

PiKoder/SSC DC User's Guide

Version 1.0
dated 06/25/17

Gregor Schlechtriem
webmaster@pikoder.com

www.pikoder.com

Content

Overview	3
Programming your PiKoder/SSC for daisy chaining	5
Testing your parameters	7
Setting up the complete system	9
Wiring diagrams	11
PiKoder/SSC DC schematic and pinning	13
miniSSC protocol description	15

1

Overview

The PiKoder/SSC DC is a base board greatly simplifying daisy chaining for up to eight PiKoders thus catering for up to 64 servo channels. It provides you with a bus driver, some logic to avoid shorts caused by communication problems, a dc-dc power converter and a holder for a radio – be it a Bluetooth module or a WLAN radio. This guide provides the respective settings, the wiring diagrams and the schematics and pinning of the board.

Daisy chaining of PiKoder/SSC's is based on the miniSSC protocol which supports up to 255 servo channels. Thereby it lends itself as a very suitable protocol for multi-channel RC applications. Each PiKoder/SSC can control up to eight servos. If you want to address more than eight channels you would thus daisy chain a respective number of PiKoder/SSCs (be it PiKoder/SSCs, PiKoder/SSC wRX, or PiKoder/SSC RX) to support the number of channels required by your application.

For correct addressing, you would assign a different miniSSC offset to each device connected to the bus, so that the PiKoders can be individually addressed by your serial commands. This will be done using the PiKoder Control Center PCC application on your PC.

For testing your configuration, you could use the Pololu Serial Transmitter application which allows you to emulate the miniSSC protocol. Please note that the PiKoder Control Center always assumes a one on one configuration and therefore does ignore any miniSSC offset.

Please note that besides the hardware setup and the programming for daisy chaining you would additionally have to make sure in your control application that no serial command you send to the controllers will cause unintended operations of the devices it was not address to. Therefore, whenever you would connect more than one PiKoder to the bus, then you would have to strictly adhere to the miniSSC command structure.

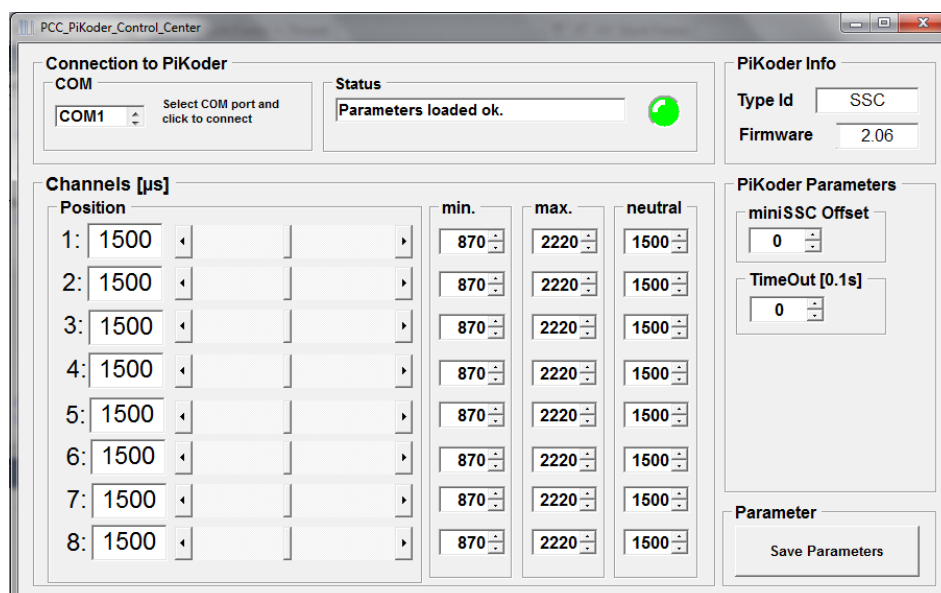
The next chapters will guide you through the steps involved to daisy chain your PiKoder/SSCs using the PiKoder/SSC DC. Please always check for updated information and new software releases on www.pikoder.com before you start.

