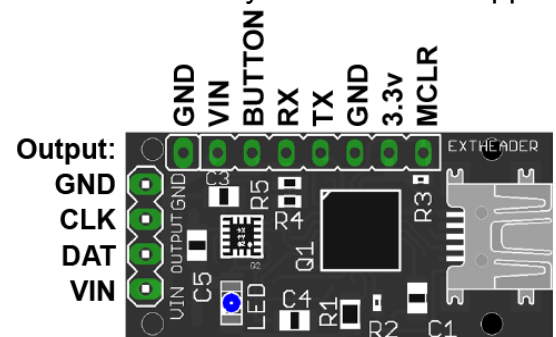## NLED Pixel Controller Ion

This controller from NLED packs many features into a small form factor. It supports various communication methods for direct control of the outputs, such as USB, TTL serial, or DMX-512. But the focus of the design is the powerful stand-alone color sequence support. The software NLED Aurora Control, allows a user to interface with the controller and create custom color sequences, including custom POVs, that can be uploaded to the controller over USB. The controller supports up to 512 channels(170 RGB pixels) and will run the loaded color sequences without any data or computer connection. The small size, low current, 3-axis accelerometer, and dynamic stand-alone color sequences make this controller a good choice for wearables, hoops, staves, poi, bikes, and other controllable pixel based projects. The device currently supports several of the most popular pixel chipsets, with the ability to easily update the firmware with new features and chipset support.

## Features:

- Stand-Alone Supports up to 512 channels or single color pixels,170 RGB Pixels, 128 RGBW.
- Supports Multiple Pixel Chipsets - WS2801, WS2811, WS2812, WS2812B, LPD8806, LPD6803, LPD1886, TM1803, APA102, APA104, SK6812, and more
- USB Connectivity - Control thousands of pixels over USB(Using the optional Mass Control firmware)
- 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., Persistence 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 29Kbyte 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 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 the device 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.

| | |
|---|---|
| **Input Voltage** | 3.3 volts to 12 volts |
| **Logic Current Draw** | > 50mA |
| **Standard Outputs** | 512 Channels(170 RGB pixels) |
| **Extended Outputs** | 3072 Channels |
| **Connectors** | Multiple Options*, 0.1" spacing |
| **USB Connector** | Micro USB |
| **Data Memory** | 21KB |
| **Dimensions** | 1.25" x 0.6" x 0.16" |



**Fig. 1a**
**See Wiring and Pinout Section**
**For Detailed Layout Information**

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

## Supported Pixel Chipsets

**The full list of tested and compatible chipsets is now availabe online at:**
**www.nledshop.com/pixelchipsets/**

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. 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 taken from the datasheet 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 NLED Aurora Control software and NLED Configuration Software offer a Device Configuration setting that is used to select your particular chipset.

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 is are separate firmware versions(Mass Pixel Control) available that allows 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 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.

**Extended Pixel Output:**

This controller can support more pixels up to 3072 channels, which is 1024 pixels in a single strand. But requires a different firmware than the stock Aurora firmware. Called 'Aurora Matrix' this firmware does not support all the sequence modes(Fade,Gradient,Instant, are not available) and only supports File Play and POV sequence modes(as of this release). This allows a lot more data to be stored internally for playback of large sequences in stand-alone mode and allows for control of a lot more pixels than is available with the standard Aurora firmware.

This feature is new and a bit complicated, details can be found at: **www.nledshop.com/nledauroramatrix/**

Allows use of software such as NLED AllPixMatrix and the third-party Glediator software.
**www.nledshop.com/nledallpixmatrix/**

