What is Vixen Lights?

Vixen Lights is software for do-it-yourself lighting automation displays. The latest version 3.x was a complete redesign to support addressable “pixel” lights.

You can download it at http://www.vixenlights.com/downloads/ This tutorial is based on versión 3.4u2 64 Bit.

What is a Pixel?

A pixel is a cluster of 3 Light Emitting Diodes (LEDs) consisting of the three primary colors (Red, Green and Blue). The intensity of these three colors (LEDs) can be varied to make other colors. The WS2812B pixels I am using in my example include an Integrate Circuit (IC) chip which accepts data in on one port, displays the information that was addressed to it and passes on data to the next pixel. For my display I purchased strings of 5 meters which have 30 pixels for every meter or 150 pixels for 5 meters. Addressable RGB “pixel” strips can usually be identified because they have 3 wires. One for power, one for ground and one for data. In contrast, RGB “dumb” strips can be identified by them having 4 wires. Usually one for power and one for each color Red, Green and Blue

Tira Estandar RGB

Tira Inteligente Pixeles

RGB Pixels use a LOT of power. While the Arduino board can power up a few pixels using the built in voltage regulator, you will quickly run out of power. Therefore we will want to use an external power supply to power the lights. The voltage required will depend on the specific lights you have purchased. The lights I am using are 5V (volt). Another requirement of a power supply is to ensure it has enough power to support the number of pixels you are using. Each pixel at full white requires approximately 60 milliamps. for 150 RGB Pixels that works out to approximately 9A (Amps).

Unfortunately, the thin copper traces used on most LED strips causes “resistance” which will lead to dropping voltage levels. If the voltage at the pixel drops too low it can cause various problems such as flicker, dim lights or simply failure to light up. To avoid these issues, you may find that you will need to “insert” power at points within the RGB Pixel Strip. You would cut the strip and then just jump the DATA line across and add a new set of POWER and GROUND wires running back to your power supply. But be aware, that the distance of your power wire will also cause resistance and result in voltage drop. To avoid this, you must use wire that is thick enough based on your power requirements. The following chart is a decent starting point for choosing the right size of power

Bigger is always better when it comes to a power supply. You will need something that can provide more power (Amps) then you require. In my case I have ordered a couple of medium sized power supplies, 40 Amp and 60 Amp. Multiple power supplies can be used however, you should connect all of the ground wires for the power supplies together.

Arduino Controller

Many of the main stream Arduino boards can be used as a controller to become the middle man between the computer running Vixen Lights and the actual RGB Pixel Strips.

Various boards have different hardware limitation such as processor speed, memory (RAM) size and storage size. However in testing, the biggest limiting factor we found was the speed of the Serial Port. Most Arduinos can not go any faster than 115,200 bps. When we push the color codes for each of the three colors for 150 Pixels (aka 450 colors) down the serial port at 115,200 bps we can calculate that it will take 45 milliseconds to complete the transmission. This means we can safely refresh each pixel every 50 milliseconds (or 20 times per second).

Arduino UNO

Arduino MEGA

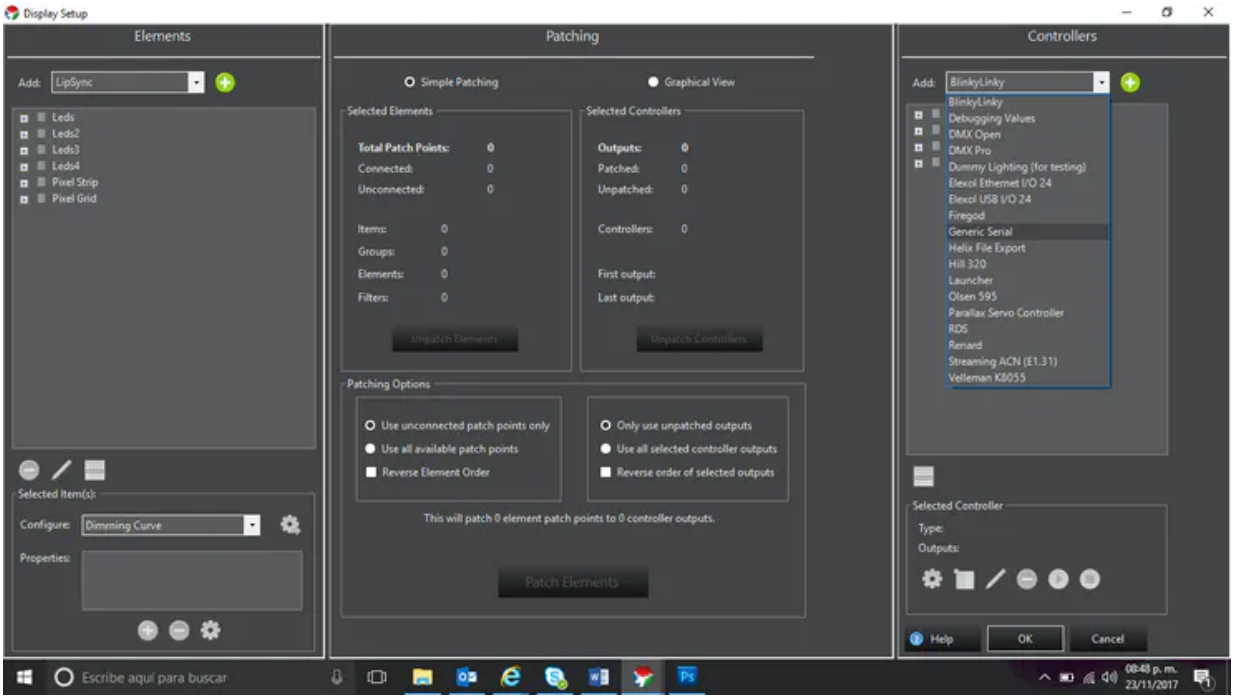
Arduino NANO

The Code on this Tutorial is base in David Hunt - blog.huntgang.com

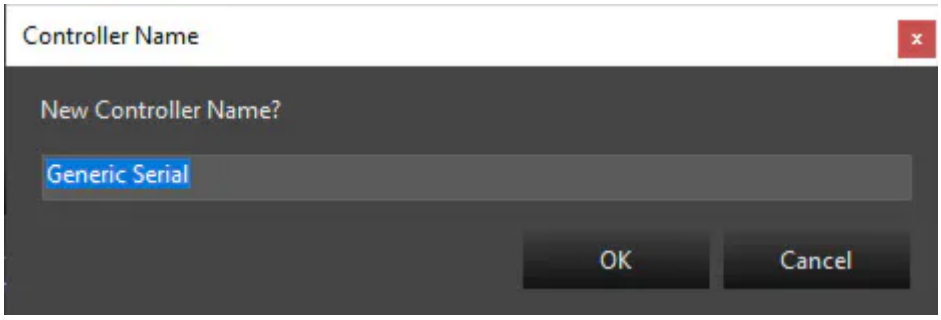
Configuring Vixen Serial Port

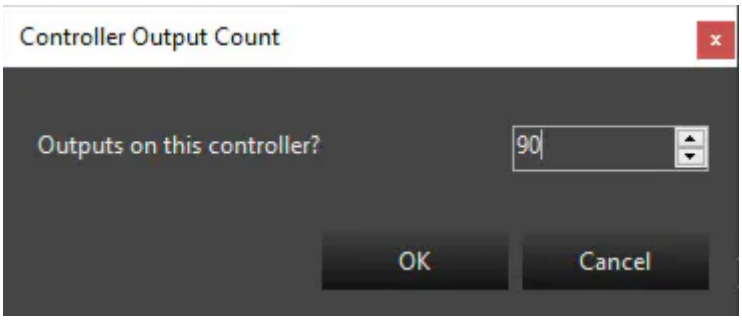
In order to use the Arduino controller, you must configure it inside of Vixen 3.x The following process was documented using

Step 1.- Add a Generic Serial Controller for the top right Menu.

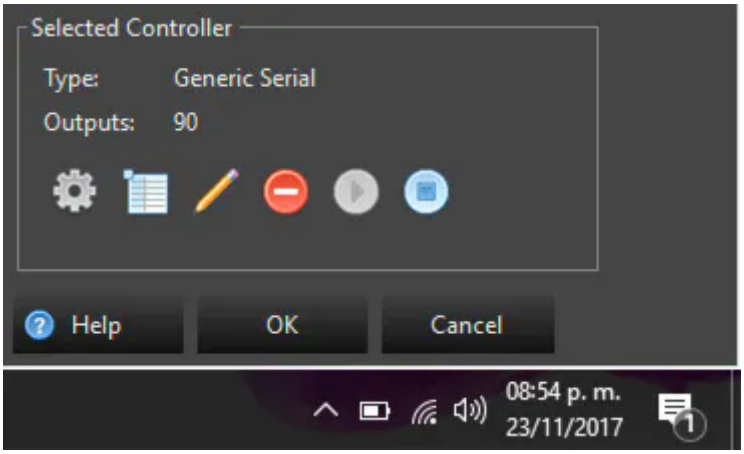


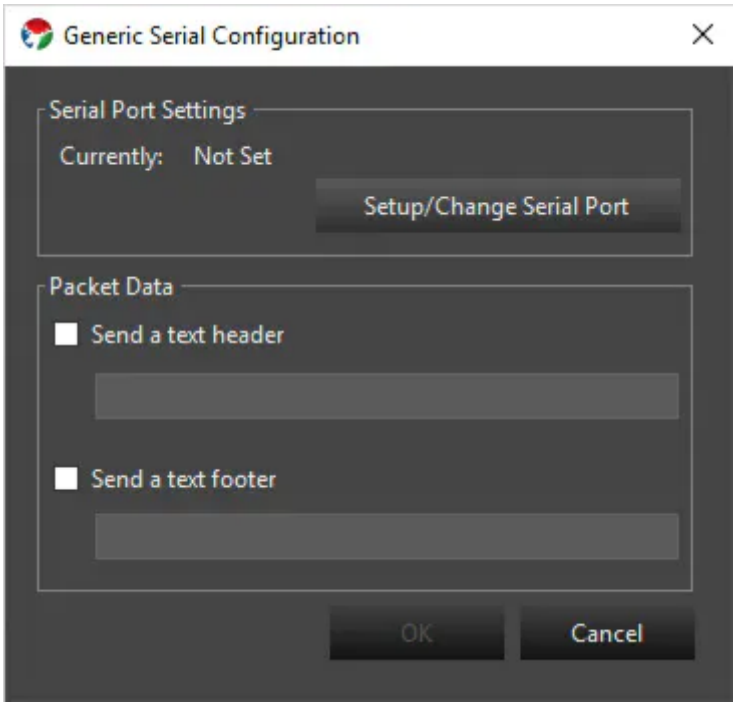
Step 2.- Set the number of outputs for the controller. This number should be 3x the number of pixels. In my example I am configuring 30 pixels which means I will set the output count to 90.

