

NLED Pixel Controller Proton

Designed for the entertainment and amusement industries, this pixel controller is quick to setup, quick to install, and offers many features to make implementation easier. Featuring four general inputs that can be used to allow external interactions, such as button and sensor inputs, to run functions on the controller. Such functions include selecting a color sequence, play/pause, intensity dimming, and many more.

Features:

- Supports all the standard pixel chipsets - See List
- Supports up to 512 channels / 170 RGB pixels / 128 RGBW Pixels
- Easy power connection using the high current power pass through, up to 5 amps
- Four input pins, easily interface and run Aurora commands from a PLC, Arduino, or buttons.
- Large internal memory for storing color sequences.
- USB interface, for configuring, loading sequences, and pixel control
- DMX-512, Serial, WiFi, and Infrared Remote control can be added..
- Full Stand-Alone Color Sequence Support Using NLED Aurora Control
- Designed and Manufactured in The United States
- Includes 3 Year Warranty and Lifetime Support

Input Voltage	5 volts to 12 volts
Logic Current Draw	< 50mA
Standard Outputs	512 Channels(170 RGB pixels)
Extended Outputs	3072 Channels
Connectors	Pluggable Screw Terminals, 5.08mm pitch
USB Connector	Micro USB
Data Memory	16 Megabytes
Board Dimensions	45 x 45 mm x 17mm
Enclosure Dimensions	50mm x 50mm x 27mm, plus mounting flanges
Power Pass Through	Up to 5 Amps

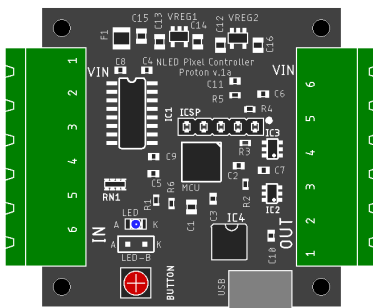


Fig. 1a

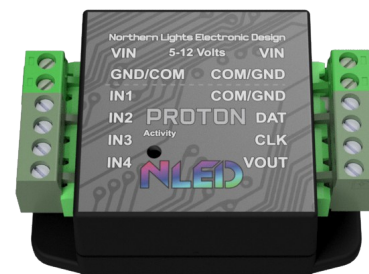


Fig. 1b

Supported Pixel Chipsets

The full list of tested and compatible chipsets is now available online at:

www.nledshop.com/pixelchipsets/

Example List of Supported Chipsets:

WS2801, WS2811, WS2812, WS2812B, WS2813, WS2815

APA102, APA104, AP106, APA107

SM16703, SM16704, UCS1903, UCS2903, UCS2904, P9823

SK6805, SK6812, LPD8806

See the the chart on the first page of this document for the maximum supported pixels.

Need Another Pixel Chipset Supported? Contact Support@NLEDshop.com to request it.

This device supports most of the commonly available pixel protocols. Typically these are referred to as chipsets, many chipsets utilize a similar protocol to control the pixels. Some require only a single wire(DATA or DAT) to control the pixels, some may require 2 wire(CLK & DAT). Each chipset has an unique control protocol, hardware specifications, voltage requirements, and color order. You must select the correct chipset and color order for your pixels. The options can only be changed through the software or USB commands. The software application NLED Aurora Control is used to configure the options and setup the device. See the Configuration Settings page for details.

All NLED pixel controllers use tight timings specific to the chipset. Tight timings are important for stable use, all timings are provided by the manufacturer's documentation and tested on actual pixels of that chipset. That means some chipsets will state or be commonly referred to being compatible with the common WS2812B, selecting that chipset may not work correctly with your pixels if they are, as an example, SK6812.

The Aurora firmware supports Pixel Packet Cloning, which allows pixel strands longer than than the maximum of 512 channels of LEDs to be controlled over stand-alone. It is described later in this datasheet and in the NLED Aurora Control Manual. Example: User wants to control 680 pixels, setup sequences for 170 pixels with a packet clone value of 4. Now pixel 1, 171, 341, 511 will all act the same, as will pixel 2, 172, 342, 512, as will pixel 170, 340, 510, 680.

See the Wiring and Pinout section for details on connecting the pixels.

NLED Aurora Control Software

Visit www.NLEDshop.com/nledaurora for more documentation and download links.

Most NLED controllers are compatible with the NLED Aurora Control software. The software is used to create a multitude of patterns and color sequences on a computer, then upload them to the compatible controller for the controller to run by itself, without a computer connection. The simple GUI makes it easy for anyone to create custom color sequences of any sort and upload them to the controller over the USB, WiFi, bluetooth, or serial connection. Great for any LED project that requires custom color sequences and control. In addition to standalone sequences, it also supports USB Live Control, send packets over USB to the device for direct control of the outputs. In addition it offers a powerful command protocol for interfacing and controlling with the device and the color sequences. It supports single color, RGB, or RGBW LED configurations. Entirely GUI based and easily to navigated.

Color Sequence: A color pattern, contains all the data to control the LEDs. Or dataset(how the color data is stored) that creates colors, patterns, and sequences. Can be single color, RGB or RGBW, all are still color sequences.

Channels: A channel represents a single output on a controller or a single color of a pixel. See controller specifications for information on how many channels each NLED device supports.

Color Channel: Represents a single LED or pixel. That could be either single color, RGB, or RGBW(or RGB+U.V.) A Color Channel is seen as 1 or more Channels. A sequence set to RGB Pixel Data Type will use 3 Channels Color Channel. A LED Icon(on the Graphic Layout tab) or a row on the Timeline tab represent a Color Channel.

Frame: The data stored in each Frame will be displayed on the outputs/pixels sequentially. A Frame can be thought of as a single frame of a video, it holds all the data to create those colors. See Manual for details.

Index: Contains an ordered list of the color sequences that will be uploaded to the controller.

