

## NLED Pixel Controller Electron

The Pixel Controller Electron is designed to be the smallest, most versatile pixel controller available with the ability to control most any type of addressable pixel. It features a 3-axis accelerometer for feedback and is optimized for flow toys, wearable applications, and dynamic lighting applications. Using the free software NLED Aurora Control, the user can create endlessly customizable color sequences and patterns, as well as P.O.V(Point of Vision) graphics. With 16 megabytes of onboard flash memory, accessible over micro USB and through addon WiFi or Bluetooth modules. No need for SD cards, readers, or extra equipment, all that is required is a micro USB cable same as your smart phone uses. It is compatible with voltages as low as 2 volts, no need for boost circuits or extra batteries. In addition to the the powerful stand-alone features, it also is capable of direct control of the outputs through TTL serial from external devices such as X-Bees, Arduinos, and FTDI adapters. It also supports DMX-512 reception, with a special stand-alone sequence control method, which allows sequences to be triggered and controlled. Designed and manufactured in the U.S.A. to high standards, this controller offers a 3 year warranty and satisfaction guarantee.

### Features:

- Up to 512 Channels in Stand-Alone Usage and Data Reception.(Future updates will allow 1024)
- Supports Multiple Pixel Chipsets - WS2801, WS2811, WS2812, WS2812B, LPD8806, LPD6803, LPD1886, TM1803, APA102, APA104, APA106, P9823, SK6812, and more. See page 2 for details.
- USB Connectivity - Control thousands of pixels over USB
- Customizable Stand-Alone Color Sequences using NLED Aurora Control Software.
- Utilizes a 3-Axis Accelerometer great for flow toys such as poi, staves, or hoops.
- Supports P.O.V., Point of Vision sequences. Display images in the air.
- Incredibly small form factor, 0.6" x 1.25" x 0.2" thick. Fits in 5/8" ID tube.
- Onboard 16 megabyte memory, NO SD card required. Accessible via native USB or addon WiFi/Bluetooth
- Supports DMX-512 with add-on transceiver. With DMX stand-alone sequence control.
- Supports DMX-512 Master transmission via expansion header or Output header instead of pixel protocol.
- Supports TTL Serial at all baud rates, control from an Arduino or other serial devices.
- Supports WiFi via an ESP8266, XBee, or similar products. Or Bluetooth via modules.
- Low voltage, 3.3v to 12 volt support. Power it and LED pixels from a single lithium ion battery.
- Bootloader Support, easily apply future upgrades and bug fixes, using only a USB cable.
- Designed and Manufactured in The United States. Includes 3 Year Warranty and Lifetime Support.

### Specification:

Input Voltage(VIN)	3.3v - 12v
Logic Current Draw	< 50mA
Standard Outputs	768 channels(256 RGB pixels)
Extended Outputs	2048 channels*
Connectors	Various**
USB Connector	Micro USB
Sequence Data Space	16 MegaByte
Output Connector Spacing	0.1"/2.54mm
Overall Dimensions	1.25" x 0.6" x 0.16"

\*See Mass Control firmware and Packet Cloning

\*\*Screw-down terminal, right angle polarized header, unpopulated(bare) or JST-SM harness, see pg. 10

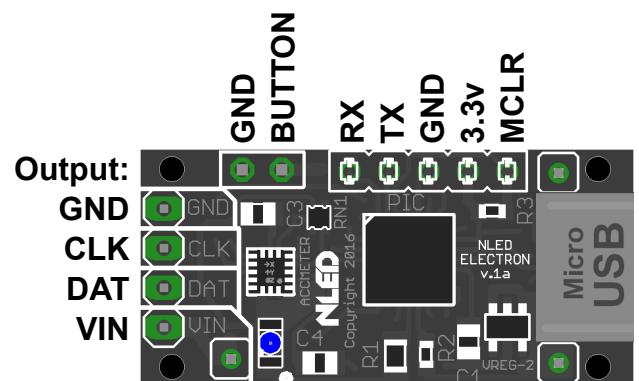


Fig. 1a

See Pg. 10

For Detailed Layout Information

## Supported Pixel Chipsets

Fig. 2a	Chipset Name	Color Bits*	Data Rate	PWM Freq.	NLED Protocol	Wires	Color Mode	Max Aurora Channels	Max Mass Chans.
	WS2811, WS2812/B	8-bit	800Khz	400Hz	A(C,E,F)	3	RGB	512	TBD
	WS2813	8-bit	800Khz	2KHz	A,B,C,D	3	RGB	512	TBD
	SK6812	8-bit	800Khz	400Hz	A(C,E,F)	3	RGBW	512	TBD
	APA102	8-bit	20Mhz	20Khz	SPI-B	4	RGB	512	TBD
	SK9822	8-bit	20Mhz?	4.7KHz	SPI-B	4	RGB	512	TBD
	WS2801 / WS2803	8-bit	2.66MHz	2.5kHz	SPI-A	4	RGB	512	TBD
	APA104 / APA106 / P9823	5-bit	588Khz	400Hz	B	3	RGB	512	TBD
	LPD6803	7-bit	10MHz	N/A	SPI-C	4	RGB	512	TBD
	LPD8806	12-bit	8MHz	4kHz	SPI-D	4	RGB	512	TBD
	LPD1886	8-bit	16Mhz	400Hz	E	3	RGB	512	TBD
	TM1803/TM1804	8-bit	1Mhz	400Hz	E	3	RGB	512	TBD
	APA107 / SK6805	8-bit	833KHz	400Hz?	F(A,C,E)	3	RGB	512	TBD
	UCS1903(Low Speed)	8-bit	400Khz	400Hz	C	3	RGB	512	TBD
	UCS2903/UCS2904(Hi)	-	800Khz	1.6Mhz?	D(B,C)	3	RGB/RGBW	512	TBD
	DMX-512(not a chipset)*		-	-	DMX-512	-	-	512	512

Need Another Pixel Chipset Supported? Contact [Support@NLEDshop.com](mailto:Support@NLEDshop.com) to request it.