2

Programming your PiKoder/SSC for daisy chaining

The PiKoder Control Center PCC provides access to all configuration options as well as for real-time control with an easy to use graphical user interface. For almost any project, you will want to start by using the control center to set up and test your PiKoder/SSC. This section explains how to assign the miniSSC offset to a PiKoder/SSC. Please refer to the User's Guide of the respective PiKoder/SSC in regards to setting up the connection between your PC and the controller.

After starting the PCC Control Center and connecting to the PiKoder/SSC the application shows information similar to the screenshot below.



PCC (PiKoder Control Center)

The miniSSC Offset is programmed in the respective field. The value in this field determines which servo numbers the device will respond to in the miniSSC protocol. The default value is '0' meaning that this PiKoder/SSC

will control channels 0 to 7 according the miniSSC protocol definition. If you wanted to address the next eight channels beginning with number 8 then you change the field value to 8. For more details on the miniSSC protocol please refer to Exhibit C.

The PiKoder/SSC will initially store the update value in its memory and this information would be lost after loosing power. In order to retain this value you would have to store it in the controller's EEPROM by clicking the 'Save Parameters' button in the lower left corner.

Please bear in mind that all controllers require a different offset to allow for individual addressing. This means that you have to program each controller following the steps outlined here.

If you want to set any other parameter to an application specific value, e.g. upper or lower limits, then this would be the right point in time to do so.

3

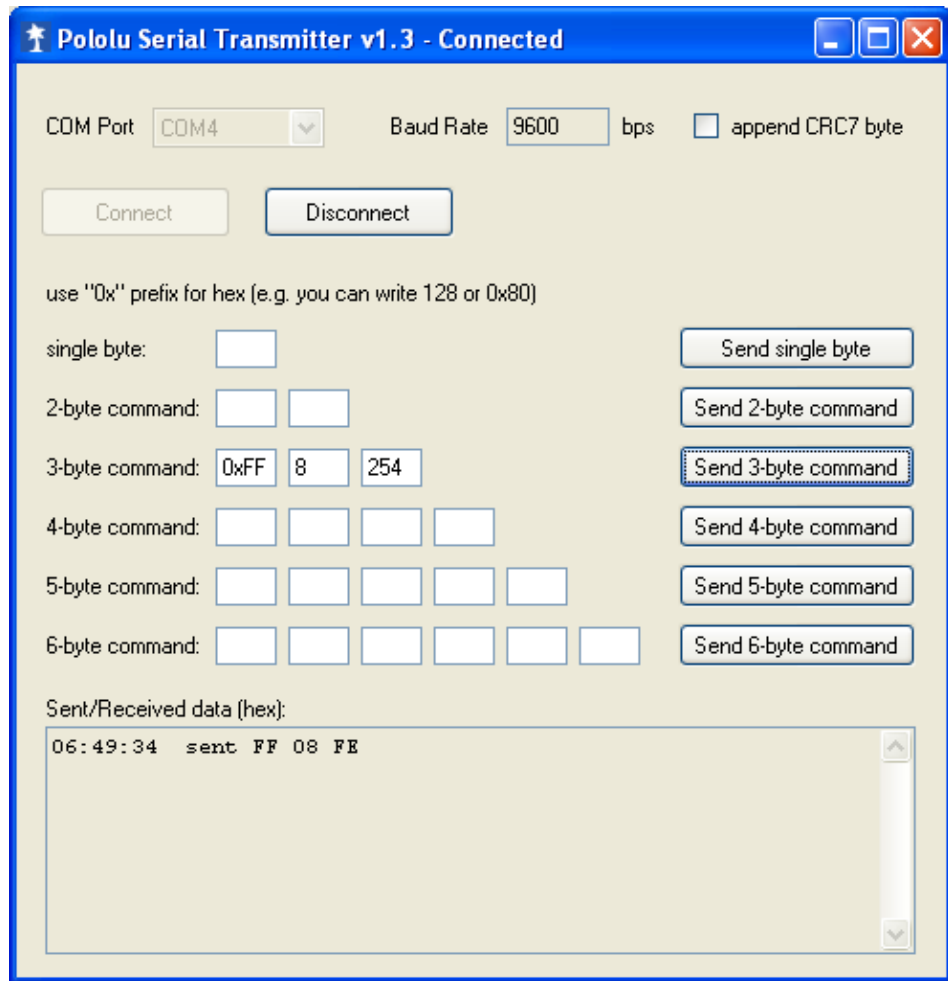
Testing your parameters

After completing the programming it is recommended that you test your programming outside of the application using the Pololu Serial Transmitter (<http://www.pololu.com/docs/0J23>).