Pixel Data Type: Tells the software and controller the LEDs that will be controlled are either a single color LED(like just blue), RGB, or RGBW(Any 4 colors)

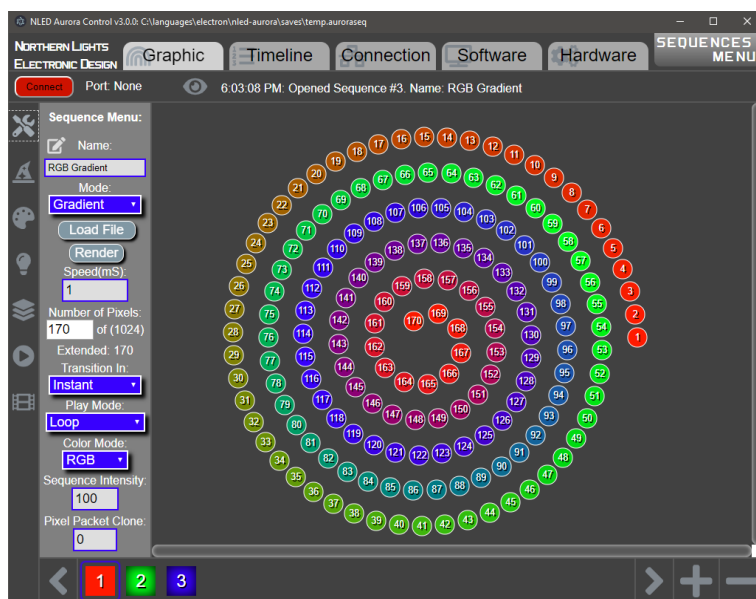


Fig. 3a

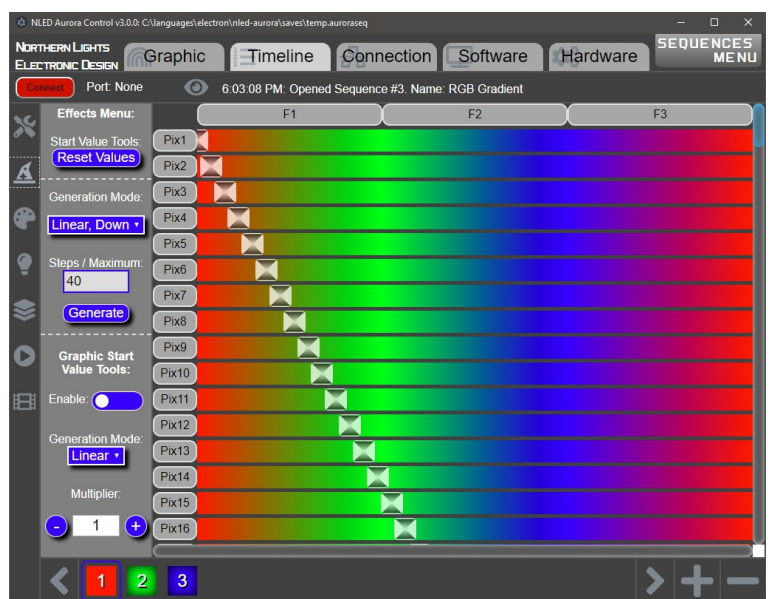


Fig. 3b

NLED Aurora Protocol

Aurora compatible controllers have a specialized command set that is used to control and command the controller from the Aurora software or from user created software applications. Almost every function and feature of the Aurora stack is available through commands. Such as play/pause, output color selection, device status, upload, download commands, and numerous others. Any software language that can communicate with serial ports can interface with Aurora controllers. The command protocol is also so simple that commands can be issued manually from a serial terminal like RealTerm or HyperTerm. The protocol involves simple character exchanges based on ASCII and 8-bit numbers.

Commands can be issued over a USB interface if available, or through the serial interfaces that all NLED controllers offer. This device when connected over USB is an Emulated Serial Port or VCP. This allows computers to see the device as a regular serial port.

Visit www.NLEDshop.com/nledaurora to find the link to the 'NLED Aurora Command Manual' for protocol documentation and more information.

Live Control - USB or Serial

Live control allows data sent from a computer or other external device to control the pixel colors and intensities. Many languages provide communication with serial devices, which allows many options for a user to create custom software to control the connected pixels over USB. Similar to using DMX or serial reception, this allows for live updates of the controller's connected LEDs and pixels. It utilizes the Aurora Protocol to initiate and transmit the data values from the computer software to device.

This feature is similar but distinctly different from the 'Serial Reception' feature. This option does not frame the data in anyway, lost or mis-sent bytes can result in the data becoming out of frame. For USB this is not an issue, but could become an issue with serial based.

Visit www.NLEDshop.com/nledaurora to find details in the 'NLED Aurora Command Manual'

Serial Command Mode Communication

All NLED devices with serial(UART) interfaces are capable of communicating with NLED Aurora Control and other serial devices using a TTL serial interface. The serial interface can communicate in a similar way to the USB interface, all Aurora Protocol commands and uploads function similarly. The serial interface can be a standard USB to serial adapters(FTDI or similar). Or more advanced devices that offer a 'transparent bridge'/'transparent serial' features such as bluetooth modules(RN4871 or similar), WiFi modules(ESP8266 or similar), Arduinos, and other serial devices.

Some NLED controllers have separate serial ports for communicating with external serial devices. Some devices share ports with other data features such as DMX-512. See Device Configurations for details specific to the NLED device. The baud rate and other configurations settings must be configured and correct in order for this feature to function correctly.

DMX-512 Reception

This device supports multiple DMX-512 reception modes, a reception mode can be selected through the Device Configurations. This controller is not focused on DMX reception, and does not have the hardware for direct control from a DMX, as it has no RS-485 transceiver. An addon card for DMX reception is available from the website, and attaches to the underside of the controller.

In order for the device to properly receive DMX data, the Expansion Header Device Configuration must be set to 'DMX Reception' and either Auto-Detect(recommended method) should be enabled, or DMX reception tile should be added to the index from NLED Aurora Control.