**Color Bits:** Same as PWM Resolution, current firmware will down-convert higher resolutions to 8-bit

**Data Rate:** How fast data can be sent to the pixels.

**PWM Frequency:** The frequency of the PWM cycles per second.

**NLED Protocol:** Internal NLED IDs for the various protocols. Used to referen

**Wires:** How many wires are required for connection. 3 wires is GND-DAT-VIN, 4 wires is GND-CLK-DAT-VIN

**Color Mode:** Either single color, dual color, RGB or RGBW light output.

Each color is a channel. Single Color = 1 channel, RGB = 3 channels, RGBW = 4 channels

**Max Aurora Channels:** Number of channels(not pixels) that the device can control with the Aurora stand-alone color sequences. Or with Live Control control mode through Aurora protocol.

**Max Mass Channels:** Requires the bootloading of special firmware. The firmware uses the same Aurora protocol for communication but designed to support more channels. At least 1024 of all chipsets is supported.

\*A new feature for version 2 is the ability to utilize the Pixel Output for DMX-512 master transmission. That can be set under the "Select The Pixel Chipset" configuration, then rather than sending out data for a pixel to receive, it will send out(over the DAT or CLK header) a DMX-512 universe at TTL levels. That signal can not be directly connected to DMX receivers(it can but not reliably) and would have to be converted to RS-485, or directed into a compatible controller(such as the Pixel Controller Mini or Micro)

**THIS LIST IS NOT COMPLETE, PLEASE SEE WEBSITE IF CHIPSET IS NOT LISTED**  
**Visit [www.NLEDshop.com/pixelchipsets](http://www.NLEDshop.com/pixelchipsets) for the up to date list of supported chipsets.**

## **Pixel Chipsets Continued**

Many of the most popular pixel chipsets are compatible with this controller. Each chipset has a 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. See the Configuration Settings page for details.

The pixels can be directly controlled over USB using USB Live Control. The standard Aurora compatible firmware supports 512 channels (170 RGB pixels). In addition, there are separate firmware versions (Mass Pixel Control) available that allow the controller to control up to 2048 pixels in a single strand over USB. The Mass Pixel Control firmware has limited stand-alone capabilities, the File Play color sequence is available.

The Aurora firmware supports Pixel Packet Cloning, which allows pixel strands longer 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.

## NLED Aurora Control Software

Most NLED controllers are compatible with the NLED Aurora Control software. The software is used to create a multitude of patterns and sequences on a computer, then upload them to the compatible controller for the device 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 command structure for interfacing and controlling with the device. It supports single color, RGB, or RGBW LED configurations. Entirely GUI based and easily to navigated.

This software is fully compatible with up to 768 channels(256 RGB pixels, 192 RGBW pixels) in stand alone mode and the communication methods.

**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.

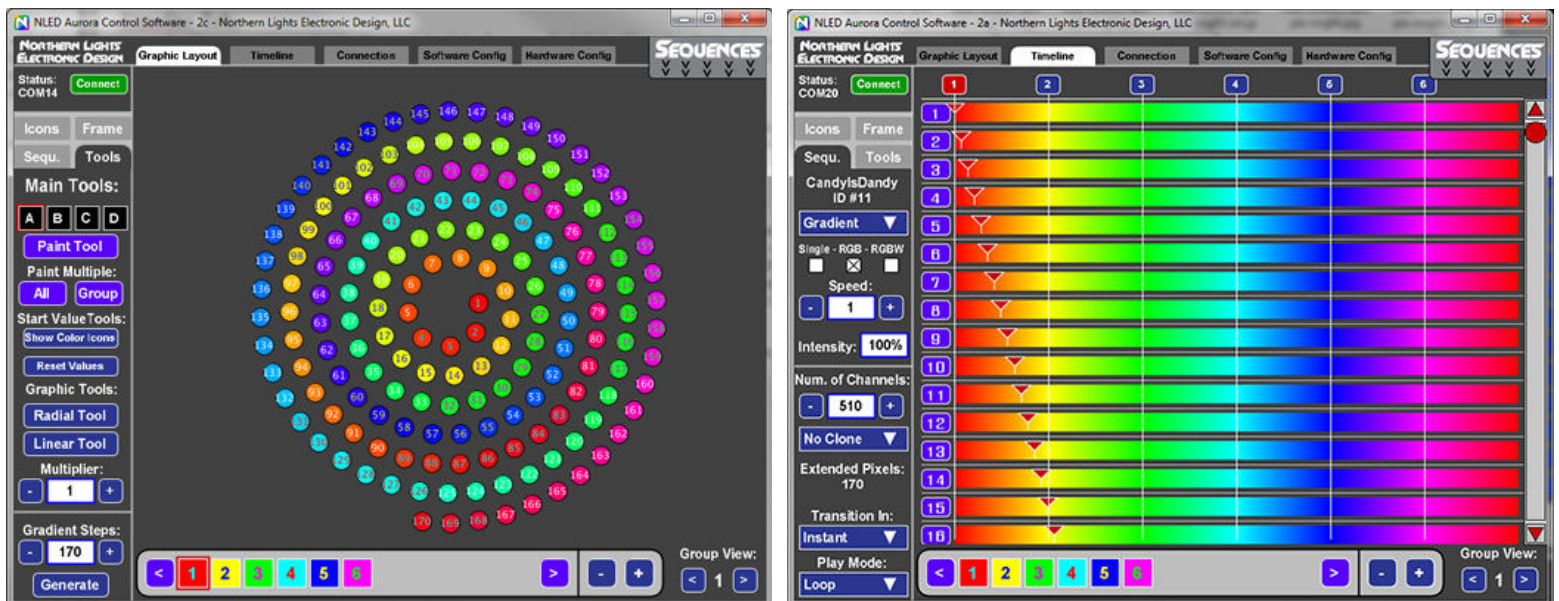
**Color Channel:** Represents a single LED or pixel. That could be either single color, RGB, or RGBW(or RGB+U.V.) A Color Channel is seen as 1 or more Channels. A sequence set to RGB Pixel Data Type will use 3 Channels Color Channel. A LED Icon(on the Graphic Layout tab) or a row on the Timeline tab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

Visit [www.NLEDshop.com/nledaurora](http://www.NLEDshop.com/nledaurora) to download and to find more information.