Download the software and follow the recommended installation procedure. Then start the program and connect to the PiKoder/SSC using the same COM port you used before with the PCC. Please make sure that you did end the SSC Comand Center application in order to release the COM port.

For the following test it is assumed that you had programmed your PiKoder/SSC's offset value to eight. In this case the PiKoder/SSC would control channels 8 through 15. The following test procedure will verify the correct programming.

Connect a servo to the PiKoder/SSC's Channel 1 (the connector closest to the servo power connector) and make sure that the PiKoder as well as the servo is powered up. Then enter a miniSSC command to move servo eight to the right as shown in the screenshot on the next page and send the commando to the PiKoder/SSC by clicking the Send-3-byte-command button. The servo should move now to the right.



4

Setting up the complete system

After completing and testing the programming you would now wire the components and commission the system.

Exhibit A shows the pinning of the respective components. Please take a look at the respective pinning of your PiKoders and connect the respective pins. The PiKoder/SSC DC provides for rows of connected pins to simplify wiring the system.

It is recommended that you perform another test as described in the previous section to check that all channels are working as anticipated.

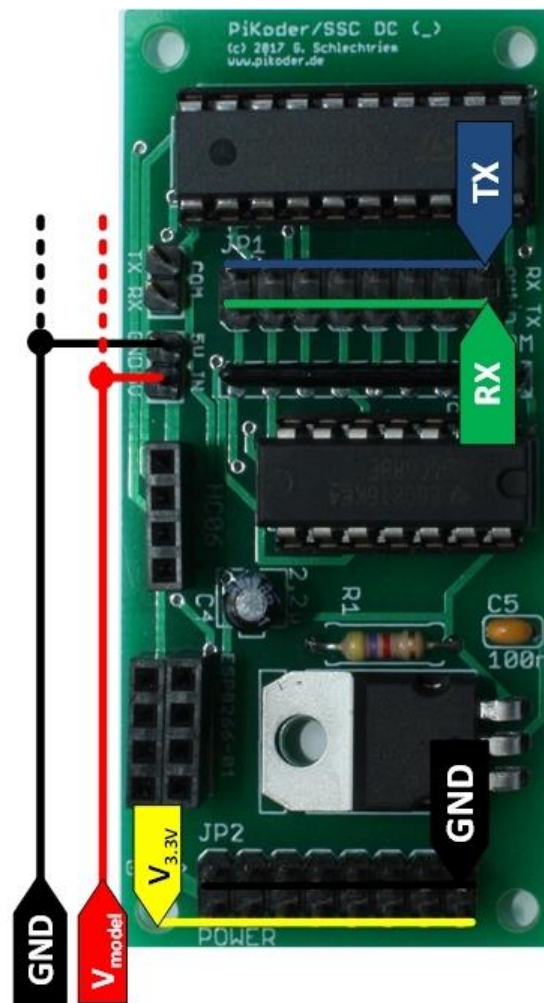
The final step would be to insert the radio (either a HC06 for Bluetooth communication or an ESP8266-01 for WLAN) into the respective position.