## 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 protoocl 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)
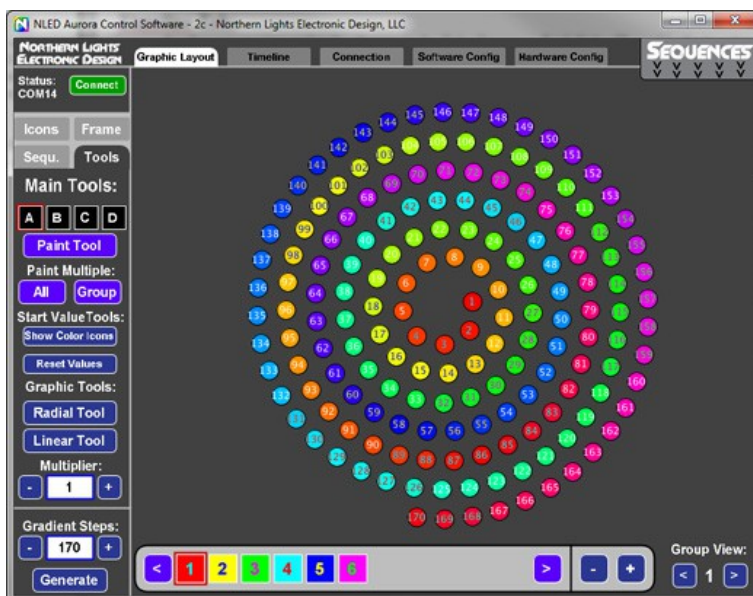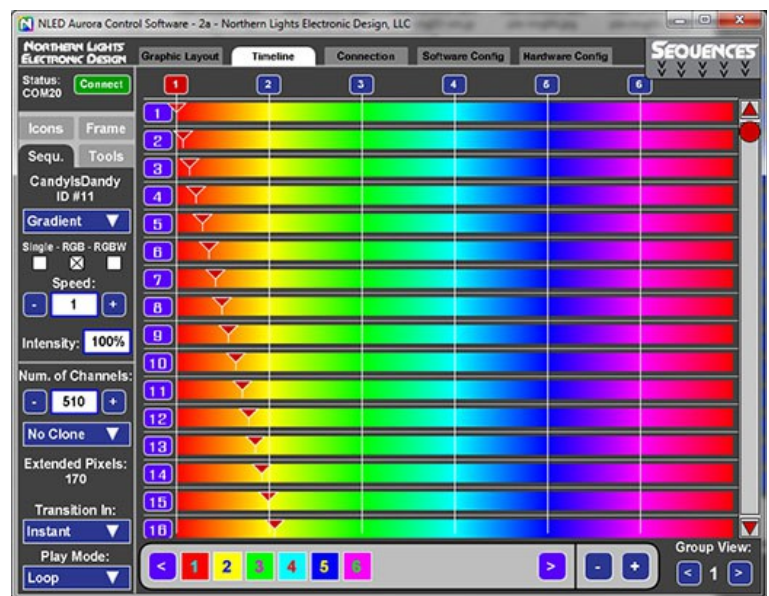


Fig. 3a



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.

This feature can be used for easy and fast direct control of the outputs(pixels or otherwise), similar to how DMX would be used but without any additional hardware or special software.

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.

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 controller's 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.

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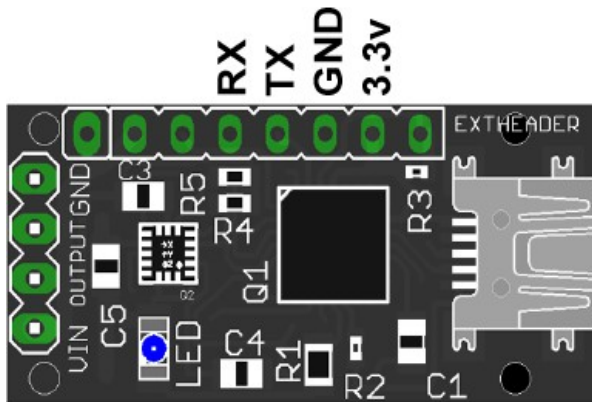
| ID# | Actual Device Baud | Closest Common Baud Rate | Error % |
|-----|-----|-----|-----|
| 0 | 9,615 | 9,600 | 0.16% |
| 1 | 19,230 | 19,200 | 0.16% |
| 2 | 38,461 | 38,400 | 0.16% |
| 3 | 57,142 | 57,600 | 0.64% |
| 4 | 117,647 | 115,200 | -0.79% |
| 5 | 222,222 | 230,400 | 2.12%* |
| 6 | 250,000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

Fig. 5a                     8-N-1

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

**Expansion Header Connection**

RX ⟵ TX
TX ⟶ RX
GND ⟷ GND
3.3V ⟷ Optional

Fig. 5a                                         Fig. 5b

Device supports both 3.3 volt and 5 volt TTL serial signals. No level translation(shifting) is required.