Direct 170 RGB / 128 RGBW Control: Direct control of the outputs. Fig. 5a

Channel	Function	Value	Description
1	Value for channel#	0 - 255	0 is 0%, 255 is 100%
...	Value for channel#	0 - 255	0 is 0%, 255 is 100%
512	Value for channel#	0 - 255	0 is 0%, 255 is 100%

Sequence Control: Control stand-alone color sequences with DMX. Fig. 5b

Channel	Function	Value	Description
1	Sequence Number	0 - 255	0 Blanks the All Outputs, 1-32 maps to indexed sequences
2	Speed	0 - 255	0 is Pause, 1 - 255 is the speed used for sequence
3	Position/Fade	0 - 255	See Note 1

Note 1: Position/Fade Control: If the Speed is 0, it allows the position of Instant Sequences to be set by the Position value. As calculated: $(\text{Position} / 255) \times \text{Amount of Data Frames}$. For Fade and Gradient Sequences, it allows the Sequence to run normally for Position value amount of times, which for 8-bit sequences, a value of 255 will run all output channels 1 frame. For Fades and Gradient sequences, the Position value used is the difference between the previous Position value and the current Position value.

RGB Compatibility: All pixels on the strand are set to the same color, not individually. Fig. 5c

Channel	Function	Value	Description
1	Red Value	0 - 255	0 is 0%, 255 is 100%
2	Green Value	0 - 255	0 is 0%, 255 is 100%
3	Blue Value	0 - 255	0 is 0%, 255 is 100%

RGBW Compatibility: All pixels on the strand are set to the same color, not individually. Fig. 5d

Channel	Function	Value	Description
1	Red Value	0 - 255	0 is 0%, 255 is 100%
2	Green Value	0 - 255	0 is 0%, 255 is 100%
3	Blue Value	0 - 255	0 is 0%, 255 is 100%
4	White/Amber/UV Value	0 - 255	0 is 0%, 255 is 100%

Serial Reception

Serial reception can directly control the device's outputs or pixels using a serial transmission device. Such as a standard COM port with TTL translation, an FTDI adapter, Arduino, PIC UART, wireless XBee, ESP8266, or similar. Serial reception is different from 'Serial Command Communication' and the Aurora protocol. This reception mode is a very basic time framed packet reception. There is a maximum of 5mS between bytes and a minimum of 5mS between packets(end-of-frame), to allow the data to latch. A delay between received data bytes of more than 5mS will reset the buffer pointer, framing the data. Partial packets will latch after the 5mS delay, or latching will occur immediately after receiving the last byte of the maximum packet size, which is the is how many channels the device supports.

The device receives data in packets, the size(in bytes) 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. The controller accepts the data in regular RGB order and will reformat it to the required output order(to GRB, BRG etc), if the device supports color re-ordering, which is standard for pixel controllers.

Serial reception can be enabled through the NLED Aurora Control software by adding the tile to the index, or by configuring the device *Configuration Settings*. The device can take several packets to detect and validate the data before starting reception. The default baud rate is 19,200, but the user can set the utilized baud rate using the device *Configuration Settings*. The byte formatting is the standard 8-N-1(8-bits, no parity, 1 stop bit).

Note: This is a different feature from "Serial Command Mode". But they both may use the same baud rate and pin connections.

ID#	Actual Device Baud	Closest Common Baud Rate	8-N-1 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%

*Note: Percentage of error is within usable limits. Matching of exact value is not required.

Fig. 6a

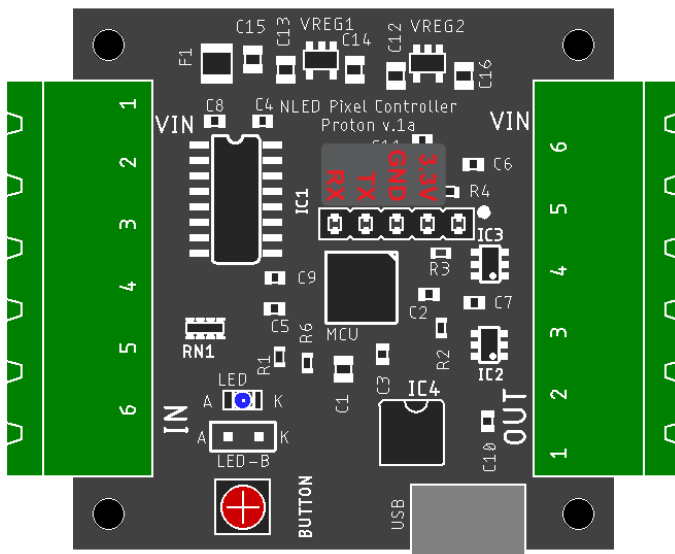


Fig. 6c

RX-TX-GND Header

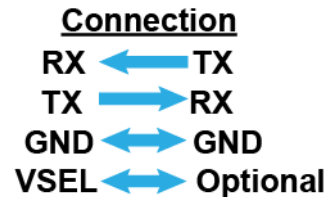


Fig. 6b

More details and information can be found at

<https://www.nledshop.com/downloads/manuals/nled-serial-reception-manual.pdf>

Firmware Updates Using The Bootloader

This device includes a bootloader feature, which allows the firmware on the device to be upgraded with new features and bug fixes using only an USB cable and a small computer program. Firmware updates are encrypted and protected with special code that prevents other devices from being programmed with the update firmware images. And prevents non-NLED firmwares from being programmed onto the devices. Contact Us for help restoring your device to original condition if your microcontroller was damaged, erased, 'bricked', or otherwise not working correctly.

Full instructions, software, and firmware downloads can be found at www.NLEDshop.com/bootloader

If the NLED device has a USB port it uses the 'USB HID Bootloader', otherwise it supports the 'Serial Bootloader'.

To Enter Device Into Bootloader Mode - Button Method: (see last page for access details)

1. With the device powered off, while holding down the button(s), or after making the jumper connection. Got device specific details, please see the wiring and pinout section.
2. Power up the device. Either by connecting the device to the computer with a USB cable, if power over USB is supported. Or by connecting the device to an external power supply.
3. Wait a second or two. If the device has an activity LED, it will start blinking or be turned on.
4. Device will then enter bootloader mode. Follow the instructions found on the manual included with the bootloader software download.