Visit [www.NLEDshop.com/nledaurora](http://www.NLEDshop.com/nledaurora) to download and to find more information.

## NLED Aurora Protocol

The NLED Aurora Protocol offers a command interface to communicate with NLED controllers. Almost every function and feature of the Aurora stack is available through commands. Such as play/pause, output color selection, device status, upload and download commands, and numerous others. The commands can be executed from any software or program that can connect to the device. Communication involves a simple byte exchange and acknowledge, then the command byte and 4 data bytes are sent to the device. Commands can be issued over a USB interface is available or through the serial interfaces that all NLED controllers offer.

Please see the software documentation for command details and usages.

Visit [www.NLEDshop.com/nledaurora](http://www.NLEDshop.com/nledaurora) for the up to date list of documents.

## USB Live Control & Commands

Live control over USB allows data sent from a computer to control the pixel colors/intensities. 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. 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. Live control is started through NLED Aurora Control commands. (Command sheet available for download) The method is very simple, send the USB Live Control command, which includes the amount the channels that should be controlled, then send the data in packets the size of the channel amount that was sent with the command.

## Serial Communication

As briefly mentioned in the NLED Aurora Control section this controller offers Dual Command Mode configuration(see Configurations Settings page). This allows the user the option to command the controller using USB as the primary communication port for commands and uploads. Or allow commands to be communicated from TTL serial devices and modules, such as WiFi, FTDI, Arduinos, or Bluetooth.

**Disabled(Default):** Aurora commands, such as sequence uploading and sequence control, can only be issued over the USB interface. All other serial and DMX functions work as normal.

**Enabled:** The controller will disable serial reception, DMX reception, and DMX master transmission. This allows for external TTL serial devices such as FTDI adapters, WiFi modules(X-Bee, ESP8266), Bluetooth modules, or Arduino's to issue commands to the controller. The USB interface works concurrently with the external serial module and either can send and receive commands.

## DMX-512 Usage

The device supports multiple DMX-512 reception modes, a reception mode can be selected through the configurations. There are two ways to enter into DMX reception, either by selecting a DMX Sequence through the Index or setting the Auto-Detect feature to DMX.

This controller is not focused on DMX reception, and does not have the hardware for direct control from a DMX universe, 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.

Common method for connecting transceiver for DMX-512 reception or for RS-485 Serial Reception.

A common SN75176 Differential Bus Transceiver can be used if 5 volts is available, otherwise the controller runs on 3.3v and a SN65HVD75DR would be compatible.

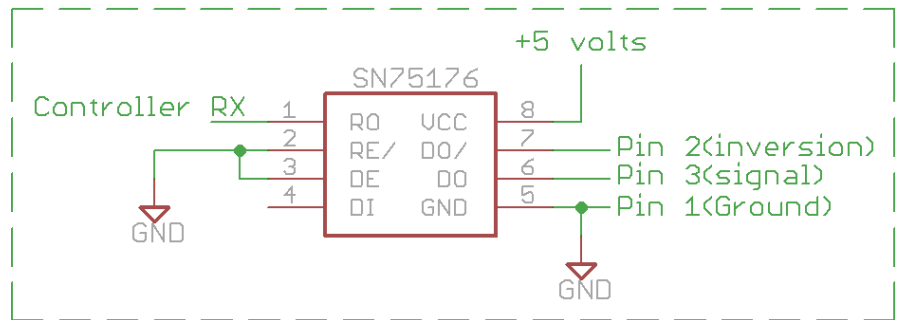


Fig. 4a

Or Check the Store for the Addon card.

### 512 Channel Mode, 8-bit: Standard Direct Control

Channel	Value	Description
1	0 - 255	0 is 0%, 255 is 100%
.....		
512	0 - 255	0 is 0%, 255 is 100%

### 3 Channel Mode: Basic Sequence Control(od6)

Channel	Function	Value	Description
1	Sequence #	0 - 32	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	If Speed is greater than 1, usage is ignored. See Note 1.

Need something different? Custom? Contact [Support@NLEDshop.com](mailto:Support@NLEDshop.com) for Help

## Auto-Detection, DMX or Serial

One of the configuration options allows the device to detect valid serial data or DMX signal and automatically switch from stand-alone sequences to the selected data signal type. For DMX it will take a few packets of good data for the device to validate the signal and then switch over. Serial auto detection is similar, it must detect several packets of serial data at the correct baud rate for it to validate the signal and switch over from stand-alone sequences. The configuration can be changed in NLED Aurora Control software or through NLED Aurora USB commands. DMX master transmission over rides auto-detect usage.

## Serial Reception Usage

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 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. 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(512th) byte.

Serial Reception can be enabled through the NLED Aurora Control software by adding it to an index, or by configuring auto-detection. Each method requires the use of the software. In the software, on the Hardware Tab, select the Auto-Detect Serial option to automatically enter Serial Reception when valid data is received. The device can take several packets to detect and validate the data before starting reception. The stock baud rate is 19,200, but the user can set the utilized baud rate using the software through the hardware tab by selecting the desired baud rate from the drop down menu. The byte formatting is the standard 8-N-1(8-bits, no parity, 1 stop bit) The controller accepts the data in regular RGB order and will reformat it to the required output order(to GRB, BRG etc) as per the configuration setting.