**More details and information can be found at**
http://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:**
1. With the device powered off, while holding down the button(s) or after making the jumper connection. Device specific details are found in the Wiring section on page x.
2. Power up the device. Either by connecting the device to the computer with a USB cable. Or by connecting external power.
3. Wait a second or two.
4. Device will then enter bootloader mode. Follow the instructions on www.NLEDshop.com/bootloader

**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.
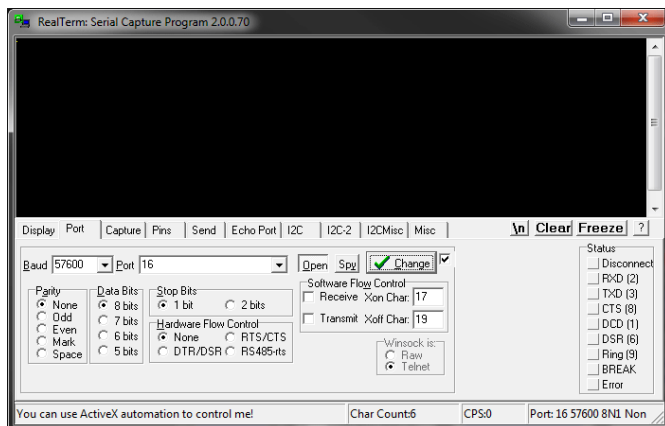8. Device will then enter bootloader mode. Follow the instructions on **www.NLEDshop.com/bootloader**



Fig. 6a



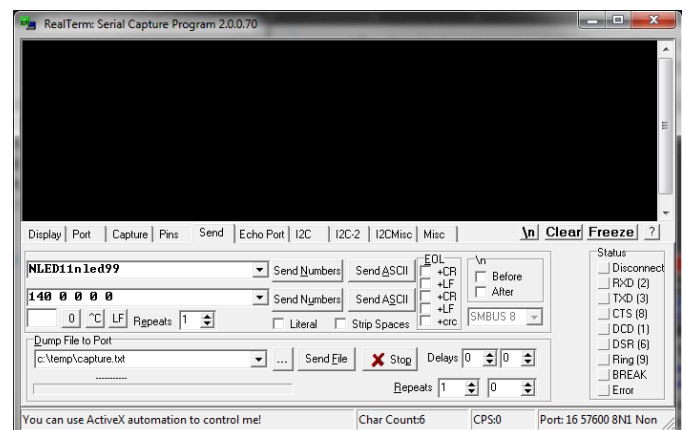Fig. 6b

# 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. The feature basically only sends and receives serial data without AT character exchanges or other protocols. So no different than using an FTDI or other USB to serial adapter, but wireless. 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 webserver.

**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, reseting, 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 2 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 outptus. The baud rate and Auto-Detect option should also be set for this to work correctly.

    **Infrared Addon Card, 19200:** Allows the NEC Infrared Decoder and Encoder addon card to be utilized for remote control of the device. The addon card must also be setup for 19200 baud rate.

    **Infrared Addon Card, 250K:** Allows the NEC Infrared Decoder and Encoder addon card to be utilized for remote control of the device. The addon card must also be setup for 250,000 baud rate.

**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 pg. 4)

**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 plaing 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 Options:** 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.

**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.

**Accelerometer Movement Detection:** 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 or 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.

## 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 movement/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.

## DMX-512 Usage

The device supports multiple DMX-512 reception modes, a reception mode can be selected through the configurations. 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.