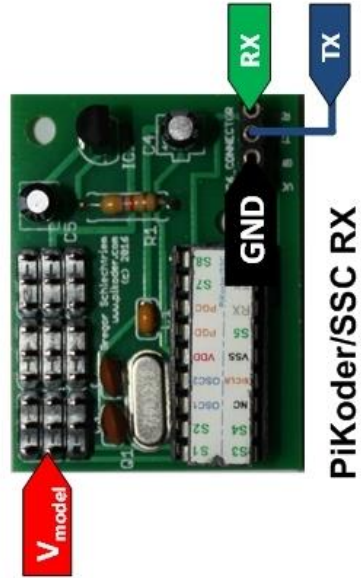
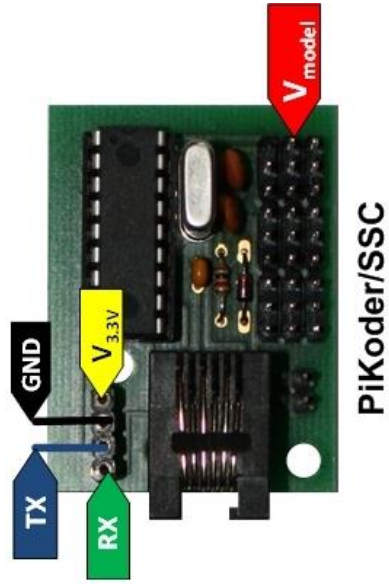
The following image shows the PiKoder/SSC DC with a plugged in ESP8266 module.



