

NLED Pixel Controller Mini

The NLED Pixel Controller Mini supports several of the most popular pixel types and has the ability to accept standard DMX-512, in addition to TTL serial signals from a computer, microcontroller, Arduino, XBee or other TTL serial devices. Several configuration options, such as pixel IC type, amount of pixels to control, reception baud rate, and end-of-frame spacing can be customized to work well with your project. All configurations are changed over the serial connection header or through the DMX hardware(RS-485) using a compatible serial adapter or DMX transmitter. This small(1.5" x 1.5" x 0.5") design is great for small amounts of pixels for DMX control, strands, rope style lights, or even low-pixel LED matrices. Serial control is compatible with NLED Matrix, the Java based open-source LED Matrix control and mixing software.

Usage Modes:

DMX-512 Control: Configure the device to control from a single pixel(3 channels) up to 170 pixels(510 channels) over standard DMX-512. Full frames can be recieved at 40 FPS, partials can be faster.

Serial Control: The DMX(RS-485) interface can be used for serial control of the device at any baud rate and data length, but requires a compatible serial adapter or DMX transmitter that is based on a serial(COM Port) design. Or a standard TTL serial signal, from microcontroller/Arduino can be connected to the RX header and the SN75176 is removed from its socket, then set the desired baud rate using the configuration software. Supports baud rates from 9600 to 1,000,000(1 megabaud). Data packets are framed using an end-of-frame timer, data is framed when there is a 255uS to 32mS(user selectable), space between recieved bytes. Once space is detected it will handle the end of frame functions, any latching will occur, and the firmware will begin waiting for the next packet. Default baud rate is 500,000 actual.

Test Mode: Runs through a 7 color+Off(8 states) test pattern. All LEDs change through the 8 states together. Pushes up to 768 pixels. Good for testing a string or looking for dead pixels.

Command Mode: Setting both DIP Switches and power up the controller will enter the device into command mode. The LED should be constant on when starting in Command Mode. Send the configuration bytes at 9600 baud(9600/8-N-1), regardless of user baud rate setting. LED will start blinking when all configurations bytes have been collected. See open source Processing code for details.

DIP Switches		
DIP Switch	1	2
DMX Control	0	0
Serial Control	1	0
Test	0	1
Command	1	1

Supported Pixel I.C. Types:

Name	DMX Pixels	Serial Pixels	Color Bits	Output Data Speed	PWM Speed
WS2811	57	158	8-bit	800Khz	400Hz
WS2812	57	158	8-bit	800Khz	400Hz
WS2812B	57	158	8-bit	800Khz	400Hz
WS2801	170	1024+*	8-bit	2.66MHz SPI	2.5kHz
WS2803	170	1024+*	8-bit	2.66MHz SPI	2.5kHz
LPD8806	170	1280+*	7-bit	8MHz SPI	4kHz
LPD6803	170	1024+*	5-bit	10MHZ SPI	N/A

