin 3. Ground

With no jumper installed servo rail +5V is disconnected from board +5V.
ESC with BEC or separate BEC is called BEC below.

1. Examples:

Power: BEC powers everything
Jumper: Between 1 and 2.
Note: One servo connector is used for ESC-BEC/BEC.

Power: BEC powers servos only. External 5V powers board logic
Jumper: No jumper!
Note: Connect external 5V to Pin 1 and external ground to Pin 3.

Power: External 5V powers everything
Jumper: See below
Note: Connect external 5V to both Pin 1 and Pin 2 (for example by soldering).
Connect external ground to Pin3.

Getting started

How to get started to actually control servos. See the protocol section to learn more about the different commands.

Connecting to the board

Establish a bluetooth serial port connection to the board. The board is visible when doing a bluetooth search and shows up as "Chroma servo board". The default pin-code is: 1234. The below examples is for Linux, replace /dev/ttyBT with your bluetooth serial port. Windows might create one incoming and one outgoing COM-port. Use the outgoing.

Since the board is connected via bluetooth, the bps-setting in your terminal program does not matter.

Using shell

Start servo test:

```
echo "st" > /dev/ttyBT
```

Stop servo test with any command or a wrong command:

```
echo "serr" > /dev/ttyBT
```

Using minicom

Install minicom (Internet connection required)

```
sudo apt-get install minicom
```

Start minicom in setup mode as root to configure it, only needed once:

```
sudo minicom -s
```

Select "Serial port setup"

Press "a" and change Serial device to /dev/ttyBT

Press enter

Press "e" and choose 9600 bps by pressing "c".

Press enter

Press enter to exit "serial port setup"

Select "Screen and keyboard"

Press "q" to enable local echo.

Press enter to exit "Screen and keyboard"

Select "Save setup as dfl" to save this as your default setting.

Select "Exit from Minicom"

From now on you can start minicom and issue commands to the servo board when you like:

```
minicom
```

Just try issuing the servo test command:

```
st
```

Using Python

To use the serial port from Python, you will need the serial module:

```
sudo apt-get install python-serial
```

Sample Python script:

```
import time
import serial

s = serial.Serial("/dev/ttyBT", 9600)

s.open()

s.write("st\n")      # Servo test command
time.sleep(5)       # Wait for 5 seconds
s.write("serr\n")   # Wrong command to stop servo test

s.close()
```

The protocol

It's a small and simple ASCII protocol. Start of command is "s" and end of command is enter (line feed or carriage return). Internal communication is at 38400bps.

Every command will either be answered with an "ACK" or a "NACK" if not understood. The only exception is the firmware version command "sn" that will answer with the string "01" for firmware version 0.1.

There are four commands that make persistent change (remembered after power loss):

- Servo Initial position All command: sia
- Servo Failsafe command: sf
- Servo Bluetooth Name command: sbn
- Servo Bluetooth Pin command: spn

Overview

Command	Name	Description
st	Servo Test	All servos move slowly between -100% and 100%
sa	Servo All	Change position of all 8 servos with one command
sav	Servo All Velocity	Change speed for all 8 servos with one command
s0 ... s7	Servo N	Set position (and speed) for servo N.
sia	Servo Initial position All	Set default/failsafe position for all 8 servos. Saved and used at power up and on loss of BT-connection if selected by sf command.
sf	Servo Failsafe	Turn on/off failsafe per servo. Default on for all.
sbn	Servo Bluetooth Name	Change the bluetooth name of the board. Board must be power cycled for this change to have effect.
sbp	Servo Bluetooth Pin	Change the bluetooth pin code. Must be a 4-number code. Board must be power cycled for this change to have effect.
sn	Servo versioN	Returns firmware version "NN". Example "02" for version 0.2.
se	Servo Enable	Enable servo output (default)
sd	Servo Disable	Disables servo output (tristates output buffer IC)
so	(Servo) Output	Experimental: Controls 8 hole connectors center 6 pins.

Servo Test

Has no parameters and will move all servos in sync between -100% and 100% at about 0.2% per millisecond. Any command or a wrong command will abort.

Servo All Velocity

Set speed at which servos should move when using "sa"-command.

Takes 0 to 8 parameters. Any parameter omitted is interpreted as 0. Parameters is servo speed in steps of 10% per second from 1 to 255 or 0 for fastest possible speed. First parameter is speed for first servo, second parameter is speed for second servo and so on.