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. 10a

| 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. 10b

| 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: (Position / 255) x 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. 10c

| 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% |

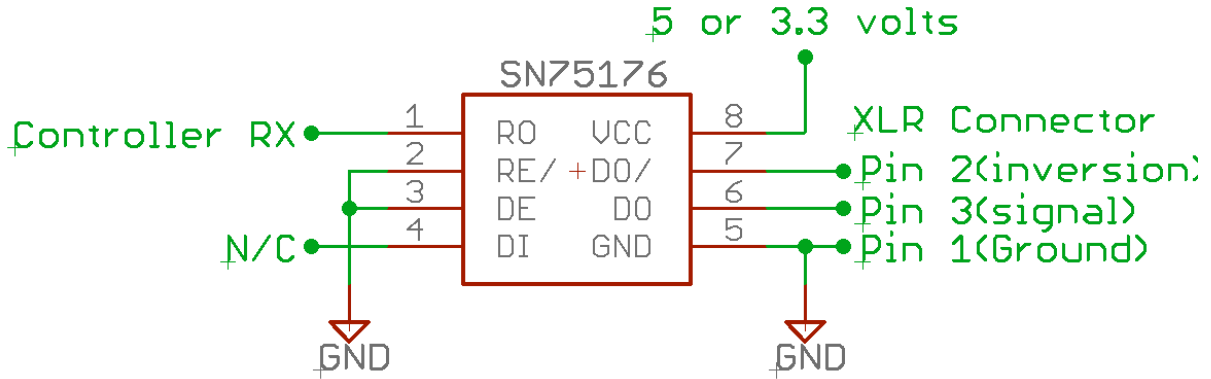**RGBW Compatibility:** All pixels on the strand are set to the same color, not individually.     Fig. 10d

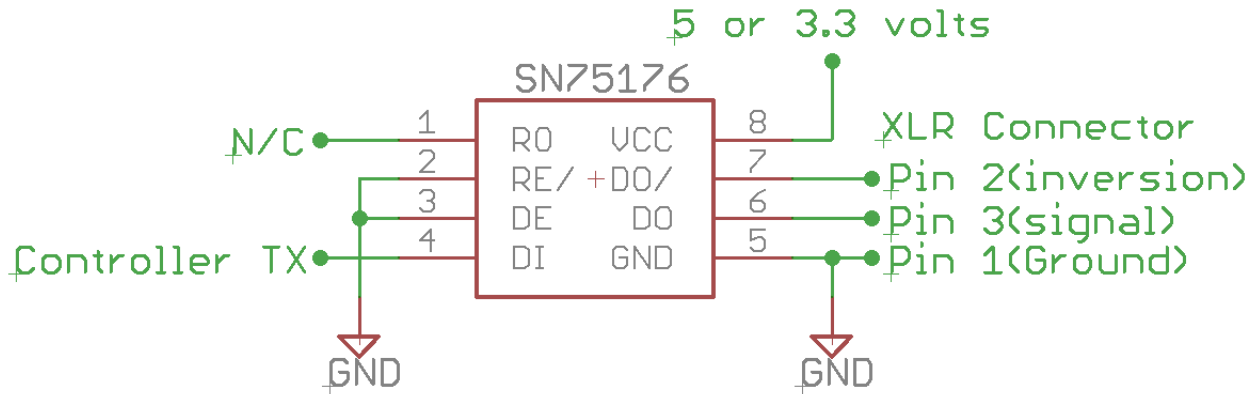| Channel | Function | Value | Description |
|---------|----------|-------|-------------|
| 1 | Red Value | 0 - 255 | 0 is 0%, 255 is 100% |
| 2 | Green Value | 0 - 255 | 0 is 0%, 255 is 100% |
| 3 | Blue Value | 0 - 255 | 0 is 0%, 255 is 100% |
| 4 | White/Amber/UV Value | 0 - 255 | 0 is 0%, 255 is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

# 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:**



**DMX Master:**



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

**Serial Reception Mode:** See Serial Reception section.

**Infrared Addon Card, 19200:** The addon card is wired on normally as a serial device. See the Serial Reception section for details. Ensure 3.3 volt is also connected as the addon card requires power.