To Enter Device Into Bootloader Mode - NLED Aurora Control Software:

In the NLED Aurora Control software the button on the upper right on the Hardware Tab can be used to enter the device into bootloader mode. Note: Not all devices and firmware versions are compatible

To Enter Device Into Bootloader Mode - Aurora Command Protocol:

Note: Not all devices and firmware versions are compatible

1. Open the correct COM port assigned to your device to be updated. Supports USB or serial.
2. Select any baud rate, doesn't matter
3. Send as ASCII "NLED11"
4. Device will respond as ASCII "a9"
5. Send as ASCII "nled99"
6. Device will respond as ASCII "f0"
7. Send as numbers "140 0 0 0 0" - see the Aurora Command Manual for command details.
8. Device will then enter bootloader mode. Follow the instructions on www.NLEDshop.com/bootloader

More details on the 'Enter Bootloader' Aurora Command, and other commands:

<https://www.nledshop.com/nledaurora>

Wireless Communication With Modules

NLED devices do not directly support wireless methods natively, but can support them a few different ways. The most important requirement is the wireless module supports what is sometimes called 'transparent bridge', 'transparent serial', or 'transparent bridging'. The feature is available out of the box on a few modules, and typically can be loaded onto wireless modules that support custom code. That feature simplifies data transmission and only sends and receives serial data without AT character exchanges or other protocols. Similar to a FTDI or other USB to serial adapter, except over a wireless connection. These typically communicate with a UDP or TCP network connections.

WiFi Options:

- ESP8266, all variants: Requires third-party firmware to be loaded. ESP-Link by Jeelabs is open source and recommended(no affiliation).
- Xbee by Digi: Some options support transparent serial bridges with stock firmware.

NLED Aurora Control supports TCP based WiFi modules. Otherwise control is available through TCP/UDP or through the module's web server.

Bluetooth Options:

- RN4871 by Microchip Inc: This is a low-energy module, and is not compatible with Bluetooth 4.0. It does require more specialized software to communicate. NLED offers sample code for cross platform phone apps.
- Numerous China brand Bluetooth 4.0 modules

External Modules:

- All NLED devices with Expansion Headers support this method.
- The module is connected to RX, TX, GND, and optionally power from the NLED device. See the 'Serial Reception' section on this document for wiring details.
- Typically interfaced using the NLED device's 'Expansion Header', and available Device Configuration settings.
- Module bootloading, resetting, and other features are not typically available.
- To use Serial Command Mode or Serial Reception, the Device Configurations must be set correctly. Ensure the option is enabled(see Expansion Header) and the baud rate is correct and matches your module's baud rate. Typically 115200 baud.

Onboard Modules:

- Some devices offer a wireless module mounted to the device standard.
- These modules typically support module bootloading through the Aurora Protocol.
- This method may offer Device Configurations to select and enable features.
- Typically the Device Configuration settings will be under 'Communication Options' header.
- Other features, such as DMX or serial reception, may be shared and can not be used simultaneously.

All features such as Serial Command Mode and Serial Reception function the same with either external or onboard modules.

Device Configuration Settings

Select The Pixel Chipset Being Used: The most important configuration setting, this option must be the same as the pixels you expect to control. See page Chipset section for pixel chipset details.

Select Correct RGB Order for Pixels: The different pixel chipsets require data to be received in various orders. Not all pixels will expect data in RGB order. It may expect GRB, BRG, or any combination, it varies chipset to chipset. This configuration setting allows the user to select the correct color order for their pixels.

Expansion Header: This device has a header that can be used for addition of hardware for communicating or other uses. Only one of these can be enabled at a time, and may require additional configuration settings.

DMX Reception(Default): This device does not have the transceiver for proper DMX-512 communication, but supports it through addon cards. Will not properly receive DMX unless reception is selected using Auto-Detect option, or through indexing.

DMX Master: Enables the controller to output a DMX-512 universe from the expansion header's TX pin. Requires the addition of the DMX(RS485) addon card with required hardware in order to communicate with standard DMX devices. Optional full or partial packet option, partial packets will be the same length as the channel amount.

Serial Command Mode: As described in a previous section, this enabled Serial Command Mode. And allows Aurora Protocol commands to be issued to the controller from a serial device. Requires the Serial Baud Rate also be set correctly to match the connected serial devices baud rate.

Serial Reception: As described in a previous section, this enables the expansion header to receive(on TX pin) serial data from a connected serial device for direct control of the output. The baud rate and Auto-Detect option should also be set for this to work correctly.

Activity LED Mode: Controls how the activity LED functions, when stand-alone sequences are running and/or a serial or DMX packet has been successfully received.

Serial Baud Rate: Sets the expected baud rate used for serial reception and serial command mode communication. The controllers baud must match the baud rate of the connected serial device.

DMX Reception Mode: Options for DMX Reception modes. See DMX-512 Reception section.

DMX Address: The starting address for DMX reception.

DMX Timeout Release: Sets the option to detect loss of DMX signal and returns to playing stand-alone color sequences. Option allows either to restore the color sequence it had been playing. Or to start playing the Idle sequence once DMX signal is lost. Timer is set to 3 seconds, if a valid DMX packet is not received within that period, the controller will start playing color sequences.

Auto-Detect Communication Mode: Options allow the device to detect and automatically switch from stand-alone to the selected data control method such as DMX or Serial via the Expansion Header.

DMX Master Packet Size: Only used for DMX master transmission via the Expansion Header. Full packets will send the DMX-512 standard of 512 addresses, regardless if data is populated on all addresses. Partial packets will send only the the data that is available, specified by the stand-alone sequence that is playing.

Device Configuration Settings(continued)

Gamma Correction: Enables or disables the use of the internal gamma correction table. See the Aurora Manual for information on this feature or search online for Gamma Correction for an overview.

Communication Options: Selects different communication modes available on the device. No additional hardware or settings are required. Any one of these modes can run con-currently with any Expansion Header modes.

Disabled: All communication options are disabled

Binary Input: Enables the binary input feature. Allows the color sequences to be triggered/selected by pulling the pins low. The pins are pulled high and are active low. There are 4 input pins, allowing 16 sequences to be mapped and selected. The INx pins reference bit mappings in standard binary.

Usage: Using a method of your choice(button, microcontroller, Arduino, PLC), connect the pin(s) to ground/common. When the device detects a pin going low, it starts a 10mS timer, every time a pin changes the timer is restarted. Once the timer elapses, the INx pins are polled and the command is parsed.

