

---

# *PiKoder/SSC*

## Daisy Chaining

---

Version 1.2  
dated 05/20/2017

Gregor Schlechtriem

## **Content**

<b>Overview</b>	<b>3</b>
<b>Assign miniSSC offset to the PiKoder/SSC</b>	<b>5</b>
<b>Testing your configuration</b>	<b>7</b>
<b>miniSSC protocol description</b>	<b>9</b>

-

---

# 1

---

## **Overview**

The miniSSCII protocol does support up to 255 servo channels. The PiKoder/SSC can control up to eight servos. If you want to address more than eight channels you would have to daisy chain a respective number of PiKoder/SSCs to support the number of channels required by your application. This application note would guide you to integrating the PiKoder/SSC into a project that has multiple TTL serial devices that use the miniSSCII protocol.

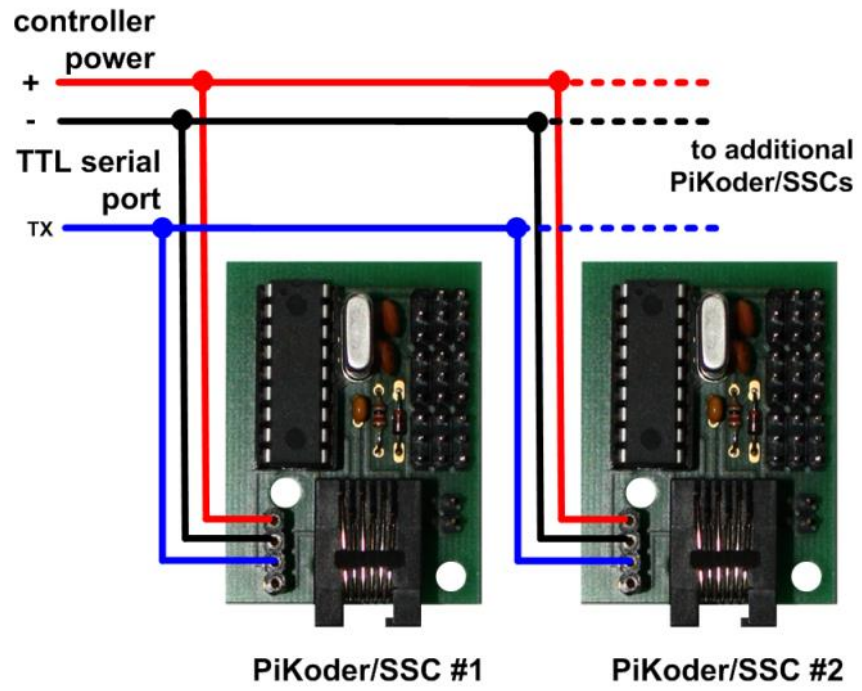
First of all, you have to make sure that no serial command you send to the daisy chained controllers will cause unintended operations of the devices it was not address to.

Secondly, assign each device in the project a different miniSSC offset, so that they can be individually addressed by your serial commands. For the PiKoder/SSC this will be done using the SSC Control Center application on your PC.

From a hardware standpoint, you will need the respective number of PiKoder/SSCs and bus cables for power and the receive line. The diagram on the next page shows how to connect the controllers into the chain.

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

The next chapters will guide you through the steps involved to daisy chain your PiKoder/SSC. Please always check for updated information and new software releases on [www.pikoder.com](http://www.pikoder.com) before you start.