**Infrared Addon Card, 250K:** The addon card is wired on normally as a serial device. See the Serial Reception section for details. Ensure 3.3 volt is also connected as the addon card requires power.
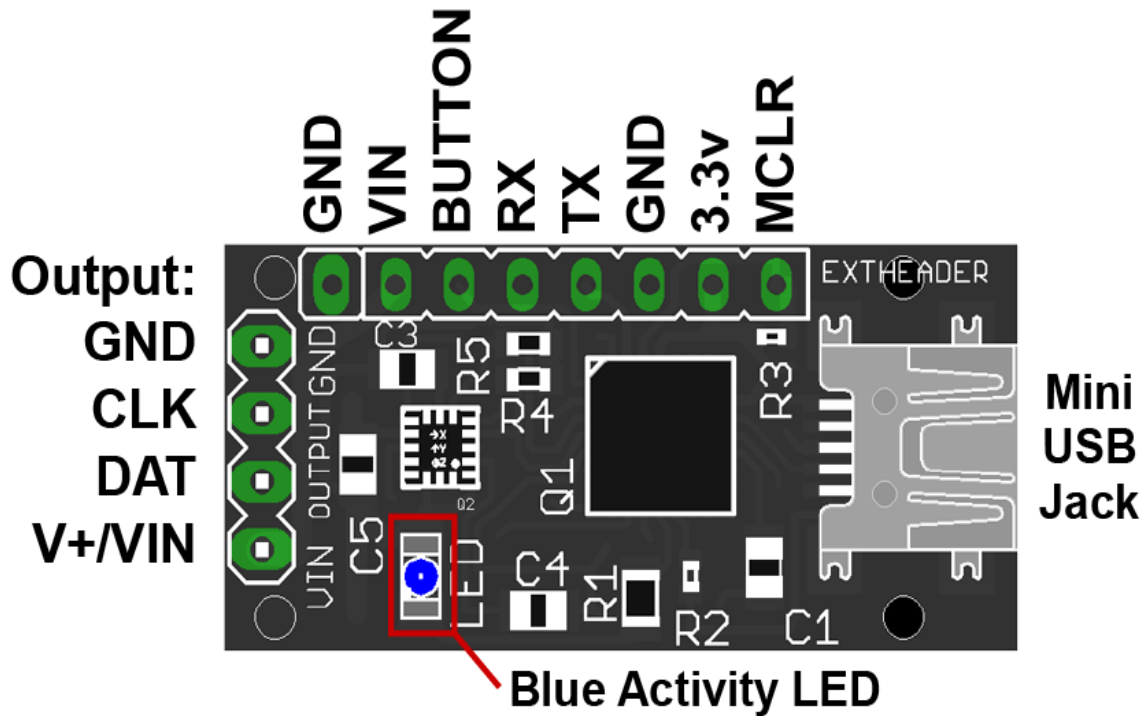
## Controller Pinout
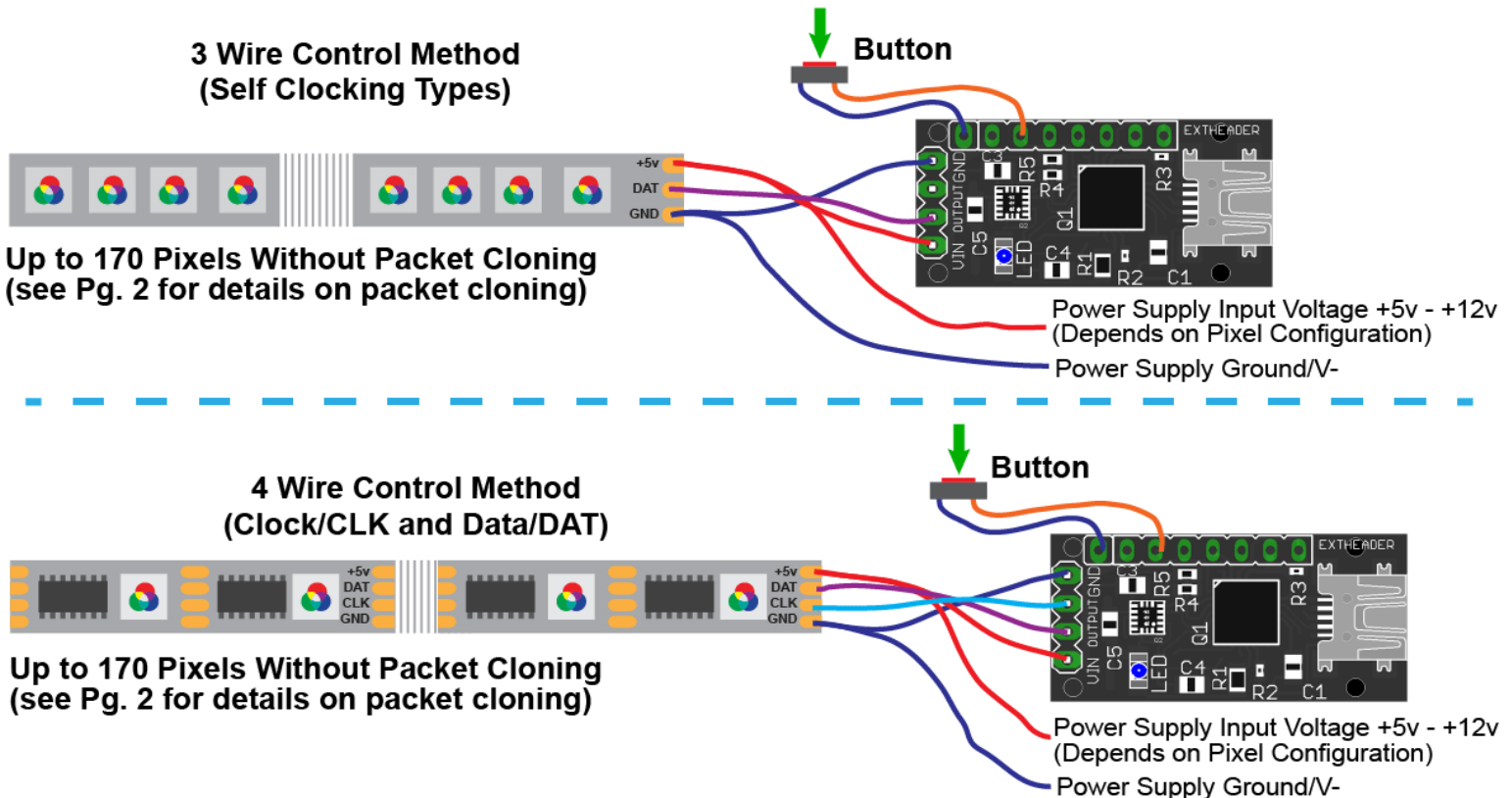