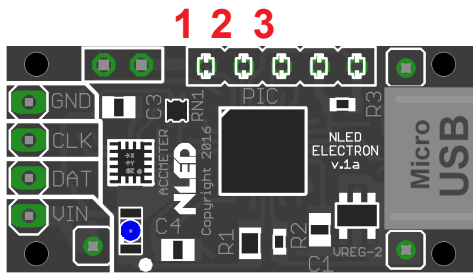


Fig. 4c

### RX-TX-GND Header Connection

Device:

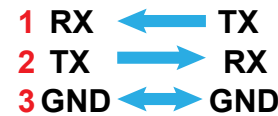


Fig. 5b

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. 5a

8-N-1

Controller supports both 3.3 volt and 5 volt logic levels.

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

Additional details and graphics can be found at:

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

## Firmware Updates Using The Bootloader

Visit [www.NLEDshop.com/bootloader](http://www.NLEDshop.com/bootloader) for downloads and additional documents.

This device includes a bootloader feature. It 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.

After the device is connected, please follow the instructions found at [www.NLEDshop.com/bootloader](http://www.NLEDshop.com/bootloader). That is also where the software download links can be found.

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. If a firmware image is loaded onto a microcontroller without the special protection code it will respond to commands and USB will work, but the outputs will be off. Contact Us for help restoring your device to original condition if your microcontroller was damaged, erased, 'bricked', or otherwise not working correctly.

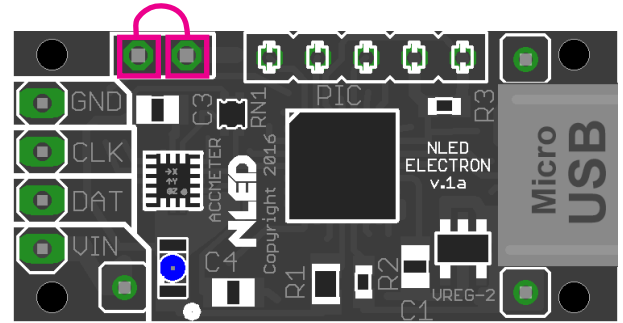


Fig. 6a

The bootloader entry method is:

- With the device powered off, while holding down the button or after making the jumper in Fig. 8a
- Power up the device. Either by connecting the device to the computer with a USB cable. Or connect external power.
- Wait a second or two.
- The device's notification LED should be full on if it entered bootloader mode and will blink steadily if in bootloader mode and successfully communicating with the host computer. When the software is started (or if it is already started) it will automatically connect to the device if one is found.

Or the device can enter the bootloader mode via USB commands. Either from NLED Aurora Control, using the button on the upper right on the Hardware Tab.

Fig. 6b

The command is:

**140 0 0 0 0**

To Enter a command:

- Open the correct COM port
- Select any baud rate, doesn't matter
- Send as ASCII "NLED11"
- Device will respond with "a9"
- Send as ASCII "nled99"
- Device will respond with "f0"
- Send as Numbers "140 0 0 0 0"
- Device will then enter bootloader mode (Note not all devices and firmware versions are compatible)

Example shown is the third party software RealTerm. (NOT AFFILIATED)

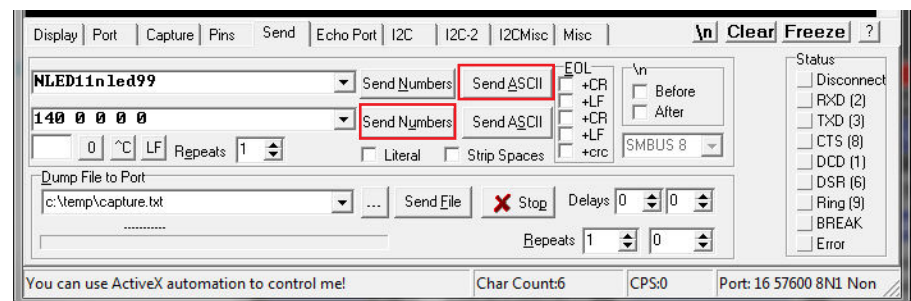
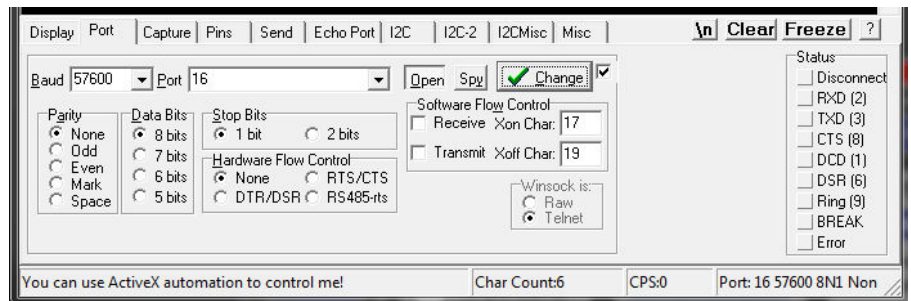


Fig. 6c



## **3 Axis Accelerometer Usage**

The onboard accelerometer can be utilized for movement speed, movement detection, and double tap presses. Its usage and function can be enabled or disabled through the configuration settings. Best for flow toys such as poi, staves, and hoops. It also offers the ability to automatically turn off the controller when not in use and to turn it back on when movement resumes. The configurations section on page 8 has the details.

**Automatic Shut Off/Resume & Time Out Value:** This setting allows the device to detect when its not physically moving and shut off the LEDs/Pixels, saving power. Moving or shaking it will wake it back up and it will resume where it left off. Good for battery powered applications.

**Enable Accelerometer Speed Adjustment:** Option allows the controller to adjust the speed of POV and Instant color sequence based on the movemenet/rotation of the controller. Best or flow toy projects.

**Enable Accelerometer Double Tap Button Press:** Allows the controller to detect a user tapping the controller twice and will emulate a button press. Which would change the color sequence to the next one in the index. There are 3 sensitivity modes, from very sensitive to more robust.