PiKoder/SSC hardware setup for daisy chaining

---

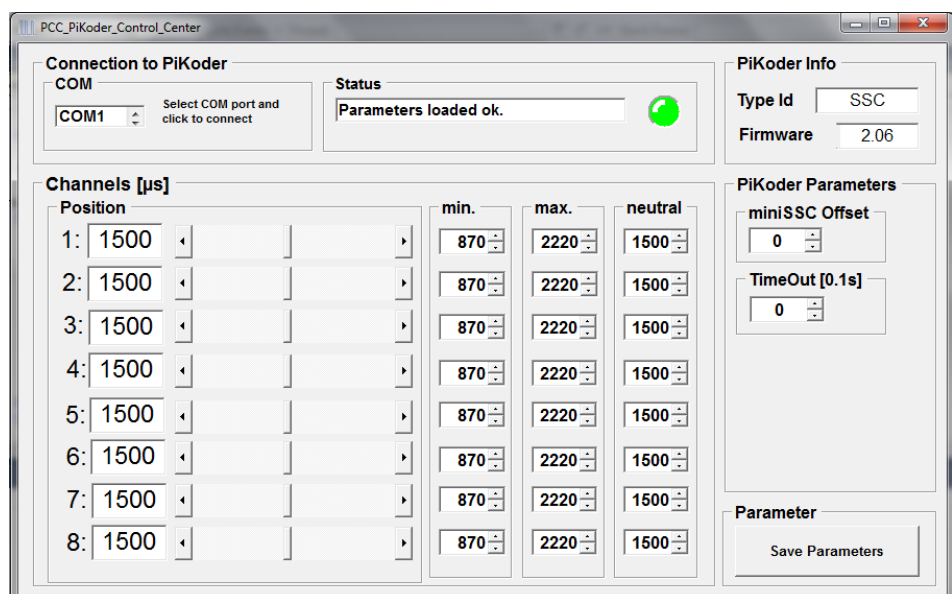
# 2

---

## *Assign miniSSC offset to the PiKoder/SSC*

The PCC Control Center 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.

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



**PiKoder/SSC 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.

The PiKoder/SSC will initially store the update value in its memory and this information would be lost after losing 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.

---

# 3

---

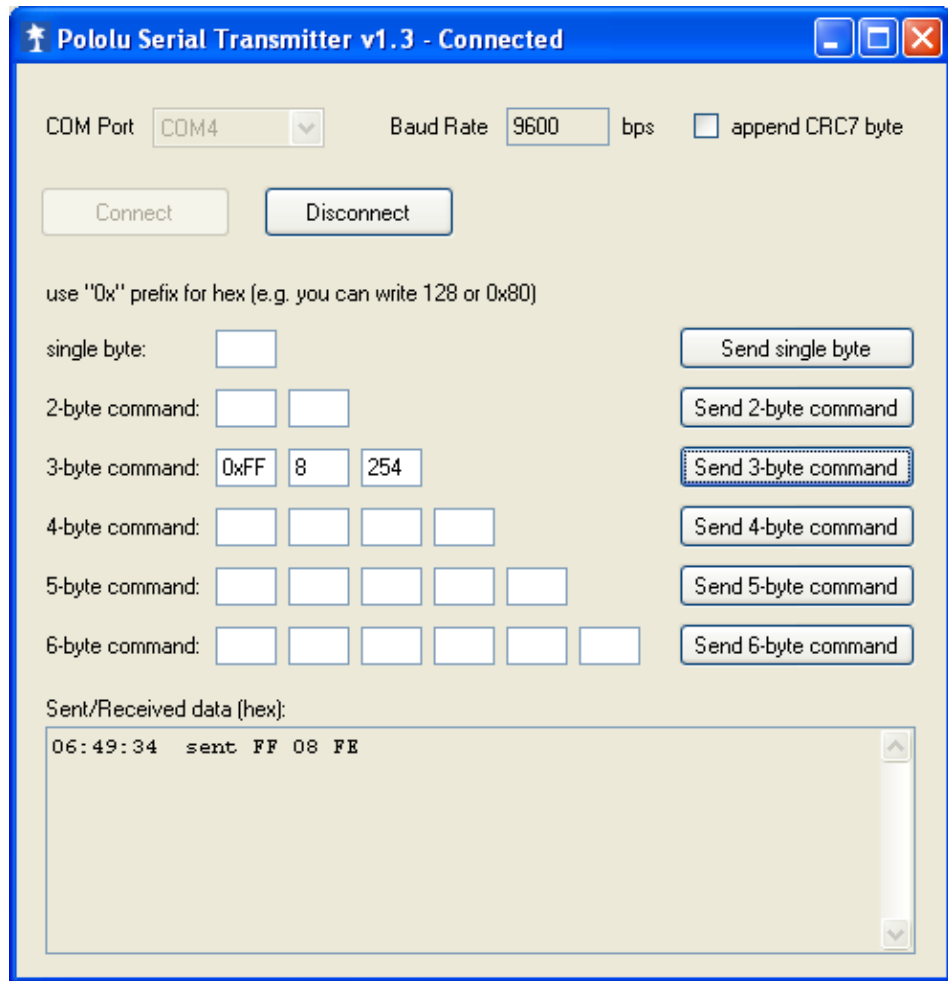
## ***Testing your configuration***

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 SSC Comand Center. 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 command to the PiKoder/SSC by clicking the Send-3-byte-command button. The servo should move now to the right.





---

# A

---

## *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.