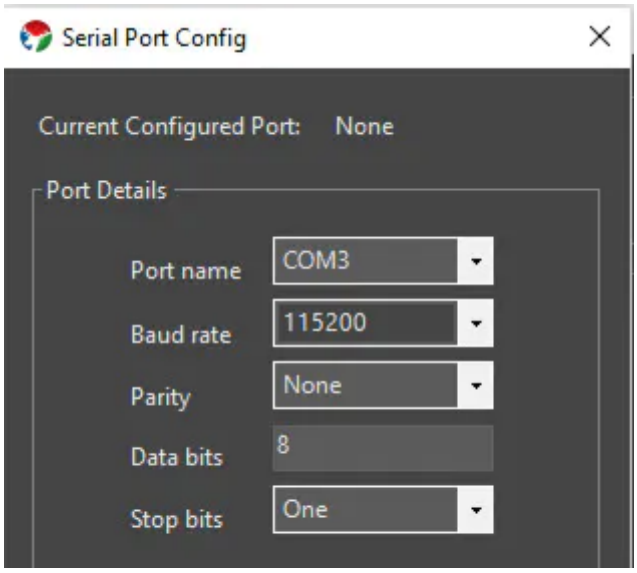


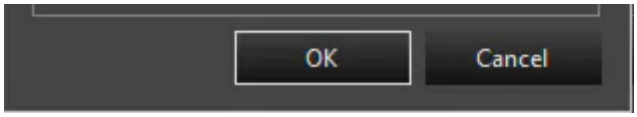


Step 3.- In the low right corner click the Gera Icon, now we will configure the COM Port. To do this we will select the COM port for the Arduino. My example is COM13 but yours will probably be different. We also want to configure the baud rate to 115200. The remainder of the settings can be left alone.





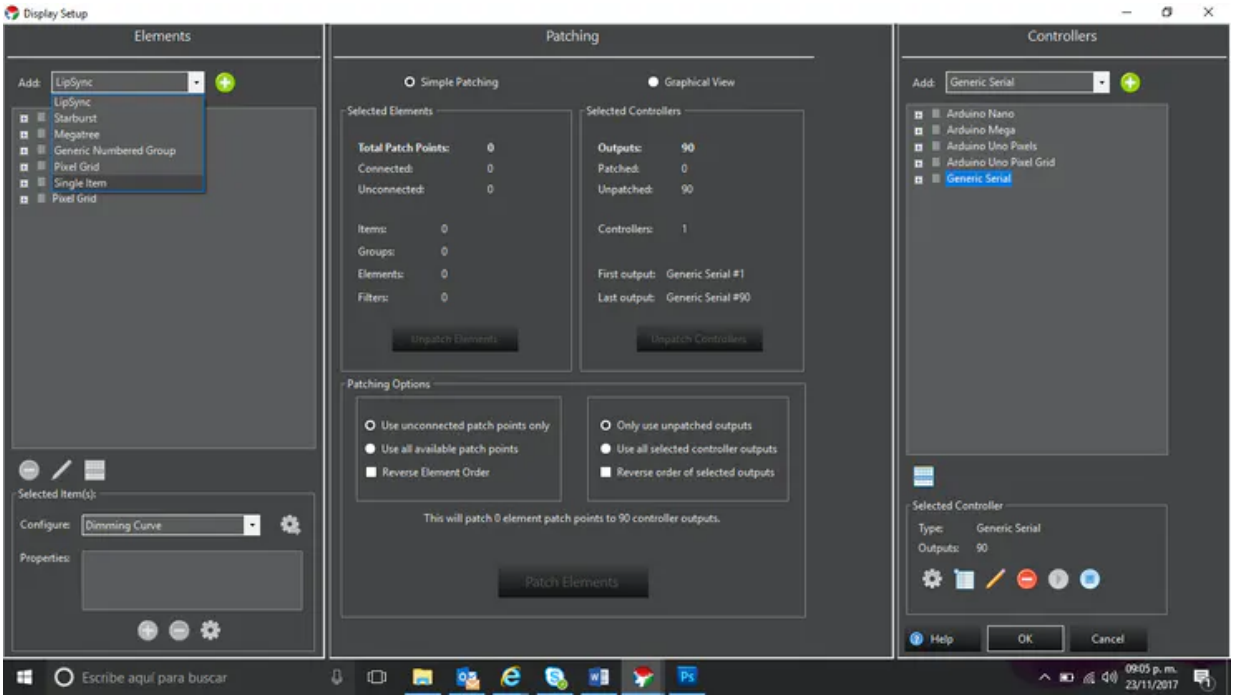




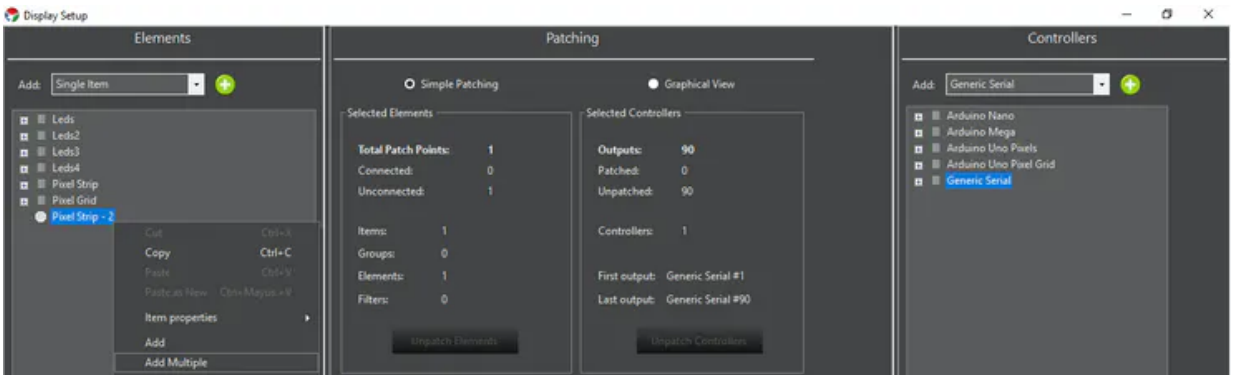
Step 4.- Here we will be adding the number of pixels into the header so that the Arduino knows how many pixels it should be receiving. The number of pixels must be 300 or less and must be entered as a three digit value. Again my example uses 030 pixels therefore I will be preceding it with two zeros. At this point you should see a bunch of blinky flashy on your Arduino as it is now receiving the serial data.

