NORTHERN LIGHTS ELECTRONIC DESIGN

www.NLEDShop.com/manuals www.NorthernLightsElectronicDesign.com

NLED Aurora Serial Control Specifications

(Not Dual Command Mode or Serial Communication)

Serial reception can directly control the device's outputs using a common serial transmission device. Such as a standard COM port with level translation, an FTDI adapter, Arduino, PIC UART, wireless XBee, or similar. The protocol is extremely simple and can be easily implemented on with any language and most hardware platforms with TTL serial out.

The device receives data in packets, the length of the packet corresponds to how many channels will be controlled. Example: Sending a packet of 90 bytes will control 30 RGB pixels, sending 510 bytes would control 170 RGB pixels. Sending 120 channels would control 30 RGBW pixels.

Each device has a specific way of entering it into serial reception. The configuration may be accessed via NLED Aurora commands, through an external LED display, or onboard DIP switches. See the device's datasheet for more information on setting it up. Look under the Serial Reception section, the Configuration section, and the pinout/hookup sections for details.

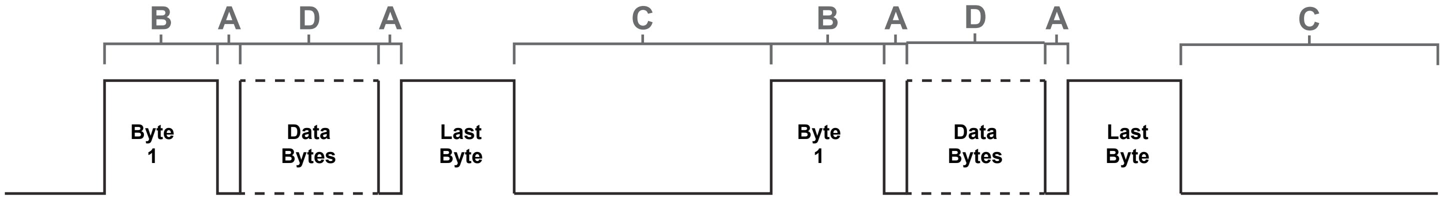
A device may have RS485 hardware that can be utilized for Serial Reception. RS-485 is a differential transmission method, that DMX-512 utilizes. The user can select through the software or hardware settings to utilize the RS-485 hardware for Serial Reception(transmitting device must be RS-485 as well) or to disable usage and utilize the RX-TX-GND header for standard TTL serial control.

Each device has a latch period(end-of-frame), that indicates end of packet. The latch period is commonly 5mS, but could be 1mS, see device datasheet for info. A delay between received data bytes of more than the latch period will reset the buffer pointer, framing the data. Partial packets will latch after the latch delay, or latching will occur immediately after receiving the last byte(if configured on certain devices).

Tested and known to be compatible with the "Glediator Protocol"

Byte Format:	8-N-1(most common)
Byte Type:	unsigned char(8-bit, 0 to 255)
Levels:	TTL(5v or 3.3v devices)

*5 volt controllers can receive from 3.3v devices without level translation.



A:	**Less than Latch Period**	**Space between bytes**
B:	**See Formula**	**Data byte length, with start and stop bits**
C:	**Equal or Greater than Latch Period**	**Space between packets**
D:	**Any amount of bytes up to device maximum, see datasheet**	

Datasheet Revision: 1

Byte Rate: 10 bits are received for each received byte. 1 stop bit, 8 data bits, 1 stop bit
Baud Rate / 10 = Maximum Bytes Per Second Ex: 250,000 / 10 = 25,000 bytes per second max
Byte Time: 1000mS / Maximum Bytes Per Second = Byte Time In Miliseconds
Ex: 1000 / 25,000 = 0.04mS = 4uS per Byte

"Actual Baud Rate" refers to the baud rate the NLED controller is expecting. The "Closest Common Baud Rate" refers to the standard baud rates available in older serial devices. Always set try to set your transmitting serial device to match the "Actual Device Baud" if possible, that will ensure the best possible signal quality. If a common/standard baud rate has to be used, the Error% indicates the potential error that 1few percent error has no effect on signal quality.

ID#	Actual Device Baud	Closest Common Baud Rate	Error %
0	9,615	9,600	0.16%
1	19,230	19,200	0.16%
2	38,461	38,400	0.16%
3	57,142	57,600	0.64%
4	117,647	115,200	-0.79%
5	222,222	230,400	2.12%*
6	250,000		
7	444,444	460,800	-3.55%*
8	500,000		
9	1,000,000	921,600	8.51%

Fig. 2a **8-N-1**

*Note: Percentage of error is within usable limits. And won't affect usage.

"Actual Device Baud" is the exact rate the controller is expecting, match it for a 0% error value.

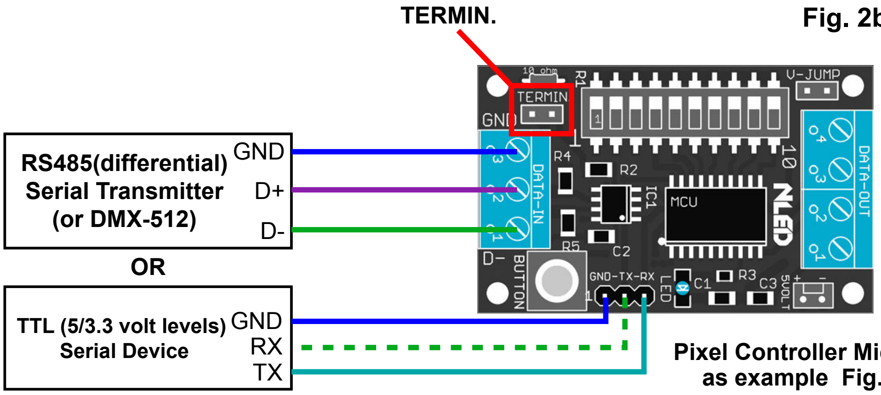
RX-TX-GND Header Connection

Device:

1 RX ← TX
2 TX → RX
3 GND ↔ GND

Fig. 2b

TERMIN. should only be installed if using RS-485 and the controller is the only device on the network. Not all controllers have it.



Pixel Controller Micro as example Fig. 2c

(RX to controller TX is not required)

Setup Instructions and Troubleshooting

1. Read the device datasheet to find any configuration settings or hardware settings(DIP switches) that is required to enter the device into serial reception mode.
2. Select the proper baud rate either through configurations or hardware settings.
3. Connect your NLED controller to your serial transmitter(FTDI, Arduino or otherwise) using the RX-TX-GND / Serial header. Remembering to connect: RX->TX , TX->RX, GND->GND.
4. If applicable, open any software and configure it by selecting the COM/Serial port, selecting the same baud rate as was selected for the controller, and setting up the software.
5. Power up the controller and transmitter/adapter.
6. If possbile, check that the NLED controller is indicating it is in serial mode.
7. Start the software or application code and begin serial transmission.
8. Your LEDs/Pixels should now be responding to the received data.

Troubleshoot:

LEDs strobing, blinking, or otherwise erratic OR notification LED blinking randomly:
Check pixel chipset configuration. Check that the baud rates match. Check that the LEDs, controller, and transmitter have enough power.