## **Live Control & Aurora Command 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. It is a simple character exchange protocol based on ASCII and 8-bit numbers. Many NLED controllers have USB compatibility, the USB drivers make it appear as a standard legacy serial or COM port to computer. Other controllers may require a USB to Serial Adapter(such an FTDI) or other form of TTL(5 volt) serial converter in order to interface with the NLED controller. Some NLED controllers can use TTL serial and USB interchangeably and transparently to computer side software. The feature is listed as “Dual Command Mode” and must be selected through hardware configurations, it may interfere with other data communication methods such as DMX or Serial, see device datasheet. Whether the device is connected over USB or serial, the command protocol works the same and is transparent to the controller and software.

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 from a serial terminal like RealTerm or HyperTerm.

Details and specifics can be found in the “NLED-Aurora-Control-Command-Sheet-vxx.pdf”

And an example software application can be found separately.

## **Device Configurations Settings**

Since this device is not compatible with an external LED display or other method of user input the only way the configurations can be changed is through the software NLED Aurora Control. The configurations affect how the controller functions, what pixel chipset it expects, notification LED options and more. After connecting the device to the software these options can be found on the Hardware Config tab.

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

**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.

**Select The Pixel Chipset Being Used:** The most important configuration setting, this option must be the same as the pixels you expect to utilize. See pg. 2 for compatibility list.

**DMX Address:** The starting address for DMX reception.

**DMX Reception Mode:** Options for DMX Reception modes(see pg. 4)

**DMX Master Mode:** Enables the controller to output a DMX-512 universe from the expansion header's TX pin. Does not have a differential driver and is a TTL signal. Optional full or partial packet option, partial packets will be the same length as the channel amount. Only 8-bit mode enabled for this controller.

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

**Auto-Detect Options:** Options allow the device to detect and automatically switch from stand-alone to the selected control method(DMX or Serial)

**Automatic Shut Off/Resume & Time Out Value:** This setting allows the device to detect when its not physically moving and shut off the LEDs/Pixels, saving power. Moving or shaking it will wake it back up and it will resume where it left off. Option for either blank the pixels or start the Idle Sequence. Great for battery powered applications.

**Enable Accelerometer Speed Adjustment:** Option allows the controller to adjust the speed of POV and Instant color sequence based on the movement/rotation of the controller. Best for flow toy projects.

**Enable Accelerometer Double Tap Button Press:** Allows the controller to detect a user physically tapping the controller twice and will emulate a button press. Which would change the color sequence to the next one in the index. There are 3 sensitivity modes, from very sensitive to more robust.

**Dual Mode Communication:** Allows a TTL serial device, such as a FTDI, X-Bee, Arduino, or ESP8266 to interface and command the controller over the RX/TX/GND serial connection. Serial control reception, DMX reception, and DMX transmission are disabled. USB remains enabled, both modes can run concurrently.

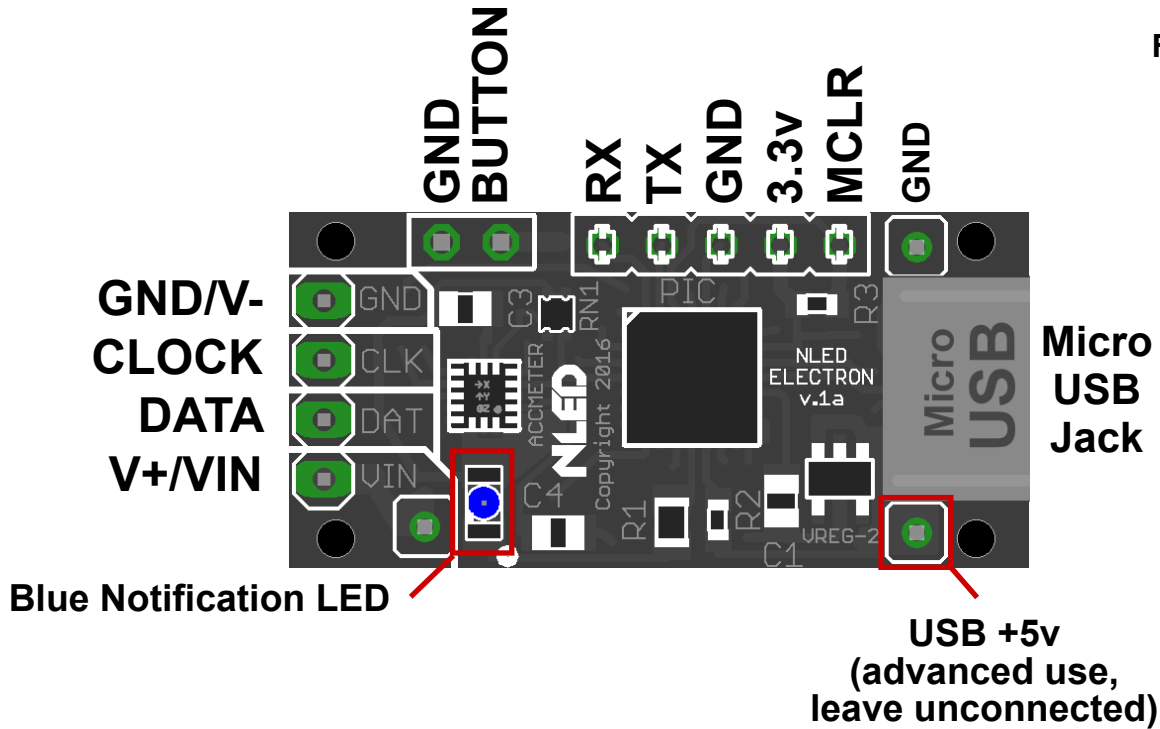
**Gamma Correction:** Enables or disables the use of the internal gamma correction table. See the Aurora Manual for information on this feature or search online for Gamma Correction for an overview.