A

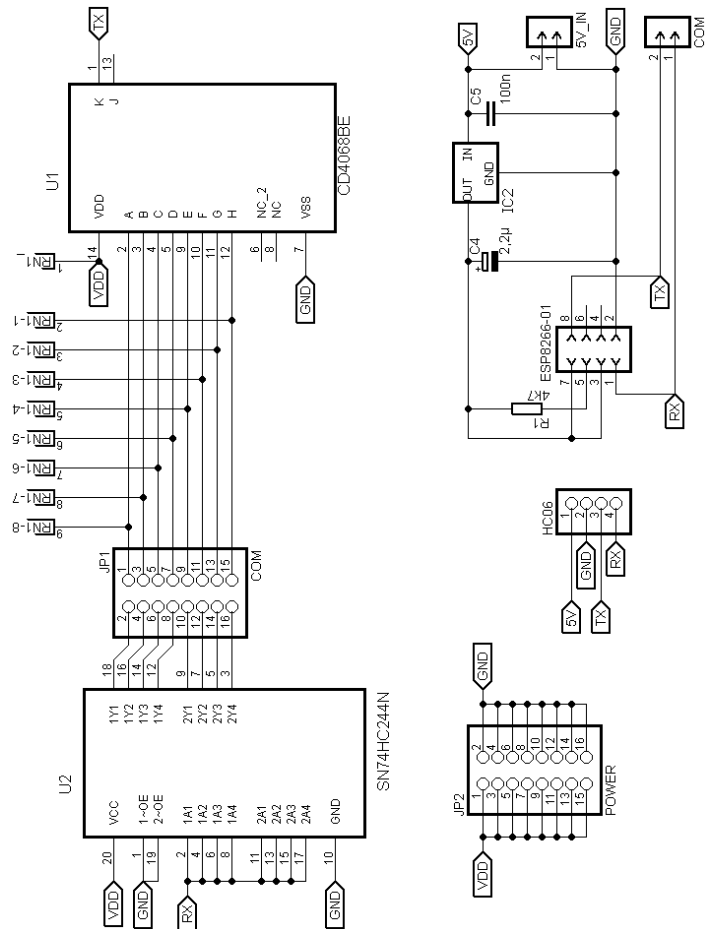
Wiring diagrams

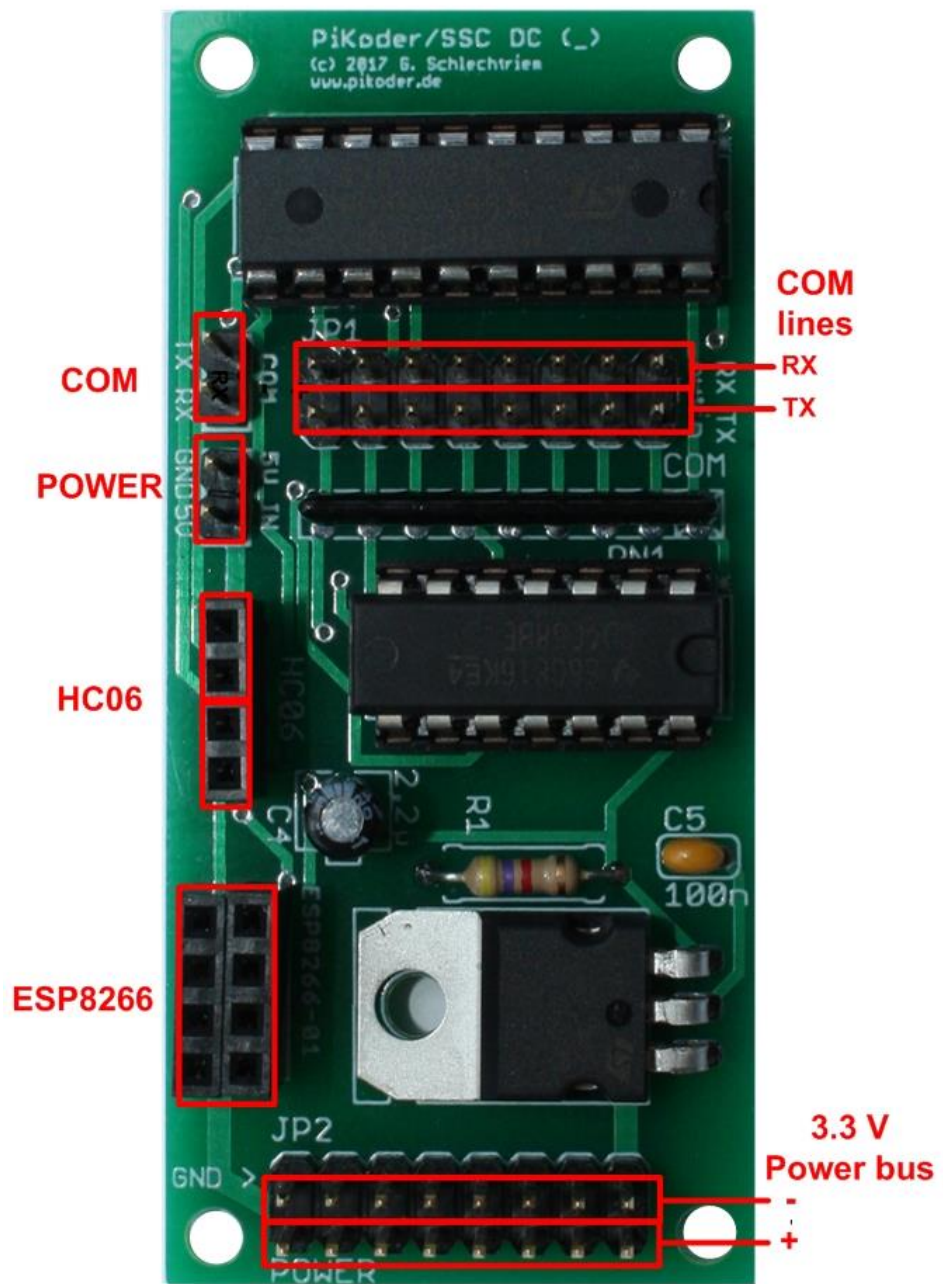




B

PiKoder/SSC DC schematic and pinning





C

miniSSC protocol description

The miniSSC protocol allows you to control up to 254 different servos by chaining multiple servo controllers together. It only takes three serial bytes to set the target of one servo, so this protocol is good if you need to send many commands rapidly. The miniSSC protocol is to transmit 0xFF (255 in decimal) as the first (command) byte, followed by a servo number byte, and then the 8-bit servo target byte. Therefore, the command packet is:

0xFF, servo number byte, servo target byte

For example, if we wanted to set the target of servo 0 to its neutral position, we could send the following byte sequence:

in hex: 0xFF, 0x00, 0x7F

in decimal: 255,0,127

The PiKoder/SSC can be configured to respond to any contiguous block of miniSSC servo numbers from 0 to 254.