Fig. 12a

## LED Pixel Connections



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

Up to 170 Pixels Without Packet Cloning
(see Pg. 2 for details on packet cloning)

Power Supply Input Voltage +5v - +12v
(Depends on Pixel Configuration)
Power Supply Ground/V-

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

Up to 170 Pixels Without Packet Cloning
(see Pg. 2 for details on packet cloning)

Power Supply Input Voltage +5v - +12v
(Depends on Pixel Configuration)
Power Supply Ground/V-

Fig. 12b

# 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.

More info and expanded details can be found at: **www.nledshop.com/projectguide/powersupplies/**

The required amperage rating can be estimated as follows:

An example with WS2812B pixels, which can draw up to 50mA each.
Ex: 170 WS2812B pixels, rated for 5 volts. 170 x 50mA = 8500mA, or 8.5 Amps Maximum.
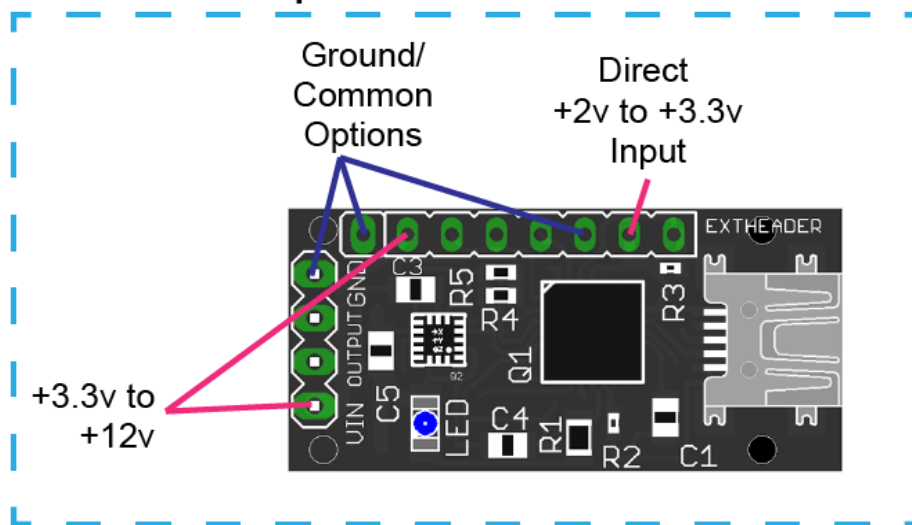
**External Power Option:**



Fig. 13a

# Battery Powered Diagram and Information

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.
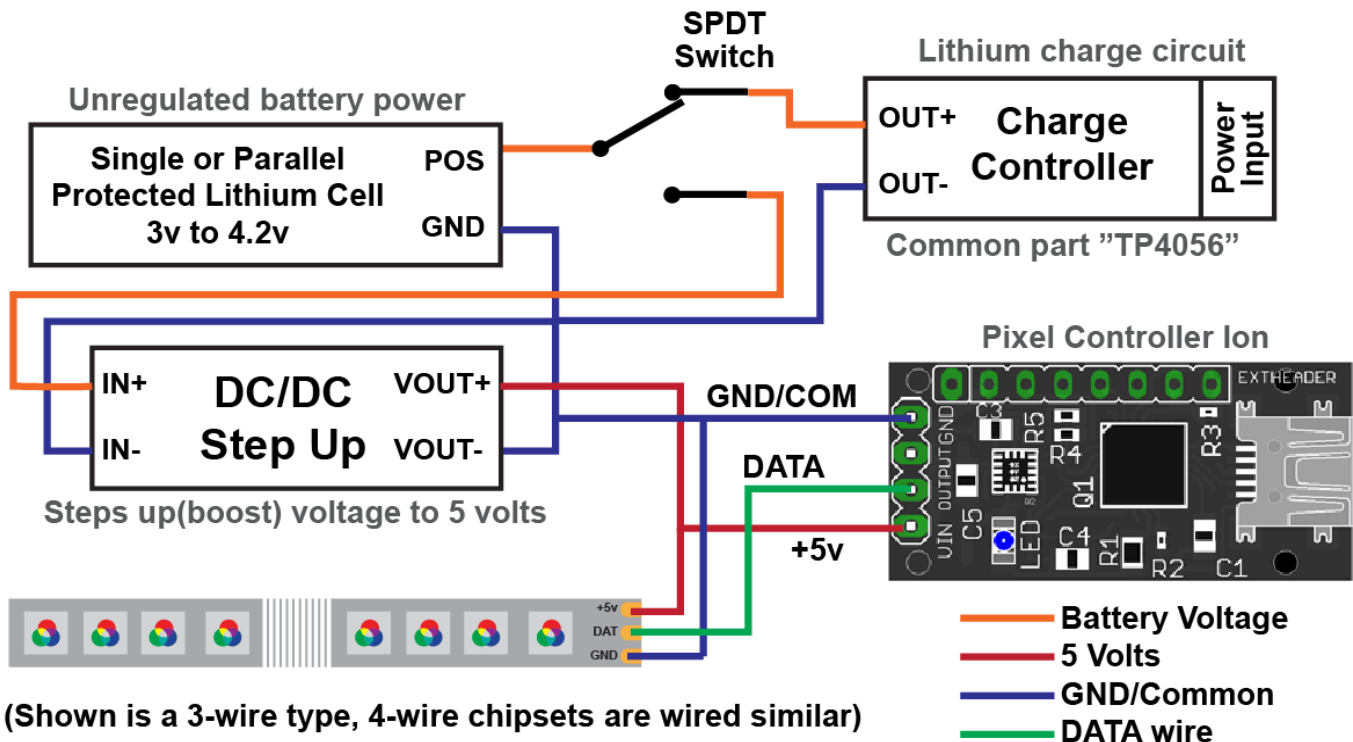
Few commonly available parts:
- TP4056 USB Charge Modules - lower charge current, but cheap and easy to use.
- SPDT Switch - comes in all sizes and shapes. Dig through catalogs or parts bins.
- 18650 Battery Holders: BH-18650-PC, BH-18650-W