Examples:

```
sav 1 10 20 20
```

Sets first servo to move at slowest possible speed: 10% per second, second servo at 100% per second, third and fourth at 200% per second. Servo five to eight will move as fast as possible.

```
sav
```

Sets all servos to move as fast as possible

```
sav 100
```

First servo is set to move at 1000% per second. Remaining seven is set to move as fast as possible.

Servo All

Set position for servos

Takes 0 to 8 parameters. Any parameter omitted is interpreted as 0. Parameters is servo position in promille from -1000 to 1000 (-2500 to 1900 is allowed for going beyond defined range). First parameter is position for first servo, second parameter is position for second servo and so on.

Examples:

```
sa 0 1000 500 -1000
```

Moves first servo to 0%, second to 100%, third to 50% and fourth to -100% the remaining 4 will be moved to 0%.

sa

All eight servos are moved to 0%

sa -1500

First servo is moved to -150% all other servos are moved to 0%.

Servo N

Move one servo to a given position at a given speed.

Takes 0 to 2 parameters. Any parameter omitted is interpreted as 0.

First parameter is servo position in promille from -1000 to 1000 (-2500 to 1900 is allowed for going beyond defined range)

Second parameters is servo speed in steps of 10% per second from 1 to 255. 0 for fastest possible.

Examples:

s0 1000 0

Will move the first servo to 100% Only physical speed of servo limits speed.

s0 1000

Will move the first servo to 100% Only physical speed of servo limits speed. The same as above

s1 500

Will move the second servo to 50% Only physical speed of servo limits speed.

s6 -1000 5

Will move the seventh servo to -100% at roughly 50% / second.

s2 1300 1

Will move the third servo to 130% at roughly 10% / second.

If the second parameter is given and is different from 0, the given speed will be used in subsequent "sa"-commands overriding any previous "sav" commands.

Servo Initial position All

Sets default position for servos, always used at powerup and at Bluetooth disconnect if enabled by sf command.

Takes 0 to 8 parameters. Any parameter omitted is interpreted as 0.

Parameters is servo position in promille from -1000 to 1000 (-2500 to 1900 is allowed for going beyond defined range). First parameter is position for first servo, second parameter is position for second servo and so on. This command will not move any servo, just save the values to be used on power up.

Examples:

```
sia 0 1000 500 -1000 -780 -50 -5 -2000
```

At every power up first servo is set to 0%, second to 100%, third to 50%, fourth to -100%, fifth to -78%, sixth to -5%, seventh to 0.5% and eight to -200%.

Servo Failsafe

Enables or disables the failsafe functionality per servo.

Takes 0 to 8 parameters. Any parameter omitted is interpreted as 0.

Parameters interpreted as such: 0 = disabled, anything else = enabled.

If failsafe is enabled for a servo and bluetooth disconnects, that servo is moved to the position given by the sia-command as fast as possible. If failsafe is disabled, the last commanded position will be held.

Examples

```
sf 1 0 0 0 0 1
```

Will enable failsafe for servo 0 and 5, the rest is disabled.

```
sf
```

Will disable failsafe for all servos.

```
sf 22 97 0 373 23 1
```

Will enable failsafe for servo 0,1,3,4 and 5, while disabled for servo 2,6 and 7.

Servo Bluetooth Pin

Sets the bluetooth pairing pin

Takes one parameter. Has to be 4 digit number or will be NACK:ed.

Change will have no effect until after power cycling the board.

Examples

```
sbp 4242
```

Will set the bluetooth pin code to 4242. (Effective after power cycling the board)

```
sbp fortytwo
```

Will result in unchanged pin code and NACK.

Servo Bluetooth Name

Sets the boards bluetooth name.

Takes one parameter, the name.

If the name contains space you can enclose the name in "".

Examples

```
sbn MyCar
```

Will set the bluetooth name to: MyCar

```
sbn "My Car"
```

Will set the bluetooth name to: My Car

```
sbn My Car
```

Will set the bluetooth name to: My

Servo version

Returns firmware version.

Takes no parameters.

Example:

```
sn
```

Returns firmware version "NN". Example "01" for version 0.1. Only command that will not return ACK/NACK. (" and " not included in return string)

Servo Enable

Enables servo pulses output (default)

Takes no parameters

Example:

```
se
```

Servo Disable

Disables servo pulses output. (by tristating buffer IC output)

Takes no parameters

Example:

```
sd
```

No output pulses will be sent to servos, commands will be accepted as usual, and any changes to servo positions will be seen upon giving the "se"-command.

This makes it possible to save power by "disabling" servos.

(Servo) Output

Experimental – might be removed/replaced in future firmware.