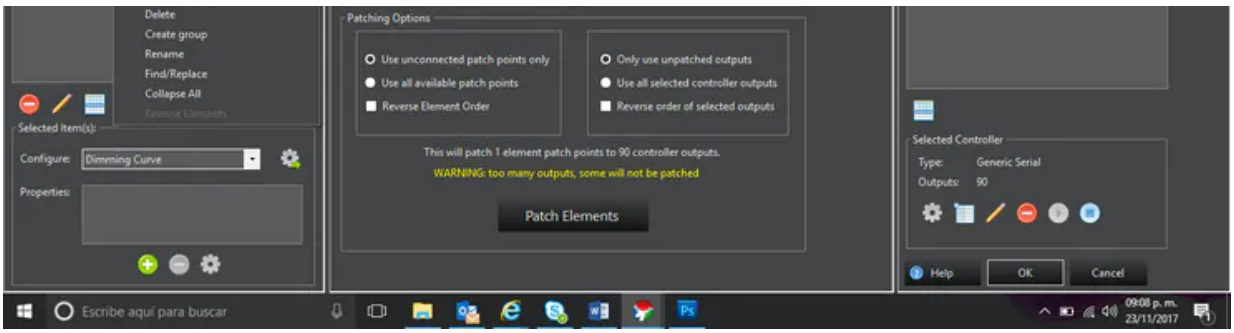
Configure Element for the Pixels

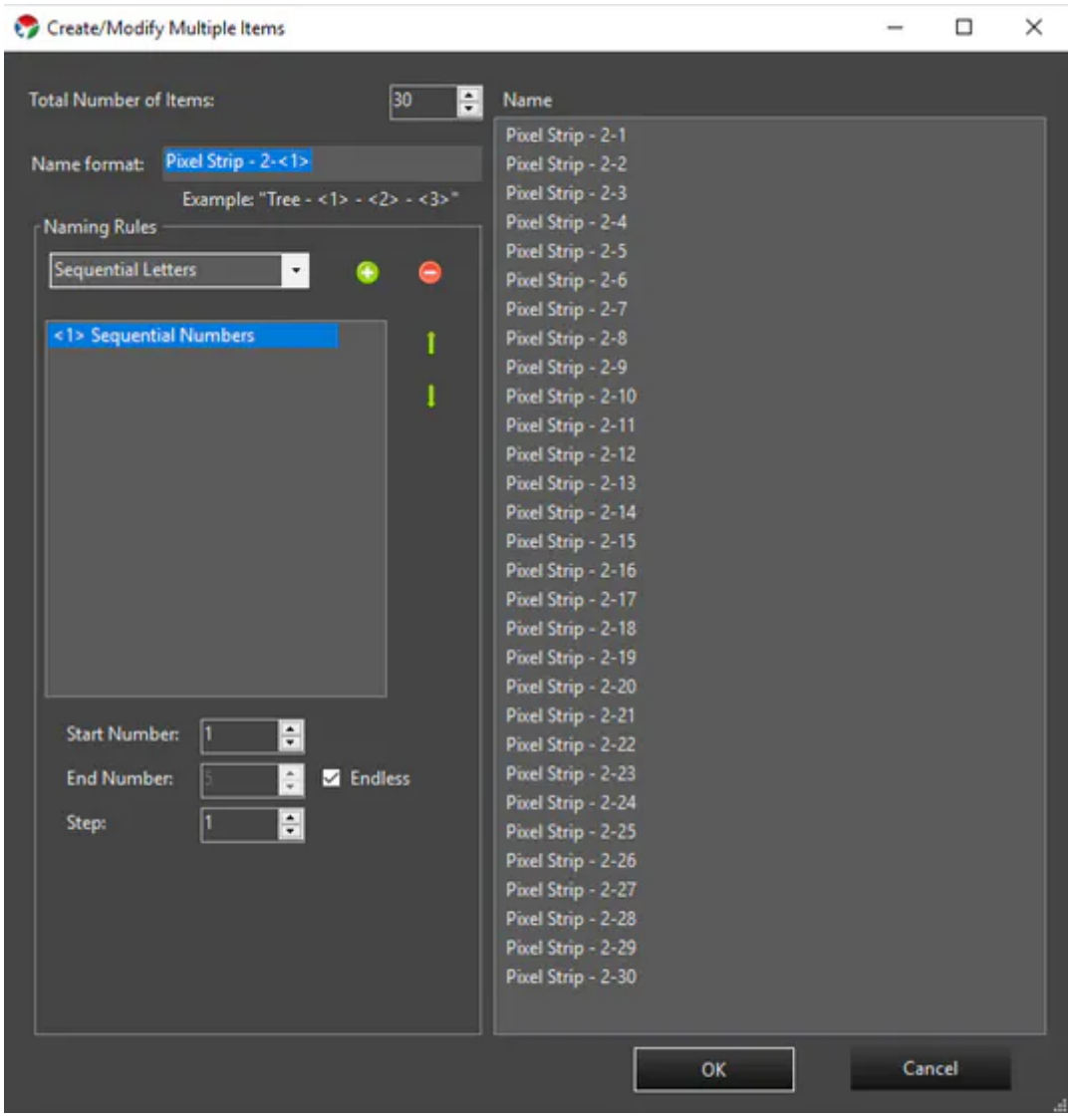
Step 5.- In the top left you will see a drop box, select Single Item, click on the Add Green button, and name it Pixel Strip.



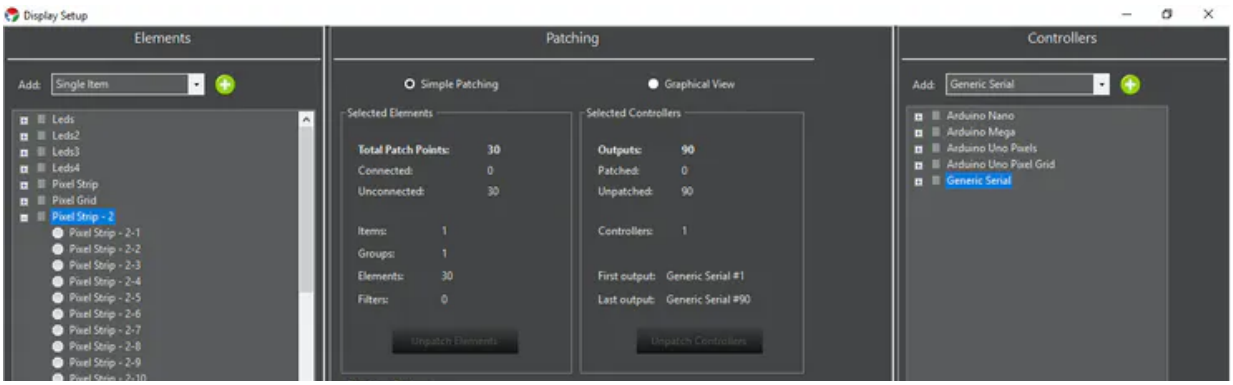
Step 6.- Next we will right click on the Pixel Strip we just created and we will select Add Multiple. To add all of the pixels, we will select Numbered Items, define a name (I used Pixel Strip) and then select the number of pixels to generate (30 in my example). You should see all of the names in the list before clicking OK.

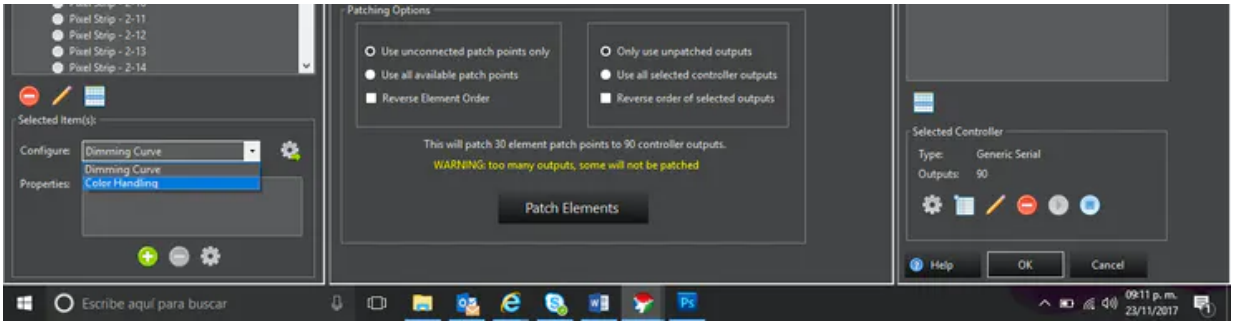


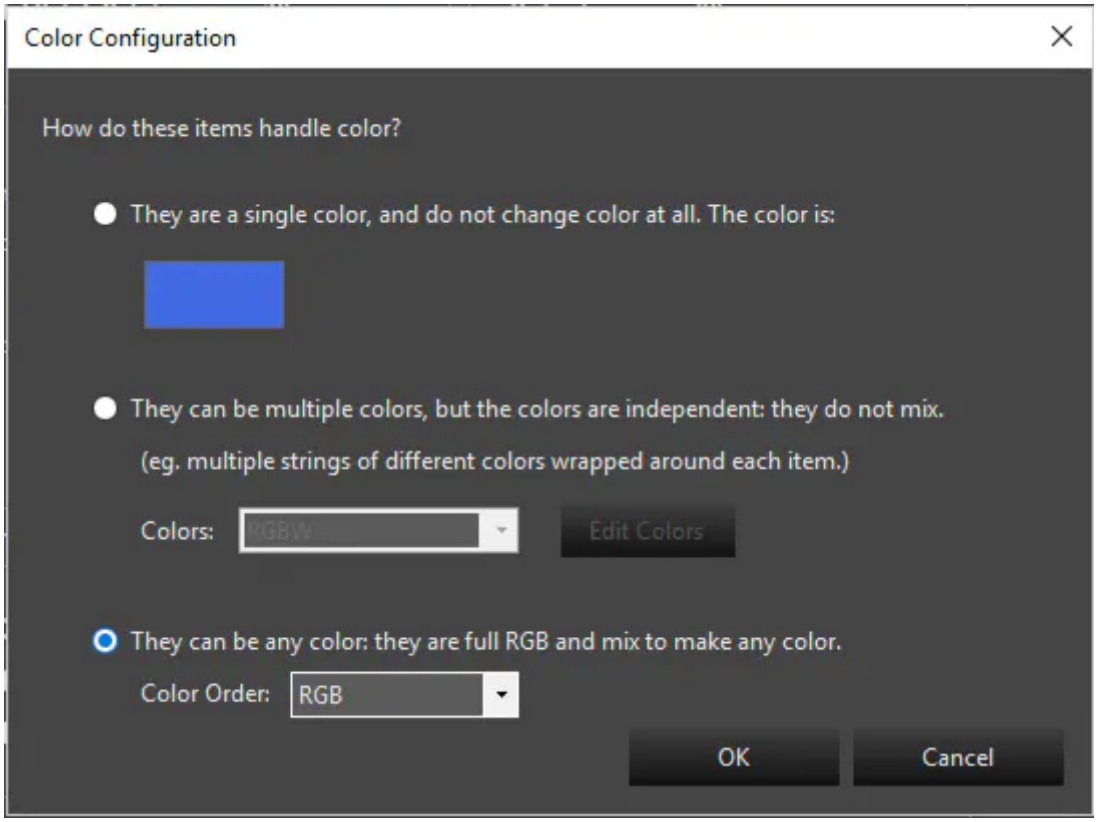




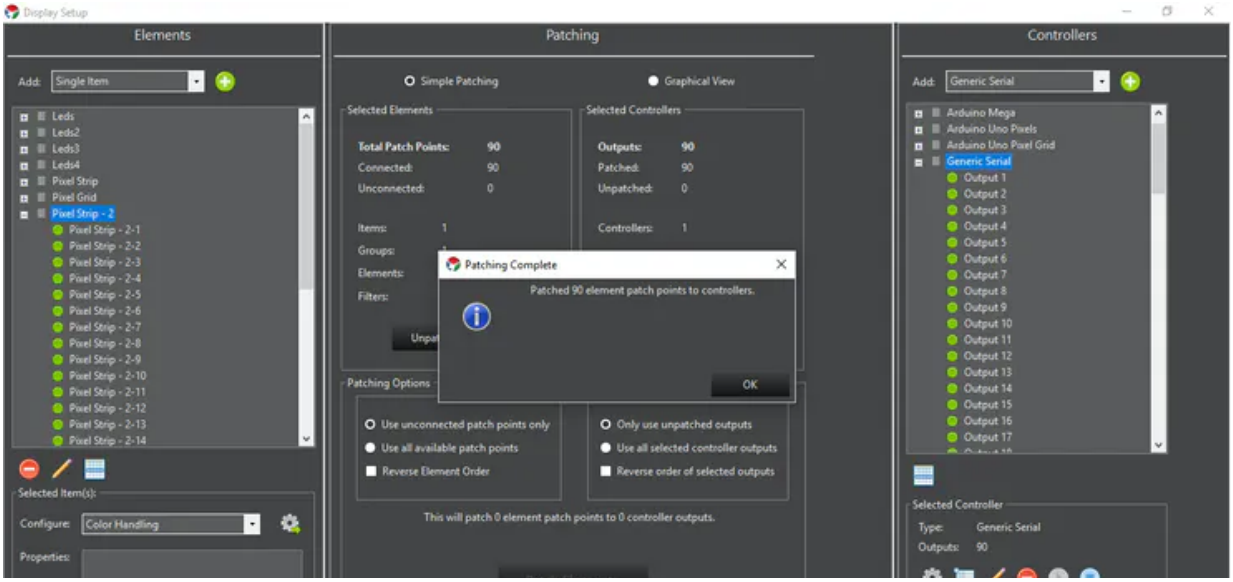
Step 7.- Now we will highlight the Pixel Strip and Configure the Color Handling property. We will select “They can be any color: they are full RGB and mix to make any color.”