## **Device Configurations Settings(Cont.)**

**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.

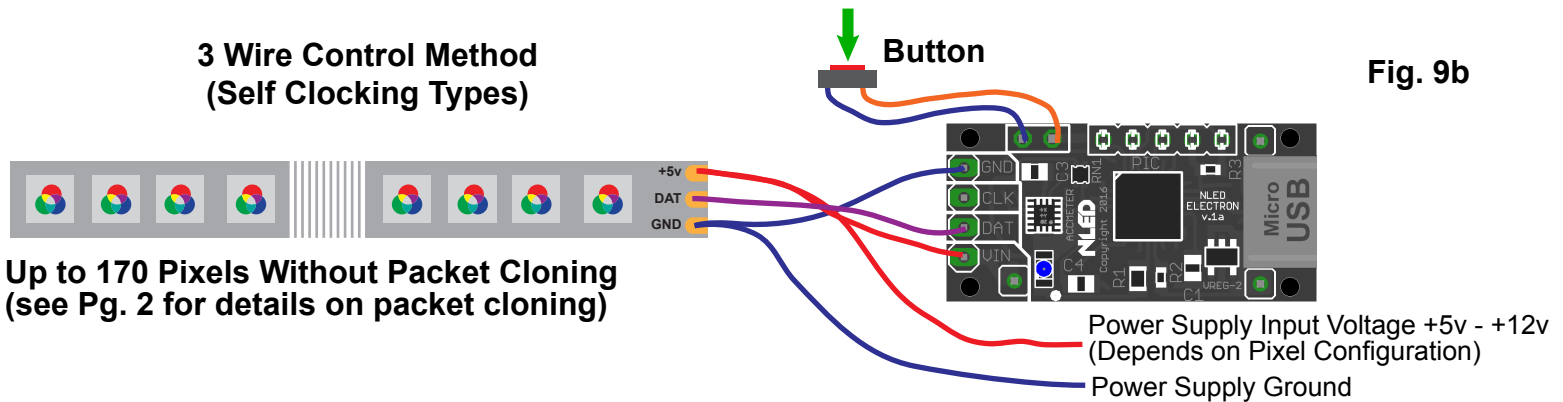
**Enable I.R. Remote Control(Addon Card):** Enables the device to be able to interface with the NEC Infrared Decoder and Encoder Board. The addon board converts infrared remote signals into serial commands that are received by the controller. The addon boards are hardware set to either use 19200 baud or 250,000 baud. To enable usage you must select the correct baud value for your addon card.

## Controller Pinout

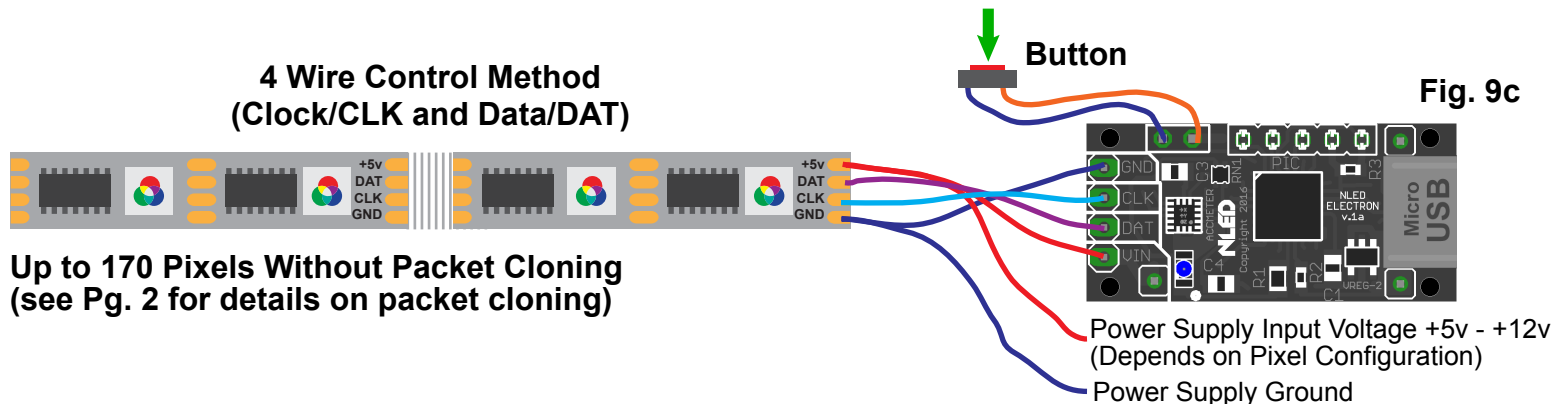


## LED Pixel Connections

**3 Wire Control Method (Self Clocking Types)**



**4 Wire Control Method (Clock/CLK and Data/DAT)**



## Power Considerations

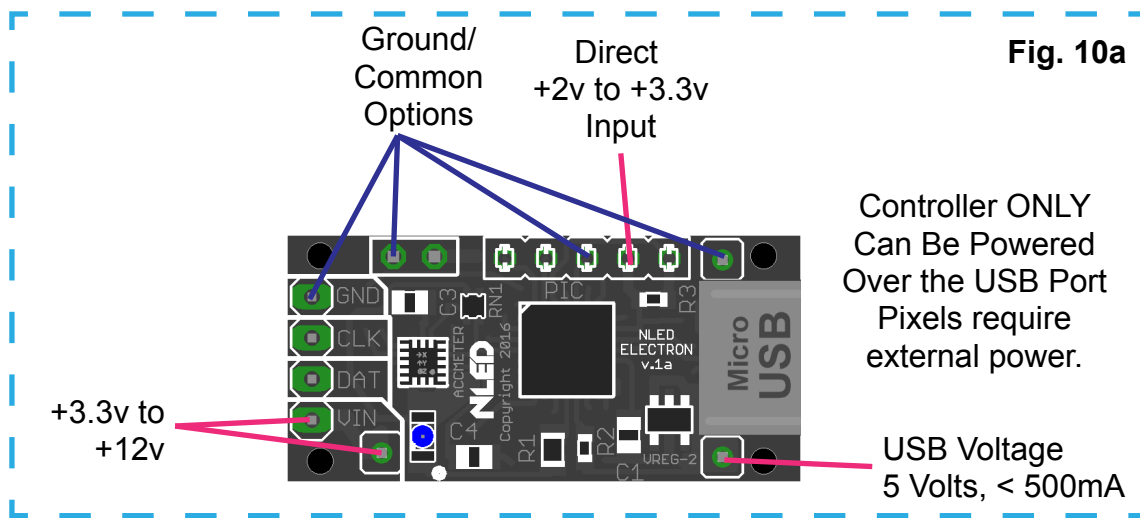
The controller accepts an input voltage of 3.3 volts to 12 volts. The controller regulates the input voltage to 3.3v to power the MCU and peripherals. When connected over USB only the controller will be powered, it can be interfaced through the software without an external power connection. Depending on the pixel configuration, some or all of the pixels may attempt to function or will light up, but will not act properly until they receive an external power connection of the proper voltage. 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. (Due to the controller receiving less than the minimum input voltage of 4 volts and browning-out or resetting) If a lower input voltage is required, the regulator can be bypassed and a direct input voltage of 2v to 3.3v could also be used by direct connection to the "3.3v" position. Voltages lower than 3.3v may not be compatible with the pixels.