Controls state of the center 6 pins of the 8 hole connector in the middle of the board. Pin 0 of the connector is 3.3V from internal regulator. Pin 1..6 is controlable output signals, either 0V or 3.3V max current draw is a 10mA per pin. Pin 7 is GND.

Pin 0 is the one closest to servo connectors.

Takes 0 to 6 parameters. Any parameter omitted is interpreted as 0. First parameter is state of pin 1 of 8 hole connector., second is pin 2 and so on. Parameters interpreted as such: 0 = 0V, anything else = 3.3V.

Example:

```
so 1 0 0 0 0 1
```

Will set pin 1 and 6 of 8 hole connector to 3.3V, the rest to 0V.

```
so 1 0 1
```

Will turn pin 1 and 3 of 8 hole connector to 3.3V, the rest to 0V.

so

Will turn all 6 signal pins of 8 hole connector to 0V.

so 1 1 1 1 1 1

Will turn all 6 signal pins of 8 hole connector to 3.3V.

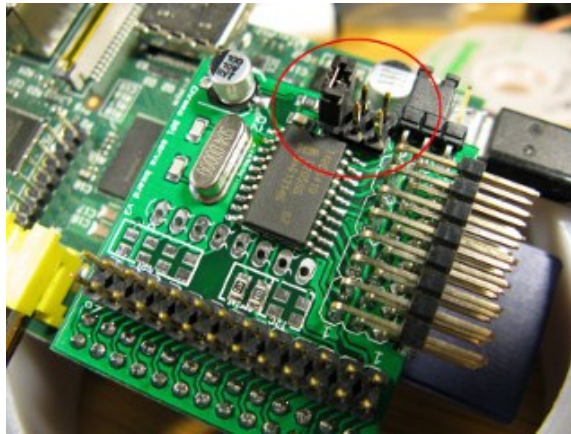
so 12 77 0 123 33 1

Will turn all but pin 3 of 8 hole connector to 3.3V.

Installing new firmware

The board has optiboot installed which allows for installation of new firmware without tools. Check <http://electronics.chroma.se/> for latest firmware.

1. Disconnect all servos/ESC's
2. Download the new firmware to your computer (or compile your own).
Let's assume it is named: main.hex
3. On your Linux computer: `sudo apt-get install avrdude`
On Windows download WinAVR.
4. Put a jumper on the servo boards ISP-connector shorting pin 5 and 6. (You can borrow the power jumper that came with the board). See picture
(Picture below does not show the Chroma Bluetooth servo board, but the Raspberry Pi board. It is similar enough to understand the procedure).



5. Now type the following as one line on your computer. Replace com3 with your bluetooth serial port (outgoing):

```
avrdude -c arduino -p atmega8 -P com3  
-U flash:w:main.hex
```

6. Hit enter and remove the jumper when Bluetooth LED lights solid.

If it fails, repeat steps 4 to 6.

Software changes

Version 0.1

- First version.

Specification

Size

PCB Size: 41 mm x 36 mm
Total Size: 41 mm x 42.5 mm
Total height: 11.5 mm

Electrical

(No servos connected.)

5V current consumption: ~45 mA unpaired
 ~43 mA paired
 ~25 mA paired sending data
 ~8 mA paired idle

Servo rail maximum continuous current: 4 A

When no more data is being sent, paired idle mode is entered after ~5 seconds
(Bluetooth host dependant?)

8-hole GPIO connector

Max current per output pin, pin 1 to 6: 10 mA
Max current draw from pin (hole) marked 3.3V: 100 mA

Pinout

The board has an internal 3.3V regulator that powers the bluetooth module and MCU. The 74HC541 buffer IC is powered directly by 5V

- Pin 1 in 8-hole connector connected to ATmega8 SDA line (pin 27 of MCU)
- Pin 2 in 8-hole connector connected to ATmega8 SCL line (pin 28 of MCU)
- Pin 3 in 8-hole connector connected to ATmega8 SS pin (pin 14 of MCU)
- Pin 4 in 8-hole connector connected to ATmega8 MOSI pin (pin 15 of MCU)
- Pin 5 in 8-hole connector connected to ATmega8 MISO pin (pin 16 of MCU)
- Pin 6 in 8-hole connector connected to ATmega8 SCK pin (pin 17 of MCU)

For 3-pin jumper/power connector and 3x8 pin servo connections see section Power Options above.

2x3 pin male header is standard AVR ISP-connector. Pin 1 is the one closest to the board edge and the 3-pin jumper/power connector. (Pin 3 in AVR ISP-connector and Pin 6 in 8-hole connector is also optiboot boot-LED)

Hints for hacking

The on-board bluetooth module is a HC-05.