Pin to Bit Mapping:

IN1	BIT0
IN2	BIT1
IN3	BIT2
IN4	BIT3

Example bit values and the function. This is not a complete list but can be extrapolated.

BIT3	BIT2	BIT1	BIT0	Action
0	0	0	0	Play idle sequence. Idle is defaulted sequence #1* if not specified.
0	0	0	1	Play sequence #1*
0	0	1	0	Play sequence #2*
0	1	0	0	Play sequence #4*
0	1	1	0	Play sequence #6*
1	0	0	0	Play sequence #8*
1	1	1	1	Play sequence #15*

*Sequence ID numbers are in reference to the index order created in NLED Aurora Control.

This is the usage at the time of this release, but all Aurora commands can be mapped to use the binary inputs. The Aurora command protocol has many commands to control and interact with the color sequences. Contact Support if you would like other functions available to the binary input feature.

Expansion Header Wiring Options

See the Device Configuration Section under the Expansion Header setting.

Find info and details for all the available addon cards online at: www.nledshop.com/addoncards/

DMX Reception:

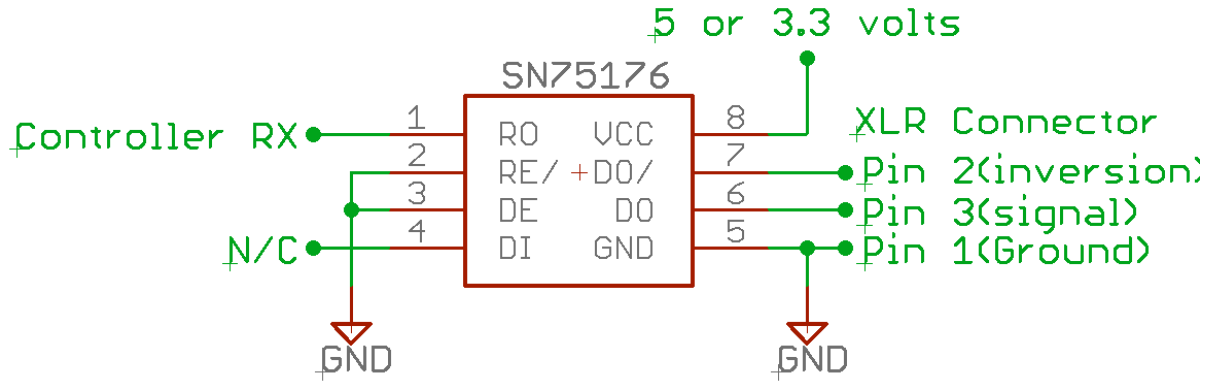


Fig. 11a

DMX Master:

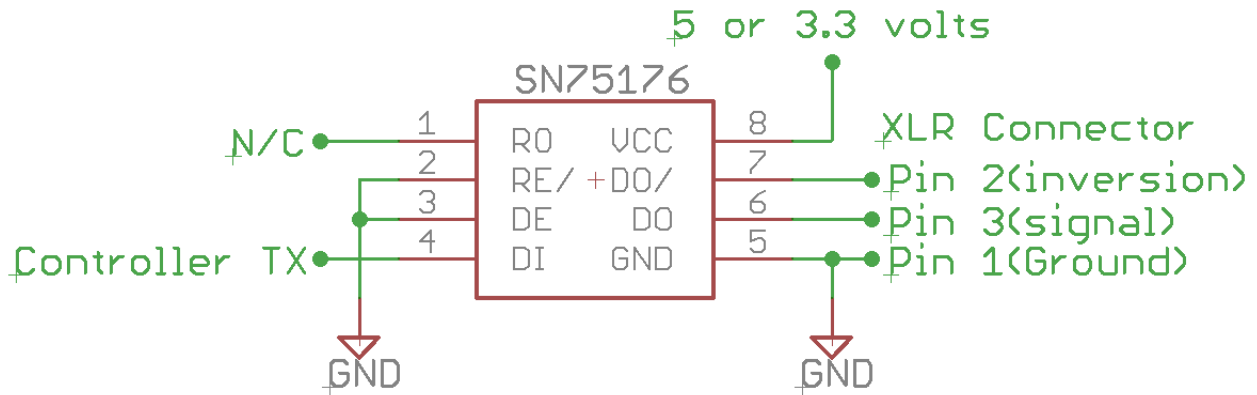


Fig. 11b

Serial Command Mode: See Serial Reception section, same connection requirements.

Serial Reception Mode: See Serial Reception section.