*More hasn't been tested but would be supported at lower frame rates

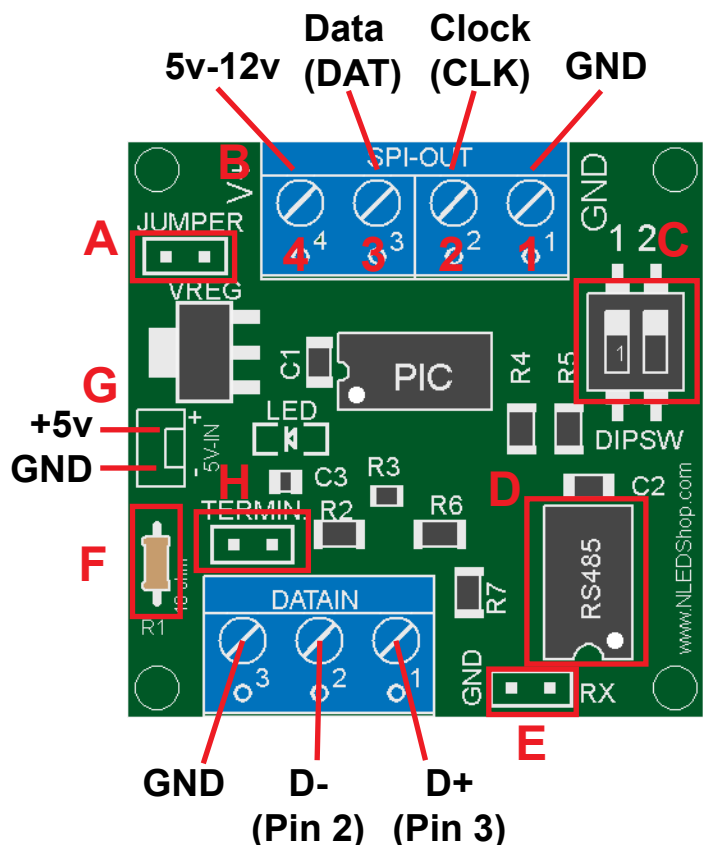
Device Specification:

Input Voltage	3.3v*, 5v**, 7v - 25v***
Current Draw	< 50mA
Input Baud	User Selectable
Connectors	Screw Terminals
Dimension	1.5" x 1.5" x 0.5"

*VREG Jumper Installed, DMX reception is disabled.

**VREG Jumper Installed

***VREG Jumper Left Open



- A:** Jumper, if using 5v input(common usage) jumper must be installed. Connect both holes together.
- B:** LED Pixel Connections, will use some or all of the positions, depending on pixel type.
- C:** DIP Switches for control of options, see table.
- D:** SN75176 RS-485 Transceiver, if using TTL serial input, it should be removed from socket.
- E:** TTL Serial Input Header, RX(RX->TX) and GND
- F:** 10 ohm resistor, acts as fuse in case of bad ground connection or failure. Easy to replace. 1/6w rated.
- G:** 5 Volt Input, optional connection for powering controller separately. Do NOT apply more than 5 volts.
- H:** DMX Termination Header, install jumper if the controller is the only or last device on the DMX chain.

LED Pixel Connections(B)

- 1: Ground
- 2: Clock
- 3: Data
- 4: Voltage Source, 5v - 12v

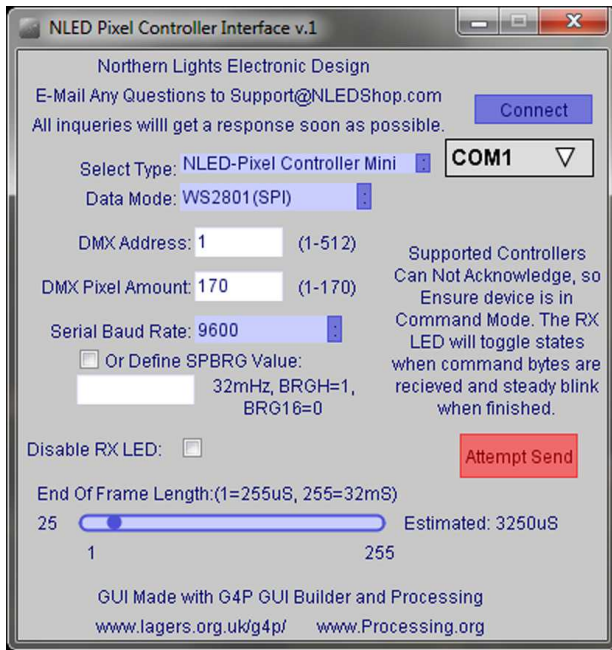
Most Pinouts Displayed on Bottom Side of PCB

Configuration Software Usage: (See Next Page for More Details)