The pixel amount and configuration must be considered when choosing a power supply. Ensure the power supply is of the proper voltage specified by the pixel manufacturer. Use the example formula to calculate the maximum amperage that would be required. This results with a the maximum value in amps that RGB(3 channels per pixel) pixels could draw if they were all turned on full intensity at the same time. Lower rated power supplies can and would work, but could result in problems such as unexpected controller resets or brown-outs, pixels acting strangely or not at all.

The required amperage rating can be estimated as follows:

Amount of Pixels x 60 = Required Rating in Milliamps, then add 10%, then divide by 1000 = Required Rating in Amps.

Ex: 170 WS2812B pixels, rated for 5 volts.  $170 \times 60\text{mA} = 10200\text{mA}$ ,  $10200 \times 1.1 = 11220\text{mA}$ ,  $11220 / 1000 = 11.22 \text{ mA}$



## Battery Powered Diagram

There are many ways to power NLED pixel controllers from batteries. The type, capacity and size of the batteries required depends on the current requirements of the project(see previous page). Pixels can end up drawing a lot of power, which can drain a battery fast. In addition to capacity requirements, batteries have a maximum discharge rate, smaller batteries can not provide enough current in some cases. Battery selection is a complicated topic, and it is recommended you research what is the best option for your project.

Common lithium ion sizes such as 14500("AA" sized) or 18650 are suitable in many cases. It is typically easier to design a project like a flow toy, staff, or wearable with removable batteries. That allows longer run times between charges since multiple batteries can be carried and it simplifies wiring. All lithium ion batteries require under-voltage and over-voltage protection, ensure you use a 'protected' lithium cell or install other protection methods.

Few useful parts:

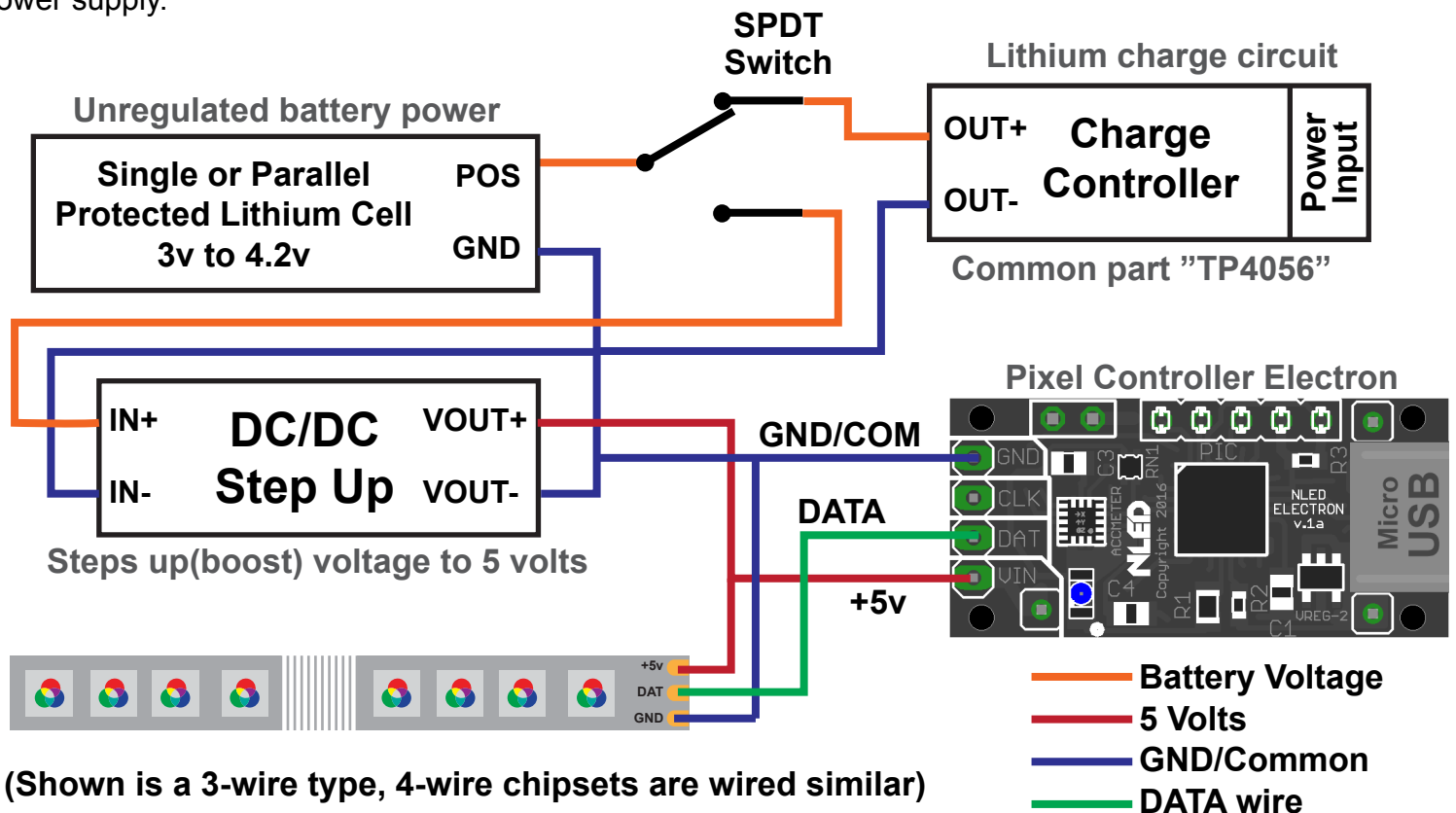
TP4056 USB Charge Modules - lower charge current, but cheap and easy to use.