Controller Pinout

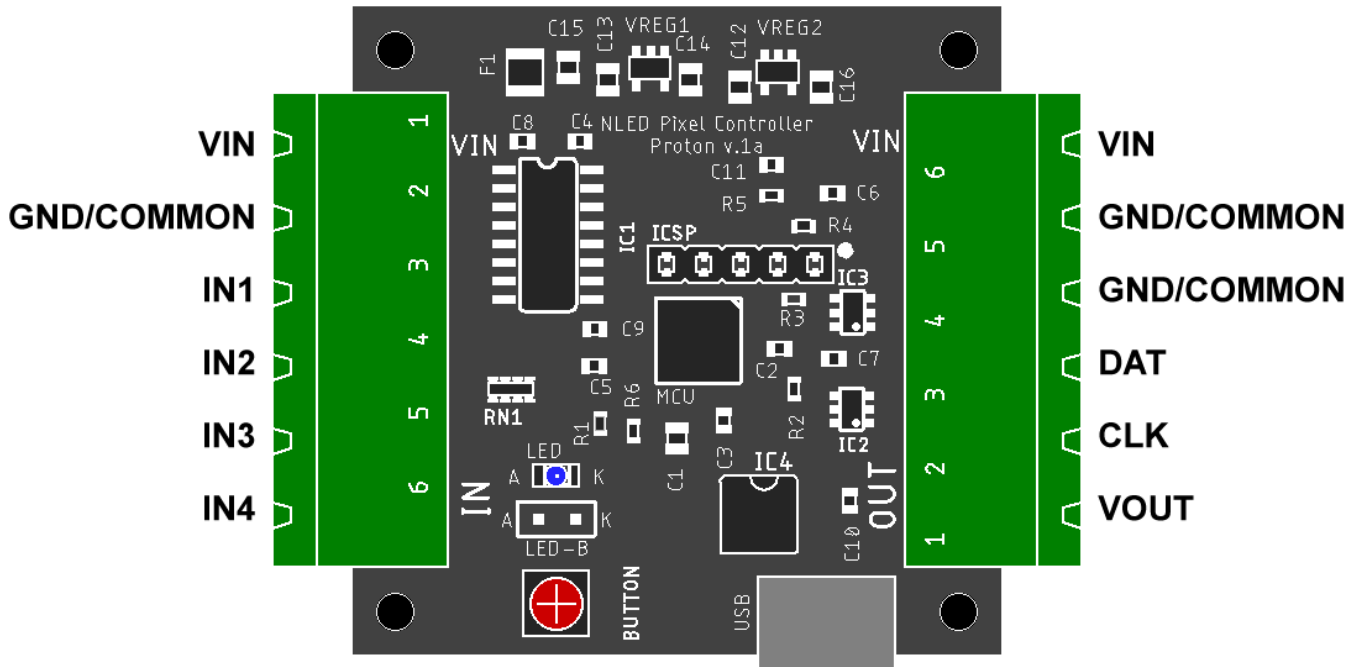


Fig. 12a

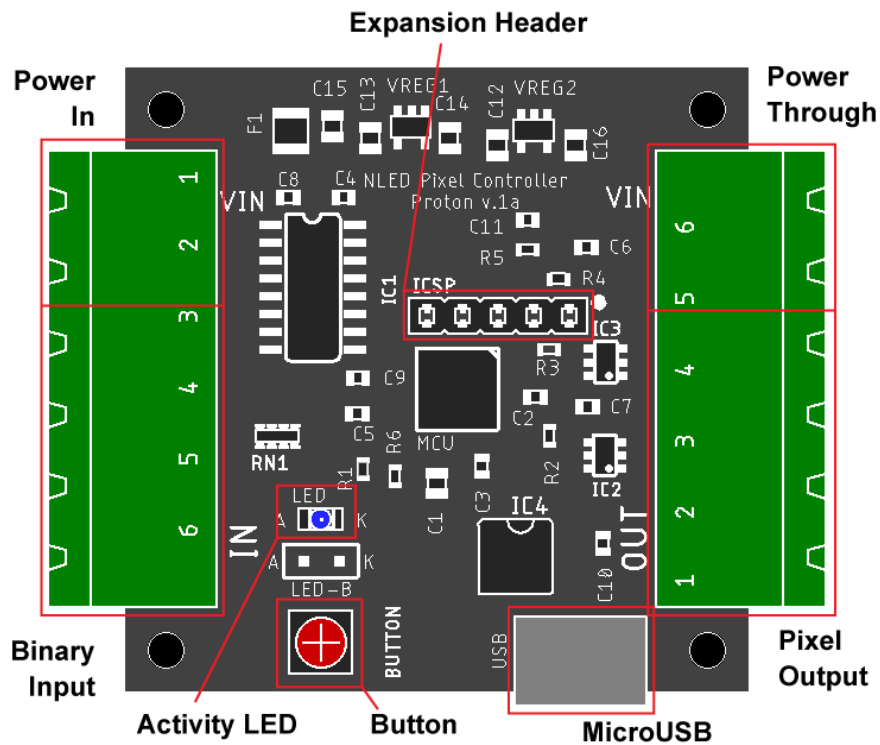


Fig. 12B

NOTE: The Button can not be accessed from outside the enclosure, the lid must be removed for access. See last page of this document. A single press of the button will go to the next sequence on the index. Holding the button at power up will enter the device into Bootloader Mode.