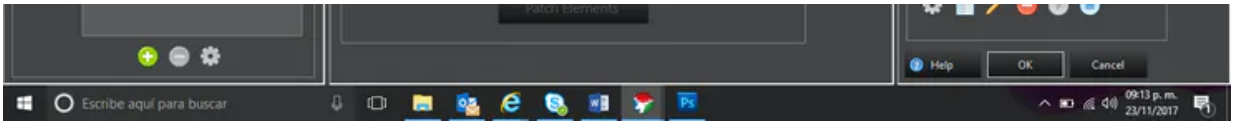




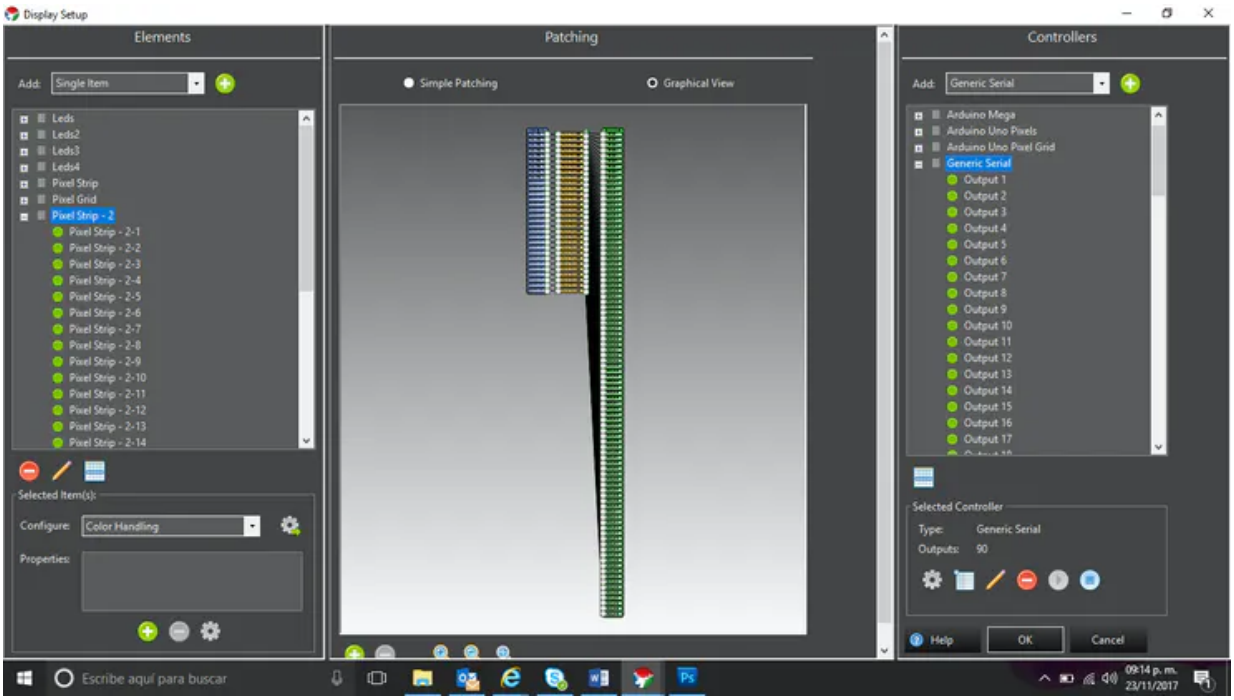


Step 8.- The final step before we can call it a day is to patch the Element to the Controller. To do this highlight the Pixel Strip on the left and the Generic Serial controller on the right. The number of Unconnected Patch Points should match. The only thing left to do is click Patch Elements to Controllers and then you are ready for Christmas Light.





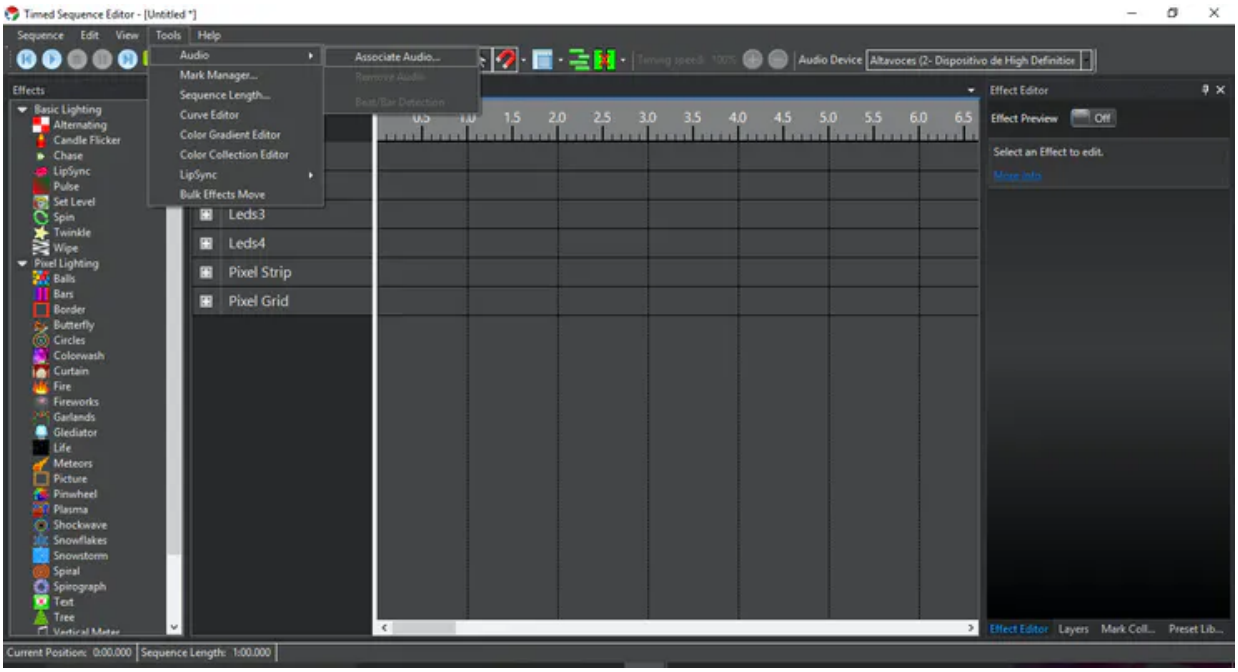
Step 9.- If you were successful your graphical view should look something like this.



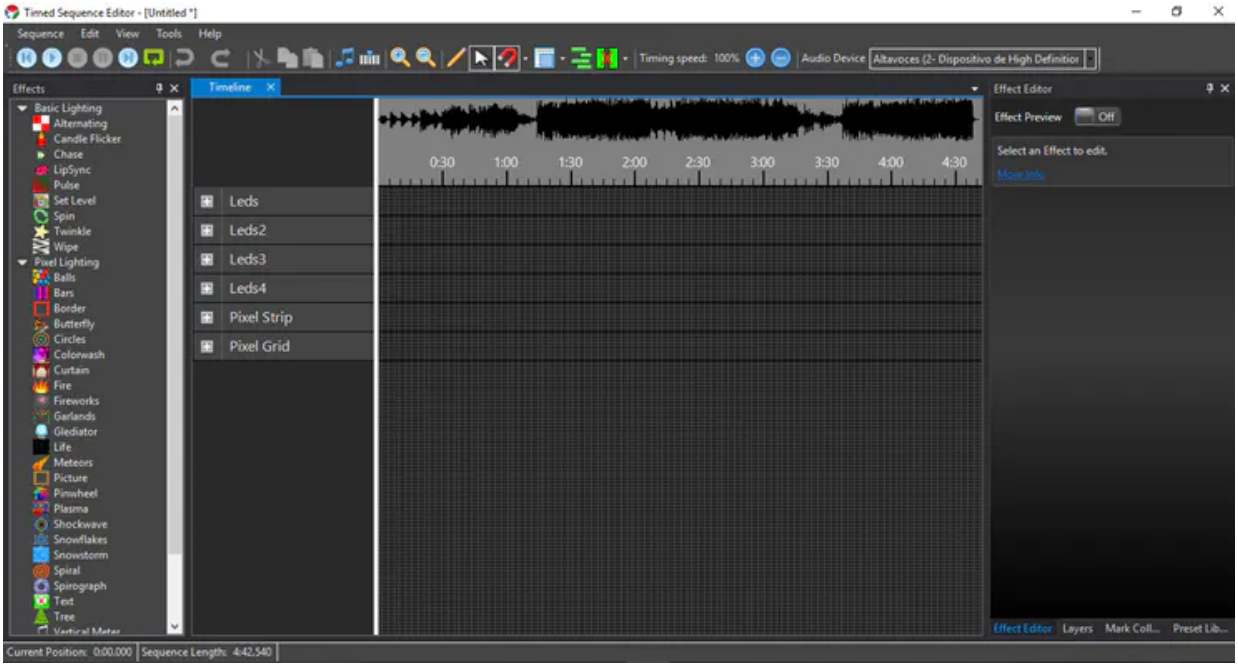
Create my First Sequence

Step 10.- Open Vixen and click on New Sequence…

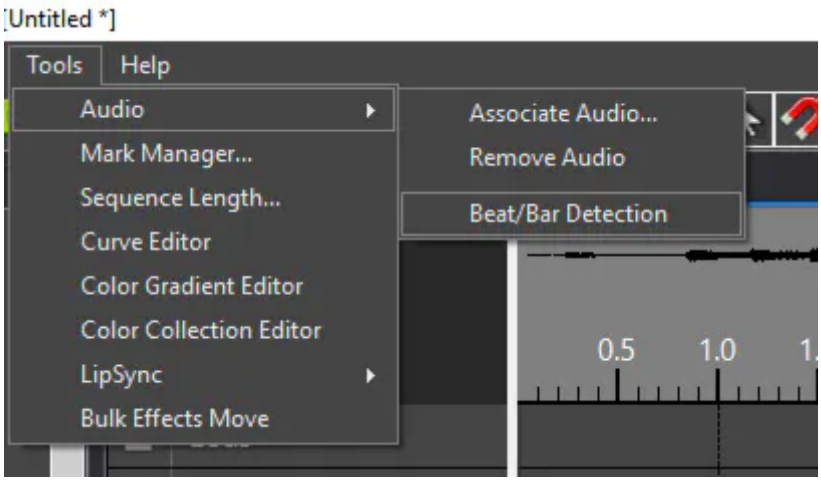
Step 11.- Import audio from the Tools menú, am using mp3.