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.

More information and details can be found online at: **www.nledshop.com/projectguide/batteries/**
Fig. 14a



**(Shown is a 3-wire type, 4-wire chipsets are wired similar)**

# 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.
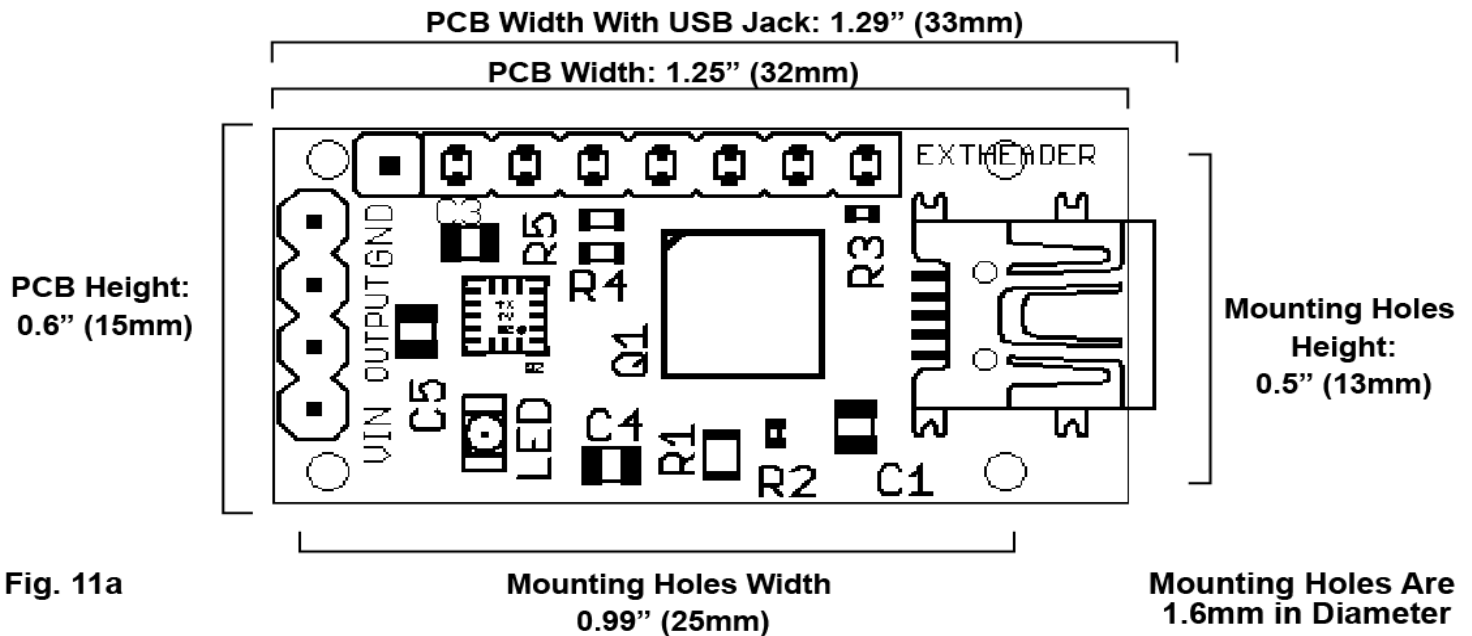
Fig. 15a



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

# 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 page 8 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 on page x.
8. Apply suitable power to the controller and pixels, see page x. 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.

## <u>Have Any Ideas For Future Updates?</u>

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.