LED Connections

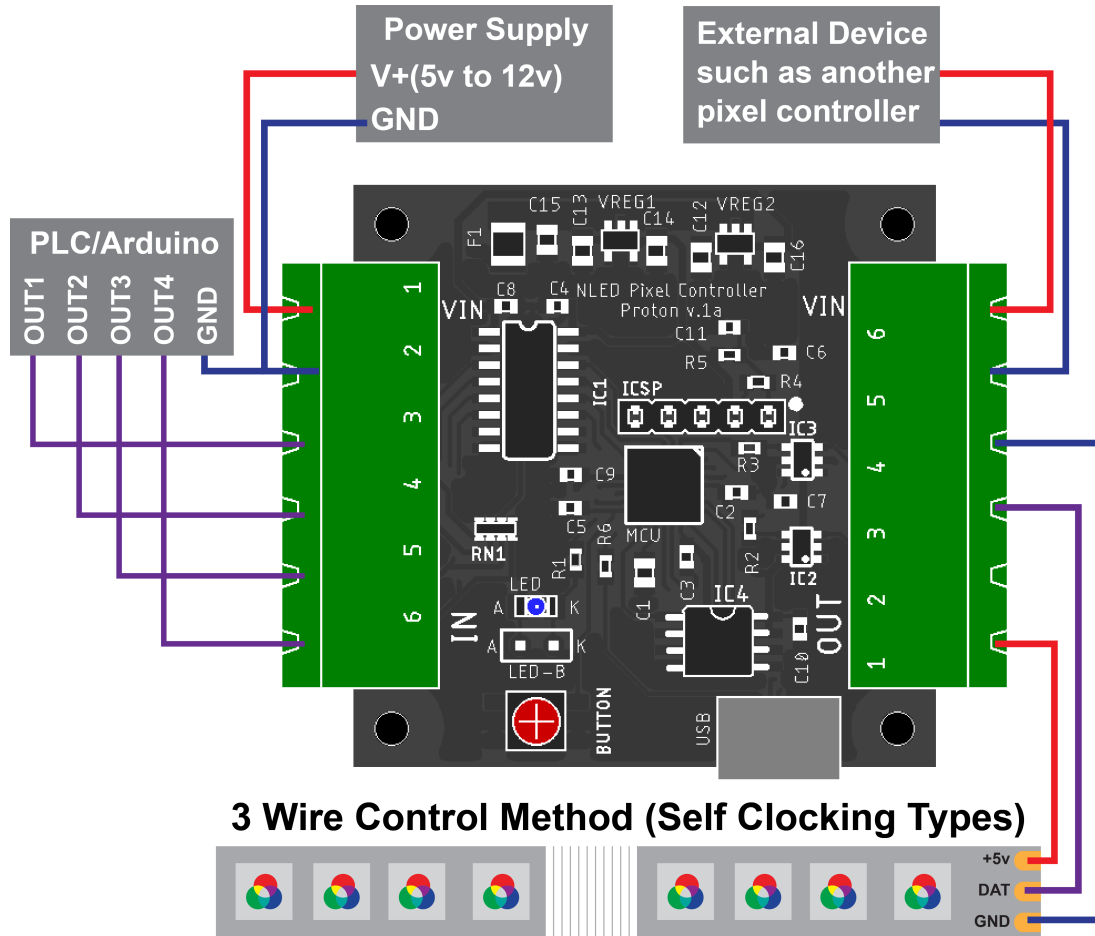


Fig. 13a

Note: 4-wire pixels, also called 'synchronous' or 'clocking' types are connected the same. But with the CLK signal is also connected.

Power Considerations

The controller accepts an input voltage of 5 volts to 12 volts. The controller regulates the input voltage to 3.3v to power the MCU and peripherals. The controller must have adequately supplied power in order to function correctly. Using an under rated power supply or improper power distribution can cause the controller to function incorrectly or not at all.

Power Pass Through:

This device features a power pass through feature. This makes it easy to connect additional devices such as PLCs, Arduinos, constant-voltage(analog) LED strip, other pixel controllers, or most any other device that uses the same voltage as this pixel controller.

This feature is essentially a way to simplify wiring. It prevents the need to fit multiple wires into each pluggable terminal, and allows devices to be easily swapped without additional connectors or clunky wiring.

This image shows the internal connections of the controller. All the connections shown in red are connected together, and any terminal position can be used interchangeably. Same for the lines shown in blue, which indicate the GND/ground/common terminal positions.

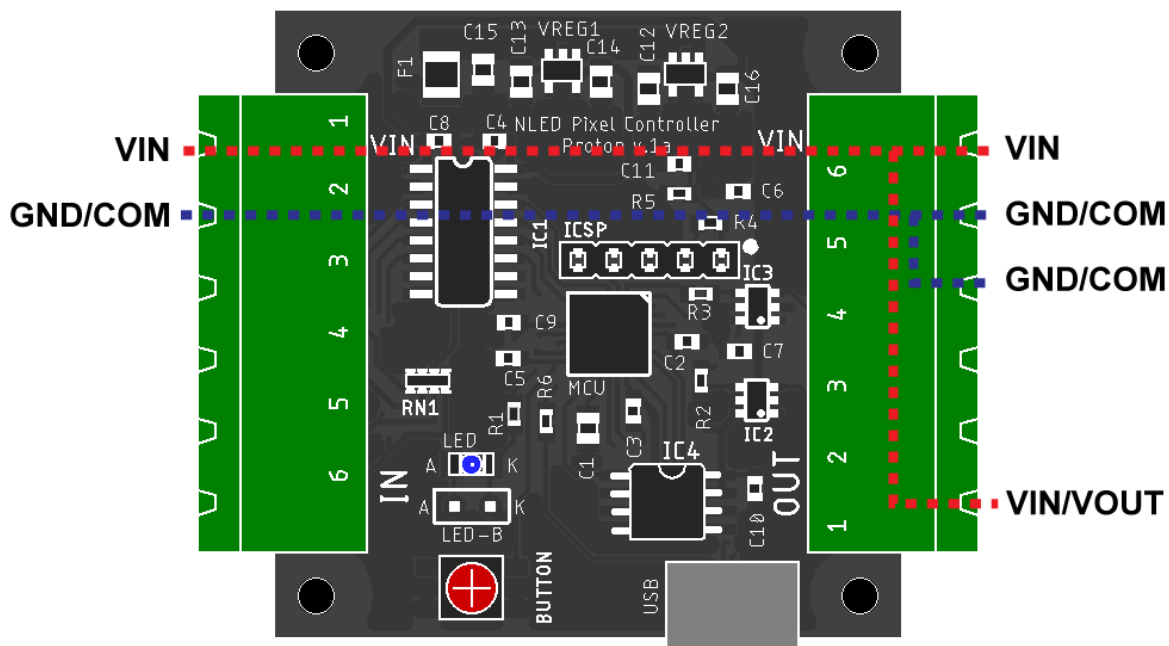


Fig. 14a

Connector Options

Pluggable Screw Terminal: This is the standard offering for this device. The power plug terminals are 2-pin, and the Binary Input is a 4-pin, and the Pixel Output is also a 4-pin. These plug terminals accept wire between 12AWG and 24AWG, depending on the wire gauge, multiple wires can be connected to the same terminal. Requires a small flat-blade or slot screw driver.

Bare/Unpopulated: Available by special order, the input and output terminal headers are not installed. The bare solder pads can be used to directly attach wires or other connectors.

Standard Screw Down: Available by special order. Regular screw down terminals can be requested.

Dimensions and Mounting

A 3D .step model can be found on the device's webpage. Connector option drawings are not provided.

Enclosure mounting details can be found on the manufacturers website: <https://www.polycase.com/sn-22>