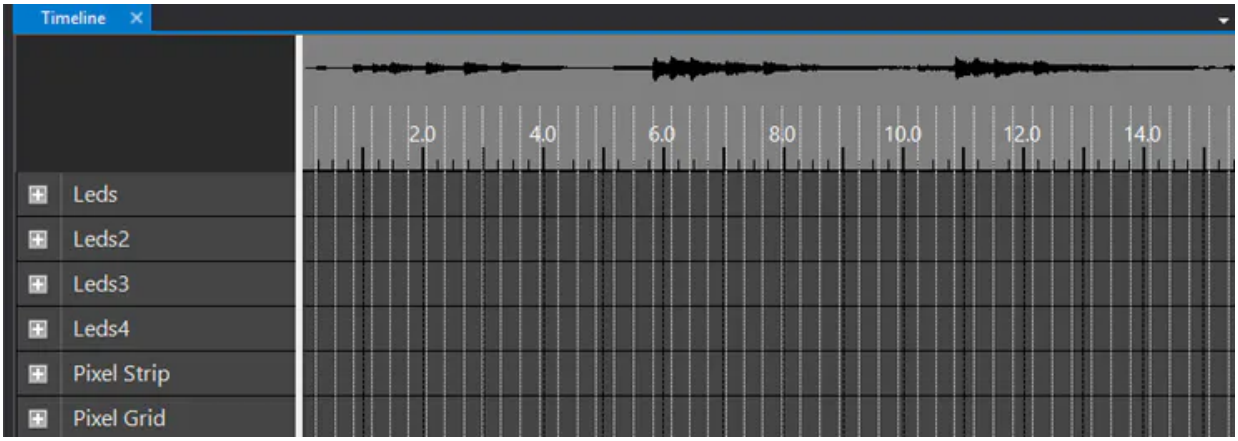


Step 12.-. If you were successful your screen look like ths, you can zoom in or zoom out using the zoom tool this will help in the Timeline.

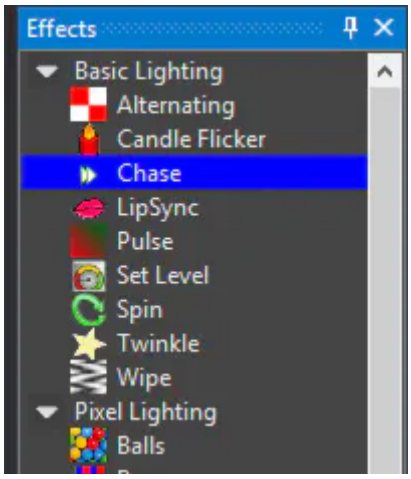


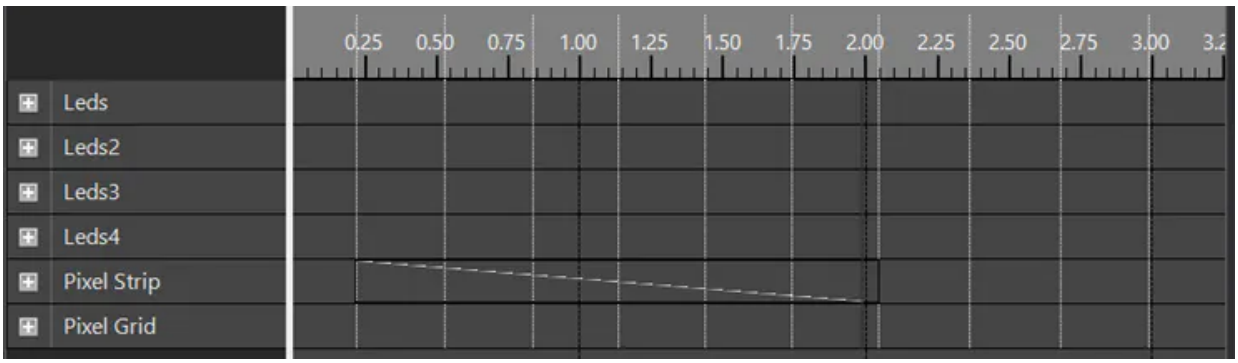
Step 13.- Now we go back to the Tools, Audio and select Beat/Bar Detector, this procces will help to aling perfectly the effects with the audio. You will see a lot of White lines.





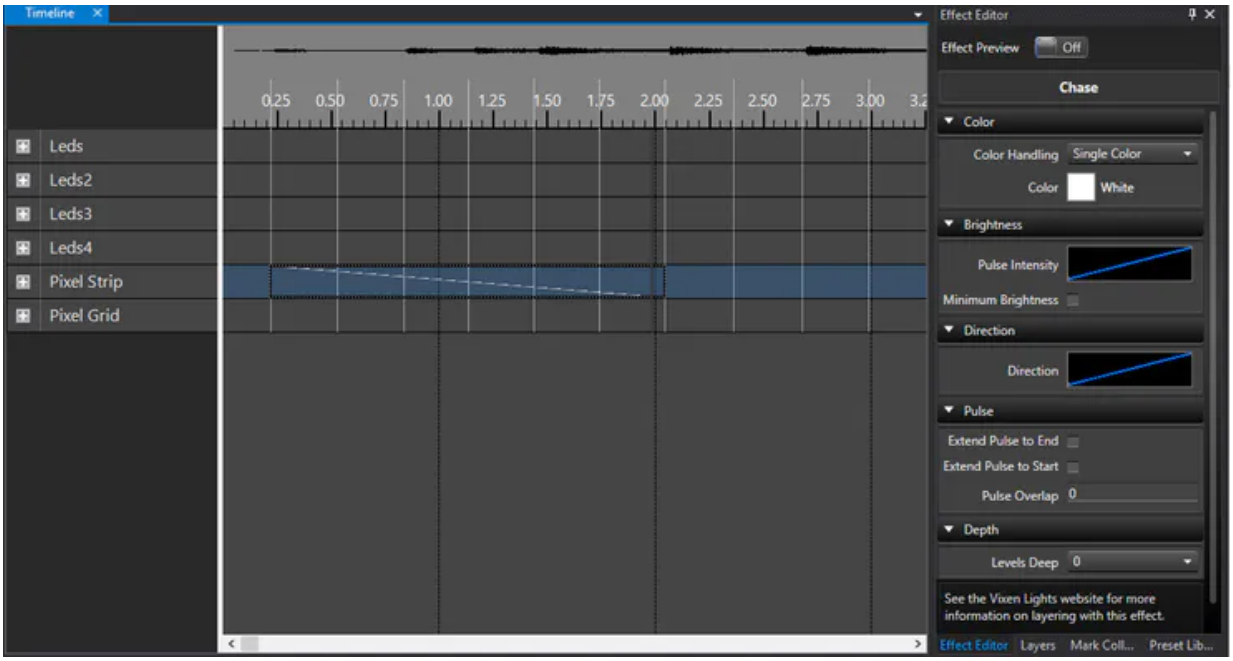
Step 14.- From the left Menu call Effects, ther are 2 submenus, Basic Lighting, Pixel Lighting, both menus can by use with pixels, let´s click on chase, Drag & drop un your Pixel Strip Line, use the mouse to resize the effect.





Note: In this example we are going to see, how the 30 pixels light up on a Chase mode.

Step 15.- Select the Effect, in the ridh menú you will see more option to change direction, color, pulse, depth, etc., play with the effect, also you can actívate the Effect Preview.



Step 16.- Click Play on the Top Left corner, have fun, youtube has many samples.

Note: Ones your Arduino is connected to the computer and open the Vixen software you will see the RX on the Arduino flashing, this means that Arduino is wating for instructions from Vixen..

View Demo

Spanish - Español

Que es Vixen Lights?

Vixen Lights es un software de DIY (hágalo usted mismo) secuencias de luces. La ultimo versión 3.x se rediseño completamente para soportar píxeles RGB inteligentes.

Lo puedes descargar en la siguiente liga http://www.vixenlights.com/downloads/ Este tutorial está basado en la versión 3.4u2 64 Bit.

Que es un Pixel?

Un pixel es un cluster de 3 leds que consiste en 3 colores primarios (rojo, verde y azul). La intensidad de estos colores puede varias para crear otros colores. La tira con pixeles WS2812B que se está usando tiene un chip que acepta data en un Puerto, despliega la información asignada y la pasa la data al siguiente pixel. Para este ejemplo se trata de una tira de pixeles de 4mts, con 60 pixel x metro. Se puede identificar muy fácil una tira de pixeles vs una tira de RGB, la tira de pixeles utiliza 3 cables, +5v, Tierra y Data, mientras una de RGB utiliza 4.

Tira Estandar RGB

Tira Inteligente Pixeles

Los pixeles RGB utilizan mucha energía. Mientras el Arduino puede alimentar algunos pixeles usando su fuente de regulador interno, pero rápidamente se quedará sin corriente. Por esto debemos utilizar una fuente externa para alimentar las luces, El voltaje requerido dependerá el las especificaciones de las luces que haya comprado. La tira de leds inteligentes que se está usando utiliza 5v, pero también hay que tomar en consideración el consumo de amperes. Cada pixel cuando se encienden los 3 leds al 100% consumen 60mA, es decir cada led consume 20mA, si consideramos los 60 pixeles por metro tenemos un consumo de 3.6A x cada metro.

Desafortunadamente las pistas de cobre que se usan en la mayoría de las tiras de leds causan resistencia por lo cual sufren una caída de voltaje. Si el voltaje cae muy bajo tendrás problemas con tus luces como parpadeos, atenuación. Para evitar estos problemas, se necesita insertar voltaje cada 50 pixeles, solo corta la pista, aliméntala y el Data solo crea un jumper. También tomemos en cuenta que si las tiradas de cable son largas debemos tomar en cuenta el grosor del cable para evitar pérdidas.

Lo grande siempre es mejor cuanto una fuente de poder se refiere, para este tipo se recomiendan fuentes de poder de 40A o 60A o utilizar varias.