- With the device powered off, change the DIP switches to Command Mode(Both Set)
 - Option 1: For direct TTL serial
 - Connect a Serial port(or TTL USB Serial such as a FTDI) to GND and RX(Connects to the TX pin of the serial port) headers on the controller
 - Carefully remove the SN75176 IC from the socket, RX Header won't function with it installed. Set it aside some where safe, ESD safe foam is recommended.
 - Option 2: Uses serial over the RS-485 transceiver, which is what DMX uses.
 - Connect a TTL Serial Adapter with RS-485 function, any USB serial based DMX transmitter will work, to the DATA IN terminals according to the pinout. Either directly, or through DMX/XLR connector
 - Continue:
 - Power up the device, after first ensuring correct polarity. The LED Indicator should light up and stay on.
 - Start the NLED Pixel Controller Interface Software. Ensure Java is updated, check online.
 - Select the COM port that your serial adapter is on.
 - Click Connect. If successful the button text will change to "Connected". Note it doesn't acknowledge the controller is connected, but that the port was opened successfully.
 - Enter all the configuration information for use with your project. Check the device datasheet for specifics.
 - Click the Attempt Send Button and watch the LED indicator on the device.
 - It should change states(on to off) with each byte received and start blinking steadily when all bytes have been received. Note it doesn't acknowledge or confirm received data.
 - Power off the controller.
 - Change the DIP switches to whatever mode is required(serial, DMX or test)
 - Make any connections to your LED strip and data connections.
 - Power the device back on.
- If it does not seem to be functioning correctly, try the process over, if it still doesn't contact support.

Configuration Software(Cont.):

Software Downloaded From:
www.NLEDshop.com/pixelmini



Connect to COM Port, input configuration data, start controller in Command Mode and Press "Attempt Send"

Data Mode:

- WS2811/WS2812/WS2812B, 800KHz
- WS2801/WS2803 or other Standard SPI Types
- LPD8806, SPI with MSB set to 1 and latch bytes.

DMX Address: Starting Address for DMX Reception

DMX Pixel Amount: The amount of pixels to be controlled using DMX-512 input. Can be left at maximum in most circumstances. The amount of pixels to receive in serial control is not constant, data is framed using packet spacing.

Serial Baud Rate: Selected baud rate for use with Serial Control. Should be the same value as the transmitter. For WS2811/WS2812/WS2812B it defaults to 460,800 baud, faster is not possible and slower will alter the maximum amount of pixels that can be controlled. To alter the baud rate for WS2811/12/B pixels from default, input a SPBRG value. See chart below for values. Contact for assistance.

Disable RX LED: Check the box to disable the LED.

End of Frame Length: For Serial Control, defines the time the firmware will wait between bytes to declare end-of-frame. Leave low if possible, but choose a higher value for slower transmissions.

TTL Serial Baud Rate:

The device does not match all common baud rates precisely, due to internal timing. The software lists the common baud rates that a standard serial port will be able to configure to. FTDI and similar serial devices may be able to be set to arbitrary baud rates but, may default to the closest standard value without the user knowing. This chart lists the selected baud rate vs actual baud rate the device is set to.

Closest Common Baud Rate	SPBRG Value	Actual Device Baud	Error %
9,600	207	9,615	0.16%
19,200	103	19,230	0.16%
57,600	34	57,142	-0.79%
115,200	16	117,647	2.12%
230,400	8	222,222	-3.55%
460,800	3	500,000	8.5%
921,600	1	1,000,000	8.5%

The transmitting device should use the Actual Device Baud rate whenever possible. Then the Error % would be 0%.

For Custom User Baud Rates Use This Formula:
 $SPBRG = ((32,000,000 / DesiredBaudRate) / 16) - 1$

Example:

$$SPBRG = ((32,000,000 / 9600) / 16) - 1$$

$$32,000,000 / 9600 = 3333.333$$

$$3333.333 / 16 = 208$$

$$208 - 1 = 207$$

Or Contact For Help