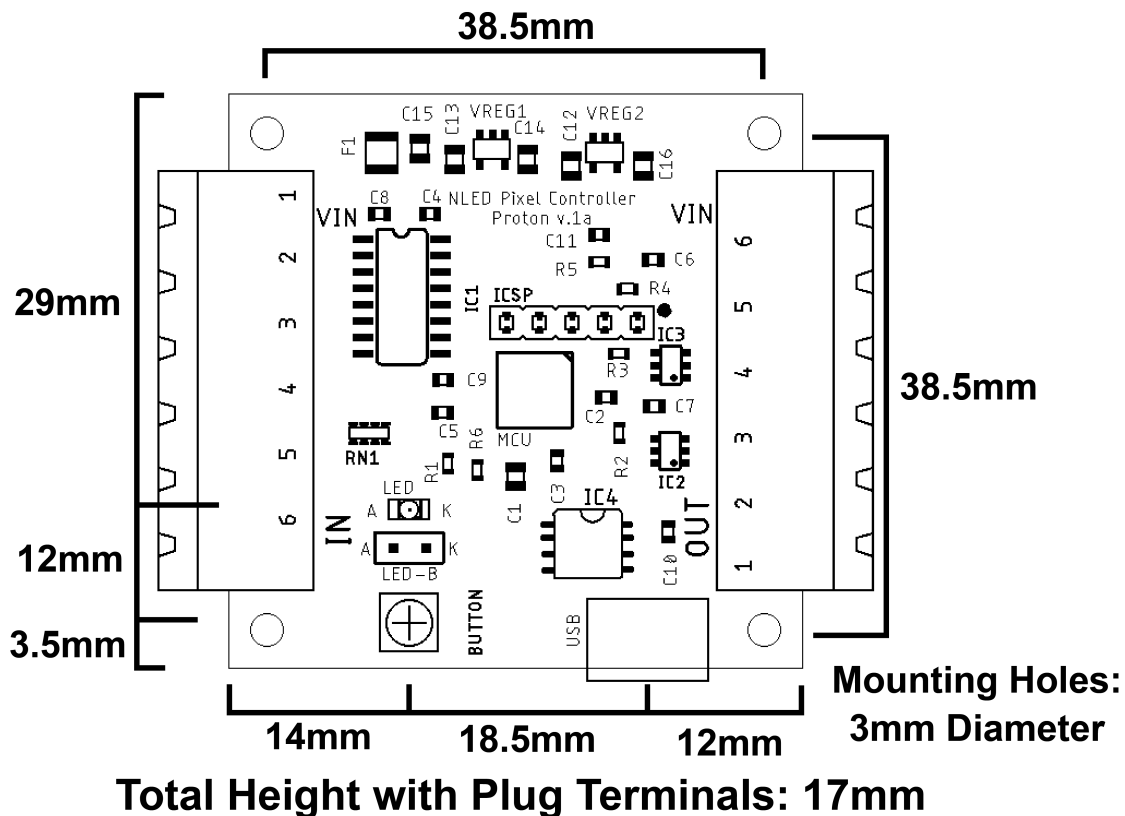


Fig. 15a

Mounting Notes:

Do not allow the PCB to touch anything conductive.

Do not allow foreign material to fall onto or accumulate on the device.

The device should be kept dry and clean. If the device becomes wet or dirty, do not use until it is cleaned and dried. Contact Us for assistance.

Enclosure

This device is compatible with a customized off-the-shelf enclosure. The enclosure is made of ABS and has mounting flanges for easy and secure mounting.

3D models and mechanical drawings can be found on the device's webpage, and from the enclosure manufacturer's webpage.

The enclosure can be opened to access the device's button, the button is used to enter the device into Bootloader Mode. And during normal usage, the button selects the next color sequence in the index.

Open The Enclosure:

1. The controller does not need to be unmounted.
2. Optional: Unplug the terminal blocks from the IN and OUT side.
3. Grasp the lid on two sides without the in/out receptacles.
4. Push one side inwards, do not squeeze both sides.
5. Then pivot the pushed in side upwards. With the opposite side staying in place.
6. The device stays captive in the lid once the lid is removed.
7. Carefully push the device out from the lid.
8. The device can now be accessed, including the button.

www.polycase.com/sn-22
www.NLEDshop.com/pixelproton

Device Setup

Setup: It is recommended you also read the NLED Aurora Manual found in the download folder.

1. Connect the controller to a computer using a micro USB cable.
2. The controller is bus powered and should power up. Wait for the computer to load the driver (most systems should not need to install the driver or check the INF driver folder in the software folder).
3. Start the software, navigate to the hardware tab, on the left side select the controller's COM port.
4. The device should identify itself, and the Connect button should turn from red to green. The hardware configuration setting controls will also be displayed.
5. Adjust the configuration setting controls on the Hardware Tab to suit your project see the Device Configurations section for details.
6. Once the configuration settings are adjusted, navigate to the Connection Tab, and click the "Upload Configurations" button, you will be notified of a successful upload.
7. Connect your LED pixels of choice as described in the LED Connections section
8. Apply suitable power to the controller and pixels, see the Controller Pinout section. The controller and pixels should start running a stand-alone color sequence, if not press the button or load new sequences.

Common Issues and Troubleshooting

Most issues can be resolved by power off the device, waiting a few seconds, and powering it back up.

Problem: Device with connected LEDs does not produce any light.

First ensure the LEDs are properly powered and wired to the controller and power source. Try to cycle through the Sequences or modes using the button. And ensure you have selected the correct pixel chipset.

Problem: Device connected, LEDs are strobing or not working

Ensure you have selected then uploaded the correct Pixel Chipset configuration setting.

Problem: Device appears to be “bricked”, unresponsive to user inputs and/or communication(USB etc)

Attempt to upload a new firmware image via the Bootloader, see page 9 for details.

Problem: USB connection is not being established.

Check to make sure the USB cable is plugged and seated correctly. Check for connection via NLED Control or your operating systems device manager, it should be listed as a COM port.

Problem: Operating systems indicates the driver can not be found.

The driver can be found possibly through Windows, or by locating and directing the operating system to the INF driver file found in the NLED Aurora Control folder.

Problem: Connected Pixels strobe or flicker

Ensure you have selected the correct pixel chipset. Some chipsets will be sold as 'WS2812B compatible' but may actually be another similar type, such as SK6812. Try other chipset options. And ensure your power supply is the correct voltage, cheap power supplies may specify '5 volts' but may actually be 5.2 volts(or lower or higher) which is not compatible with 5 volt pixels.

Have Any Ideas For Future Updates?

Northern Lights Electronic Design, LLC is constantly looking to make our products better and improve upon our designs. If you have any ideas for future products, updates to current products, or features that you would find useful, please Contact Us at Sales@NLEDshop.com. There is a good chance your ideas could be utilized, you could also receive coupons or free items for your feedback.

NLED is available to create new designs and derivatives of current designs customized to the clients requirements, please Contact Us with your specifications.