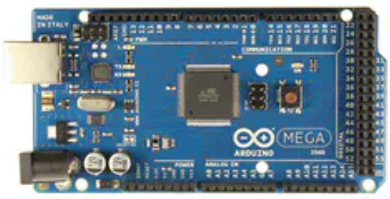
Arduino como Controlador

La mayoría de las tarjetas de Arduino se pueden usar como controladores que sirven de interface entre la computadora que corre Vixen Lights y las titas de pixeles.

Solo que varias tarjetas están limitadas por la velocidad del procesador, tamaño de memoria y tamaño de almacenamiento, pero la mayor limitante es la velocidad de puerto serial. La mayoría de los arduinos no pueden ir más rápido de 115,200 baudios, el topo de pixeles es de 300 para que se puedan refrescar cada 50ms.

Si quieres incrementar el número de pixeles tienes que incrementar en Vixen la frecuencia de actualización a 100ms.

Arduino UNO



Arduino MEGA

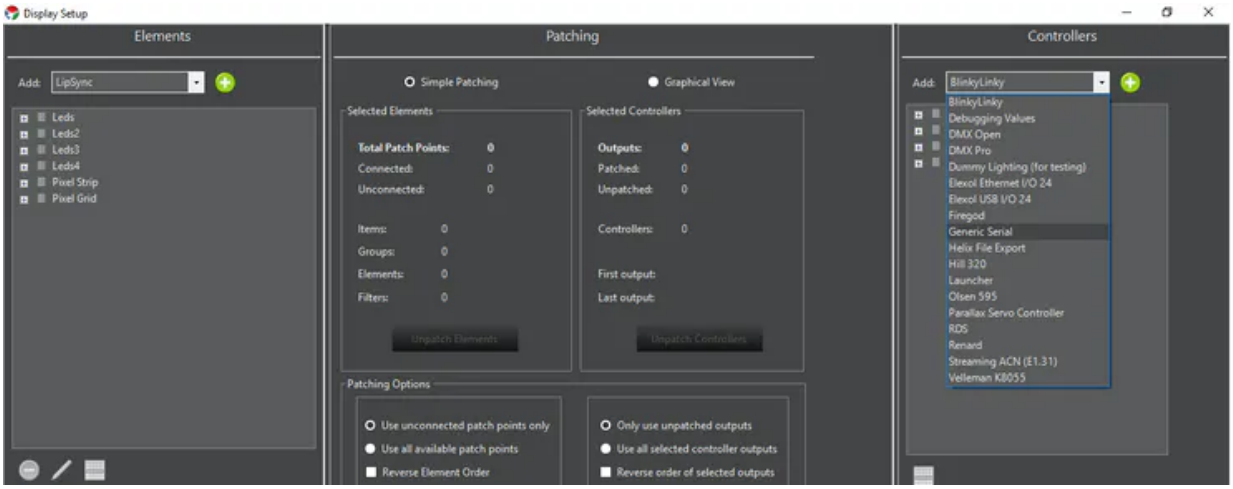
Arduino NANO

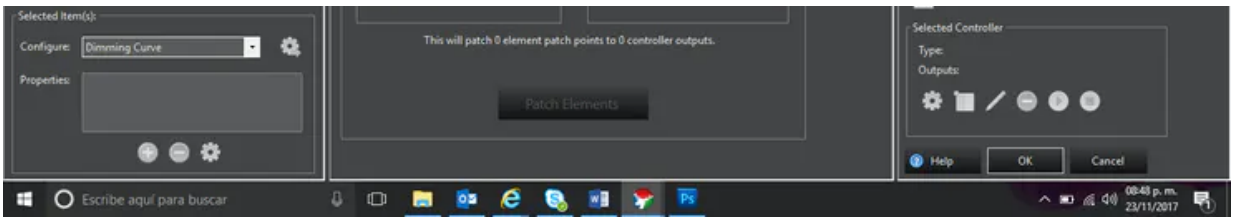
Este tutorial esta basado en el tutorial de David Hunt - blog.huntgang.com

Configurando el Puerto Serial en Vixen

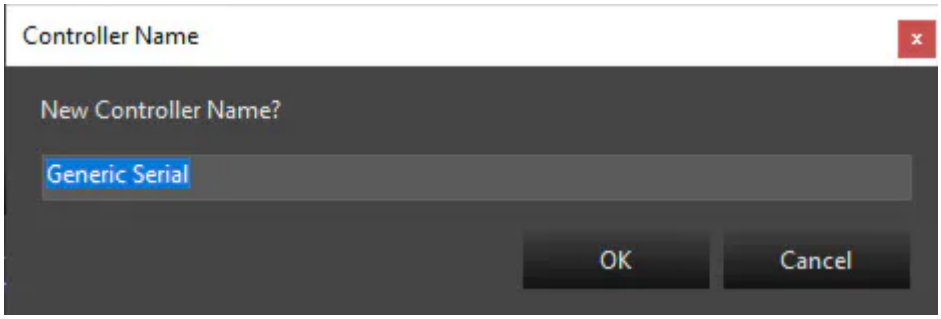
Para poder utilizar el Arduino como controlador, tienes que primero configurar dentro de Vixen 3.x el siguiente proceso.

Paso 1.- Agrega un Generic Serial Controller del menú

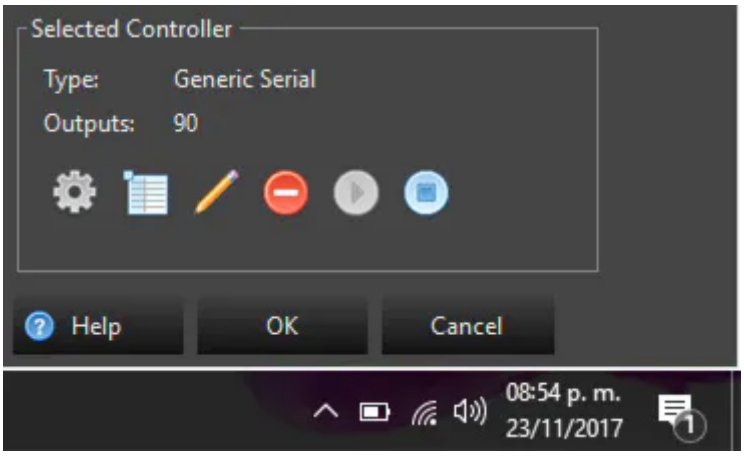


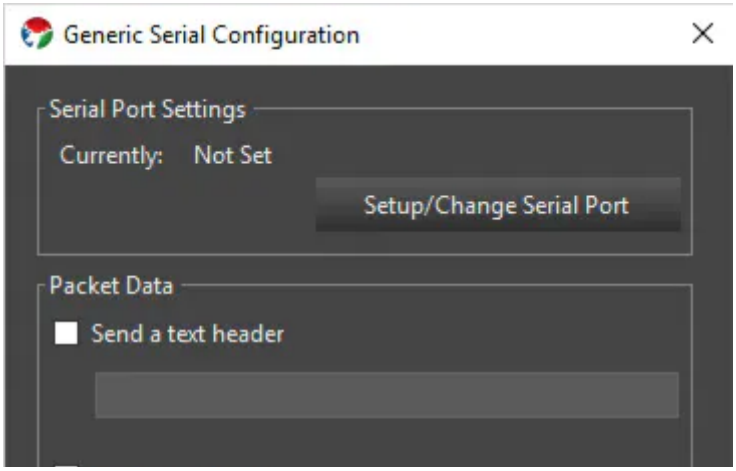


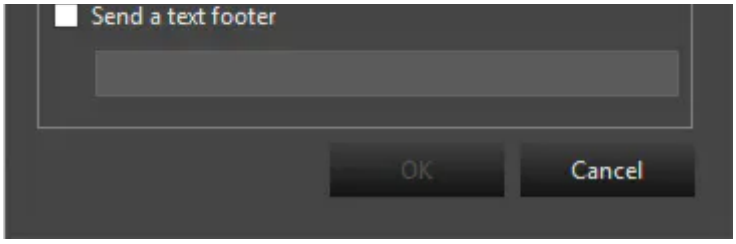
Paso 2.- Ahora definiremos el # de salidas de nuestro controlador, si en mi caso use 30 pixeles debo multiplicarlo x 3, es decir cada pixel tiene 3 leds en su interior, lo que da igual a 90 salidas.

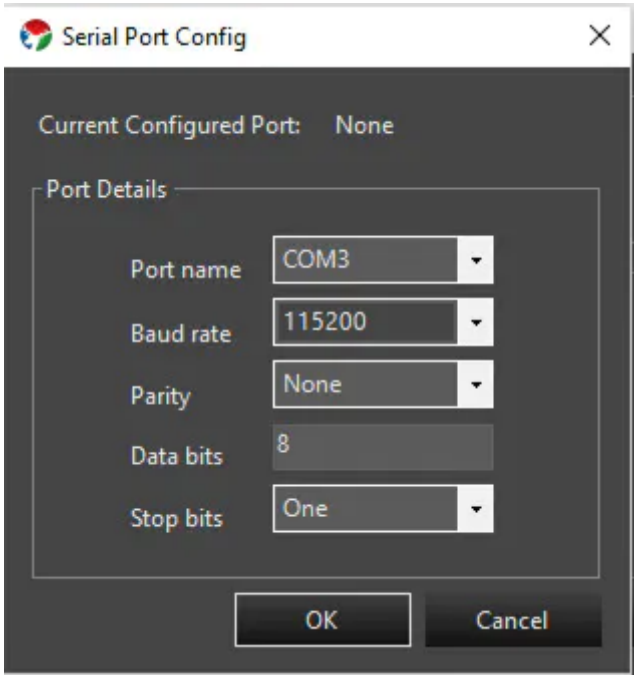


Paso 3.- En la parte inferior derecha dar clic en el engrane, ahora a configurar el puerto COM. Para hacer esto daremos clic en el puerto y seleccionamos el puerto que tiene asignado el Arduino, también configuraremos el velocidad de los baudios a 115200, el resto se queda como esta.