SPDT Switch - comes in all sizes and shapes. Dig through catalogs or parts bins.

18650 Battery Holders: BH-18650-PC, BH-18650-W

The schematic below illustrates the simplest way to integrate a pixel controller into a battery powered project with self-charging and load disconnect.

The SPDT switch selects between powering the pixels and charge mode. This is a simple approach that doesn't require more complicated load sharing. Load sharing is when you have a device charging that could also be powered on, thus preventing charging and possible drawing too much current from the charging power supply.



## Connector Options

**Bare/Solder Pad/Unpopulated:** The output positions(VIN, CLK, DAT, GND) are left unpopulated, user solders on their own connector, such as a 3 or 4 pin JST-SM wire harness.

**Right Angle Header:** A 4-pin Molex right angle polarized header, DIP spaced, is soldered onto the output position. Note this option makes the longest dimension 50mm(with housing mated to header)

**Screw Down Terminal:** A 4-pin screw down terminal is soldered onto the output position. Very small screws, requires eye glass size flat/slotted point screw driver. This makes it 9mm tall.

See device store listing for details and photos of the different options.

## Dimensions and Mounting

This controller was designed to fit easily into 5/8"(16mm) tubes. And is approximately 0.16" (4mm) thick, leaving plenty of room for addon modules, wires, and other components.

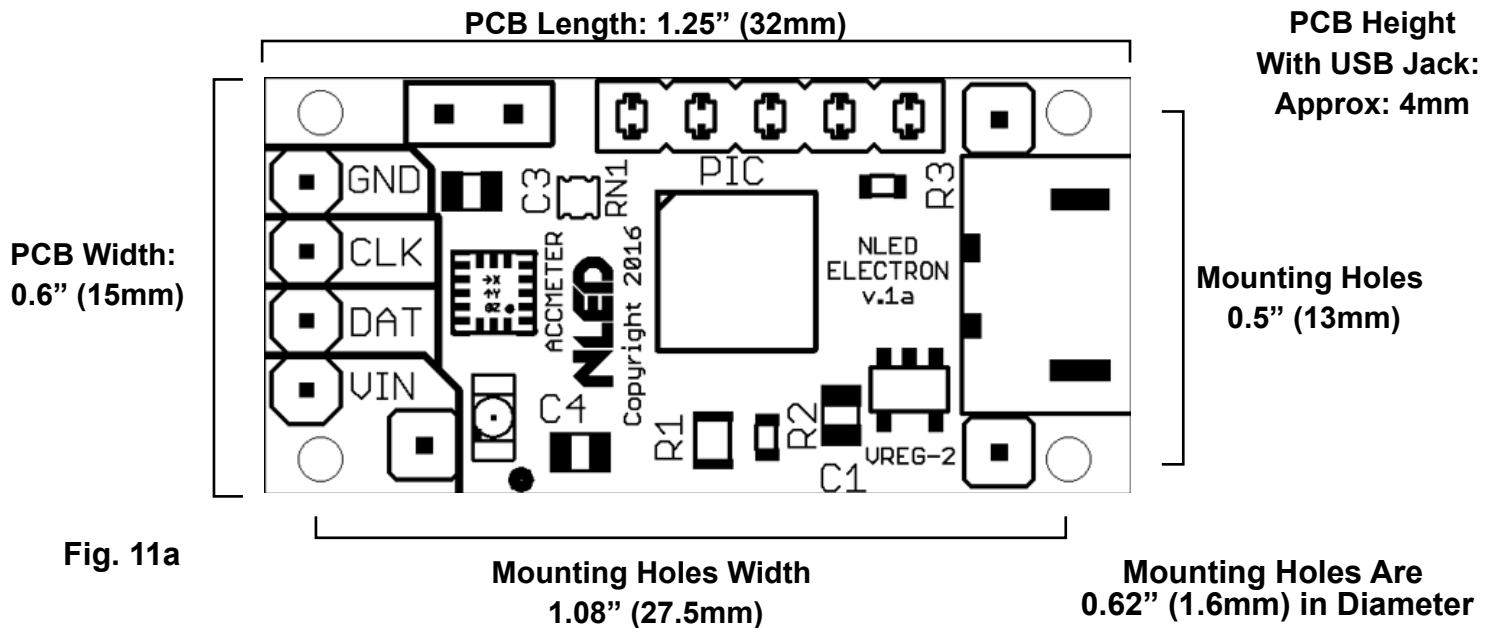


Fig. 11a

### 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.

## Usage Instructions

**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).
4. Start the software, navigate to the hardware tab, on the left side select the controller's COM port.
5. The device should identify itself, and the Connect button should turn from red to green. The hardware configuration setting controls will also be displayed.
6. Adjust the configuration setting controls on the Hardware Tab to suit your project see page 8 for details.
7. 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.
8. Connect your LED pixels of choice as described on page 9.
9. Apply suitable power to the controller and pixels, see page 10. 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

Please Contact [Support@NLEDshop.com](mailto:Support@NLEDshop.com) with any Questions, Comments, or Bug Reports.

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.

## 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](mailto: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.