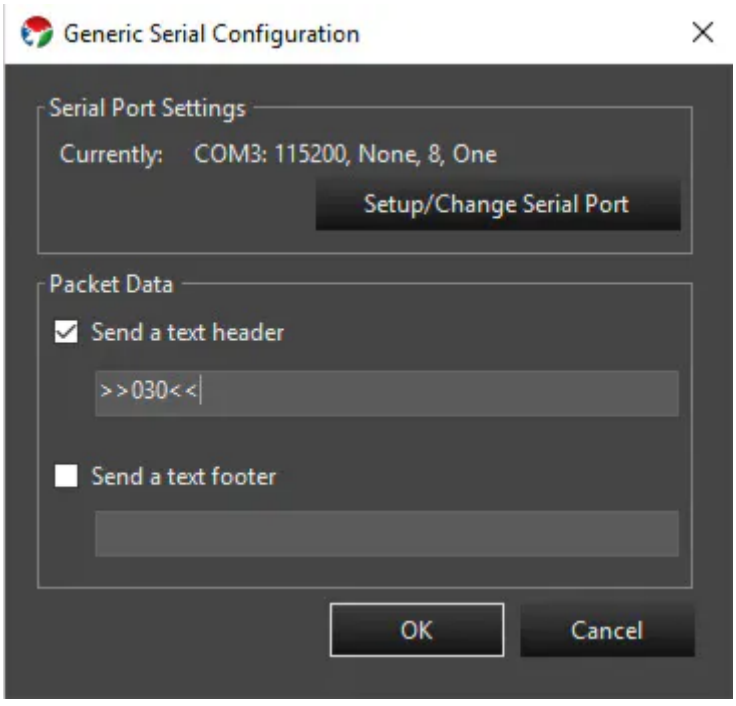






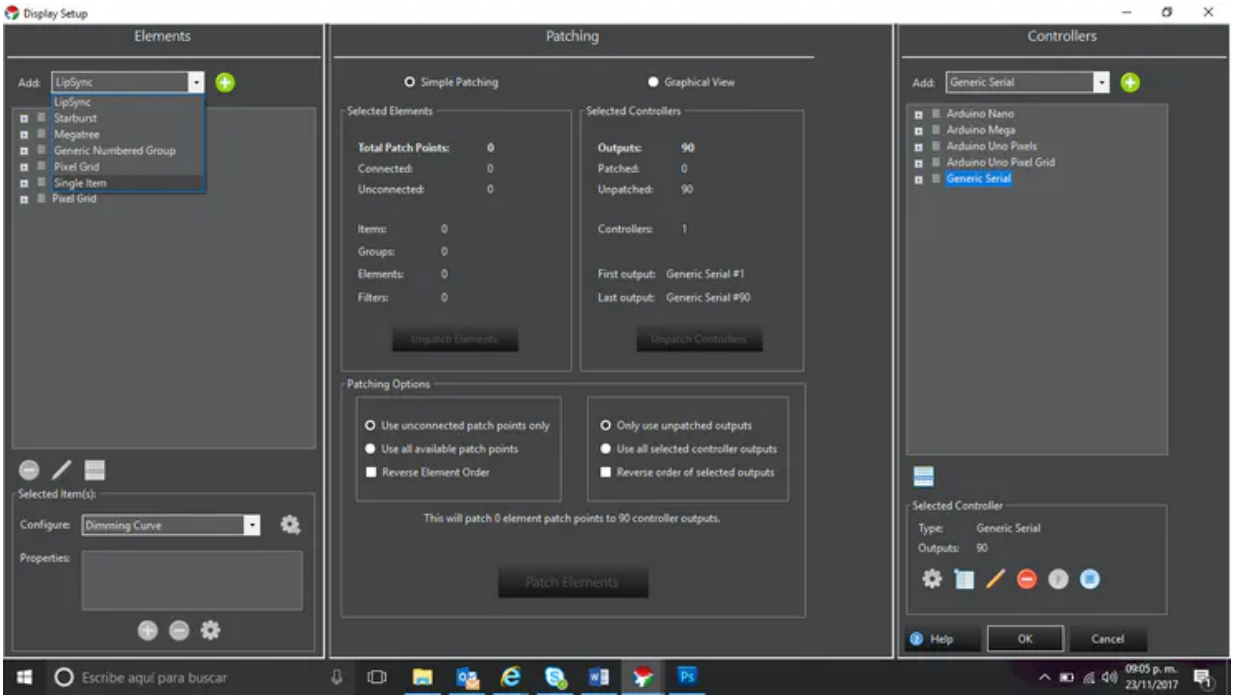


Step 4.- Configuraremos el Texto de Encabezado, en mi caso es >>030<<, esto le indica al código instalado en arduino el número de pixeles que estará recibiendo, siempre el número se debe ingresar en valor de 3 dígitos por eso es importante colocar ceros al inicio, sino tendrás error en los pixeles.

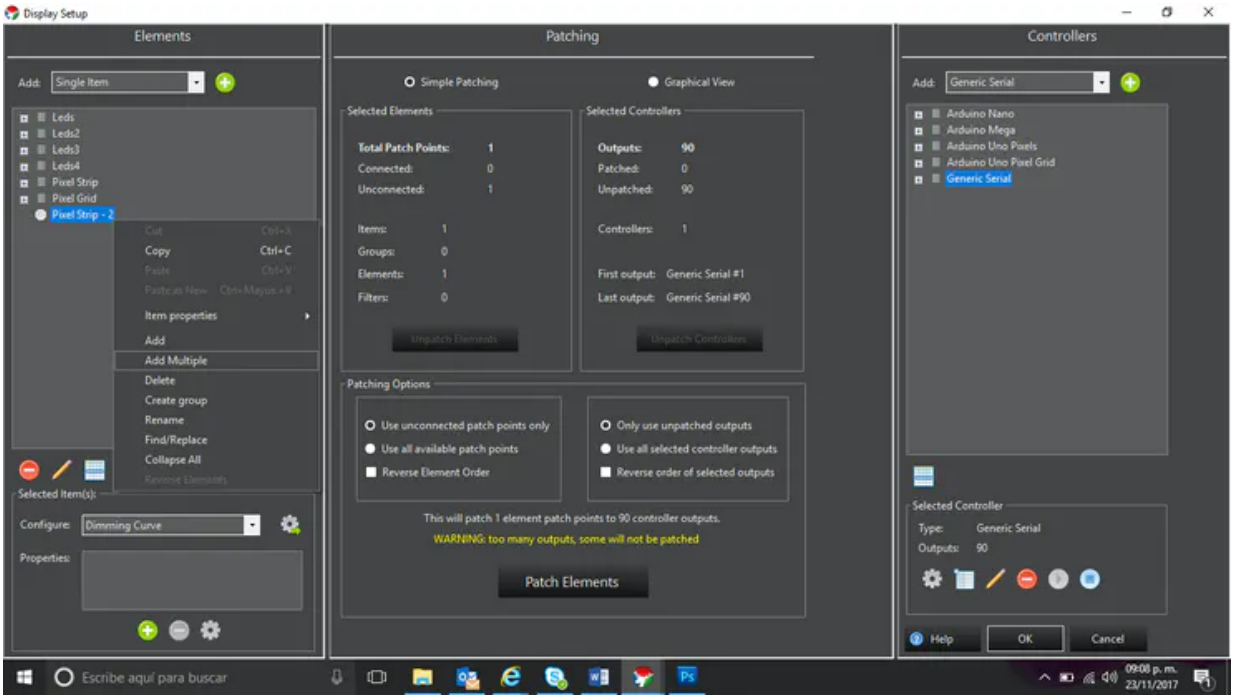


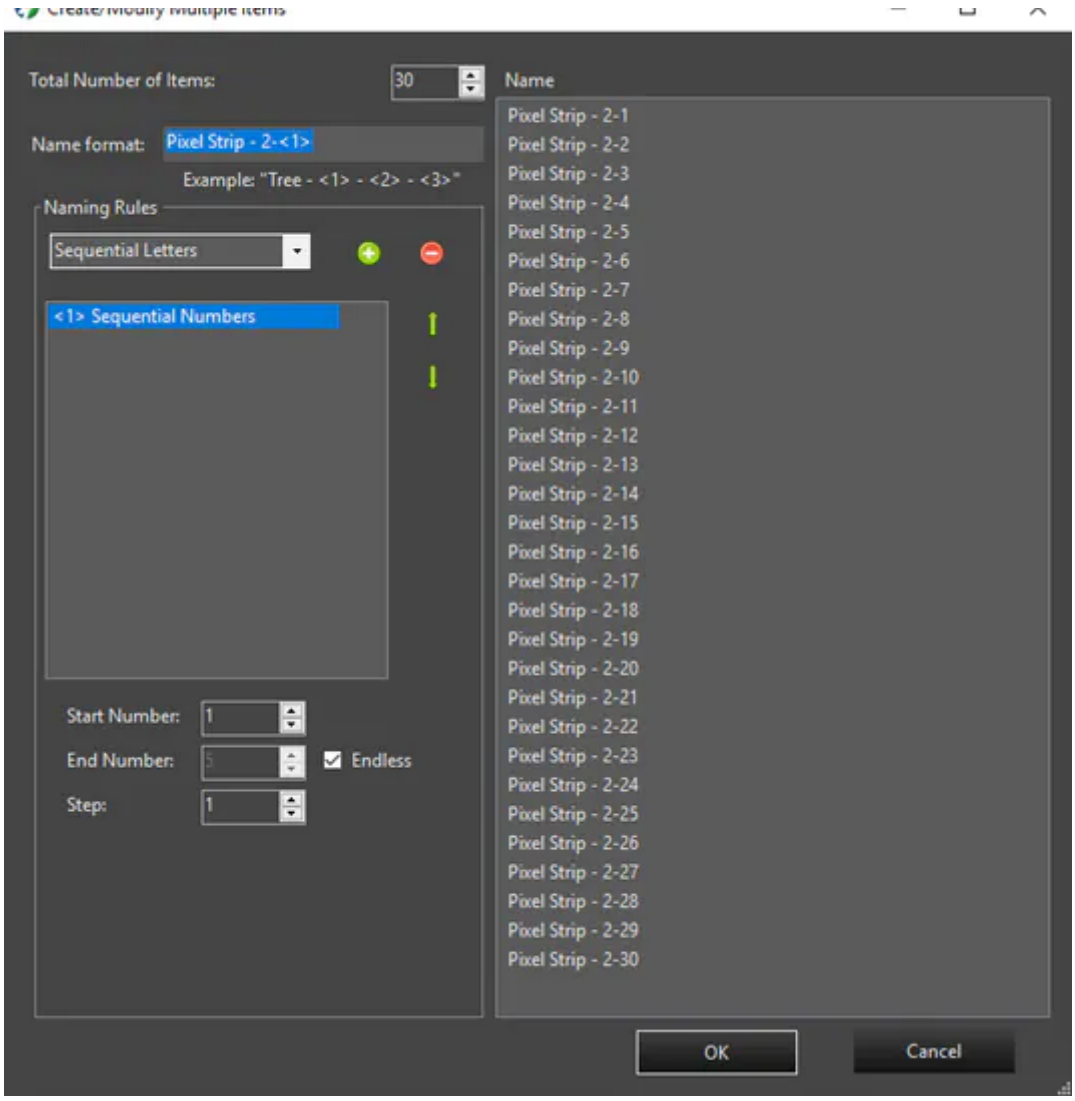
Configurar un Elemento para Pixeles

Paso 5.- En la esquina superior izquierda dentro del cuadro de selección, seleccionamos Single Item, damos clic en el botón verde con un signo de +, nómbralo como Pixel Strip.

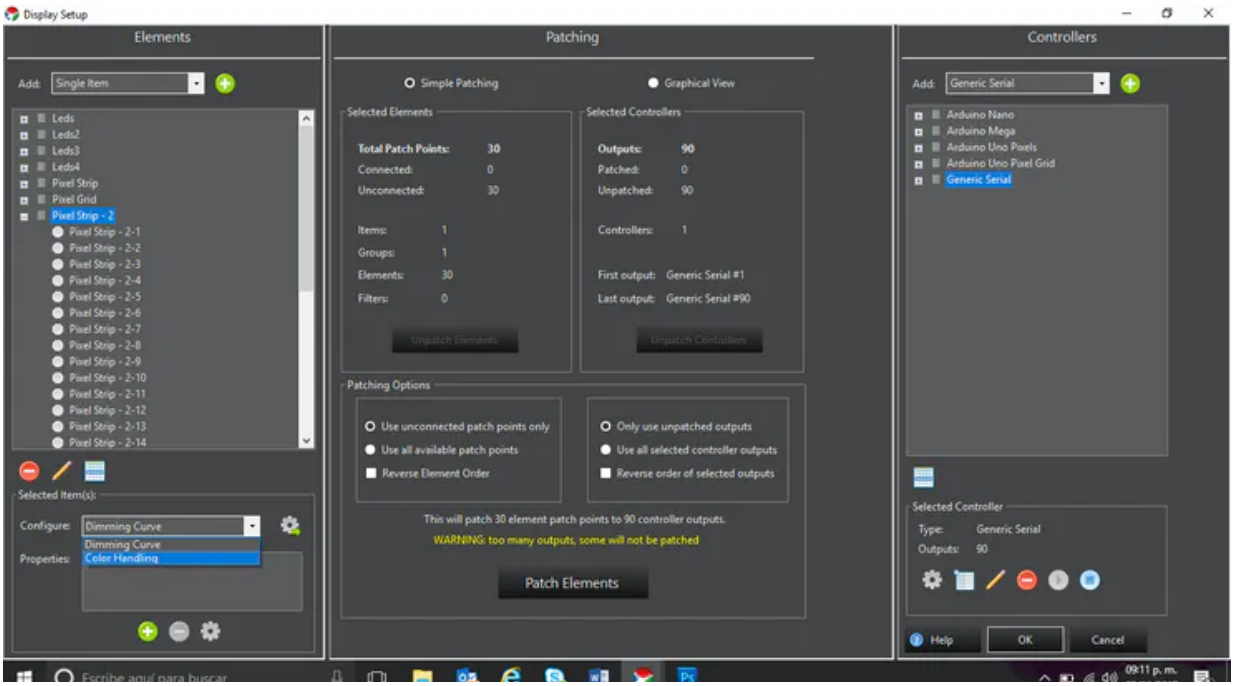


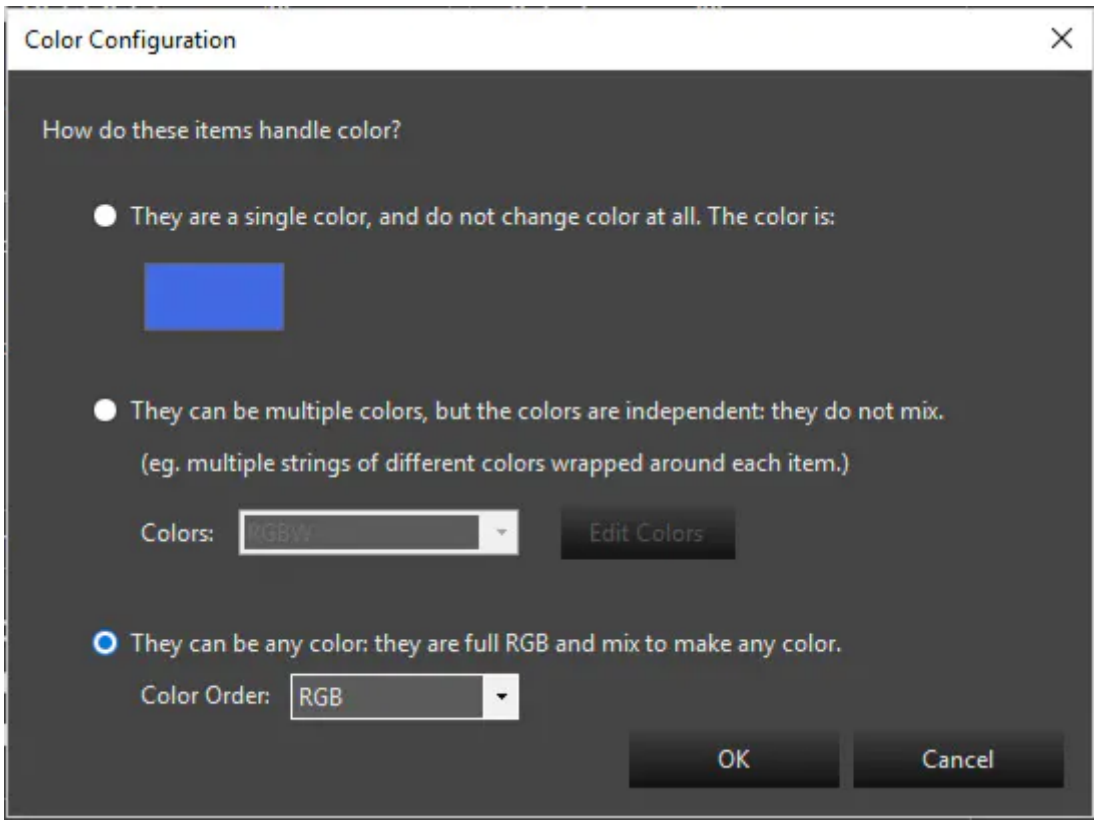
Paso 6.- Sobre el nombre de Pixel Strip que acabas de crear da clic derecho y selecciona Add Multiple. Para añadir todos los pixeles, seleccionaremos ítem numerados, define un nombre, (yo use Pixel Strip) y luego selecciona el número de pixeles que en mi caso con 30. Verás todos los nombres antes de dar clic OK.



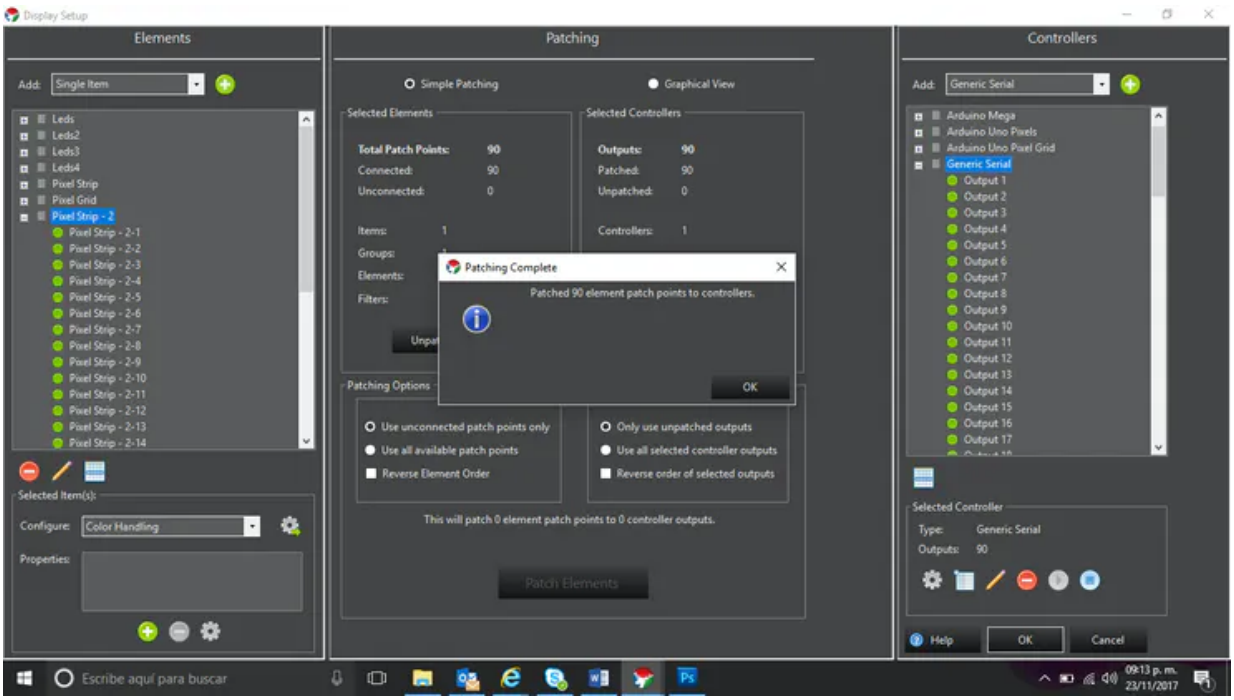


Paso 7.- Ahora seleccionamos el Pixel Strip y configuraremos las propiedades del Color Handiling, que se encuentra en la esquina inferior izquierda en un cuadro que dice Configure:, da clic y selecciona Color Handiling. Seleccionaremos *“They can be any color: they are full RGB and mix any color”*.

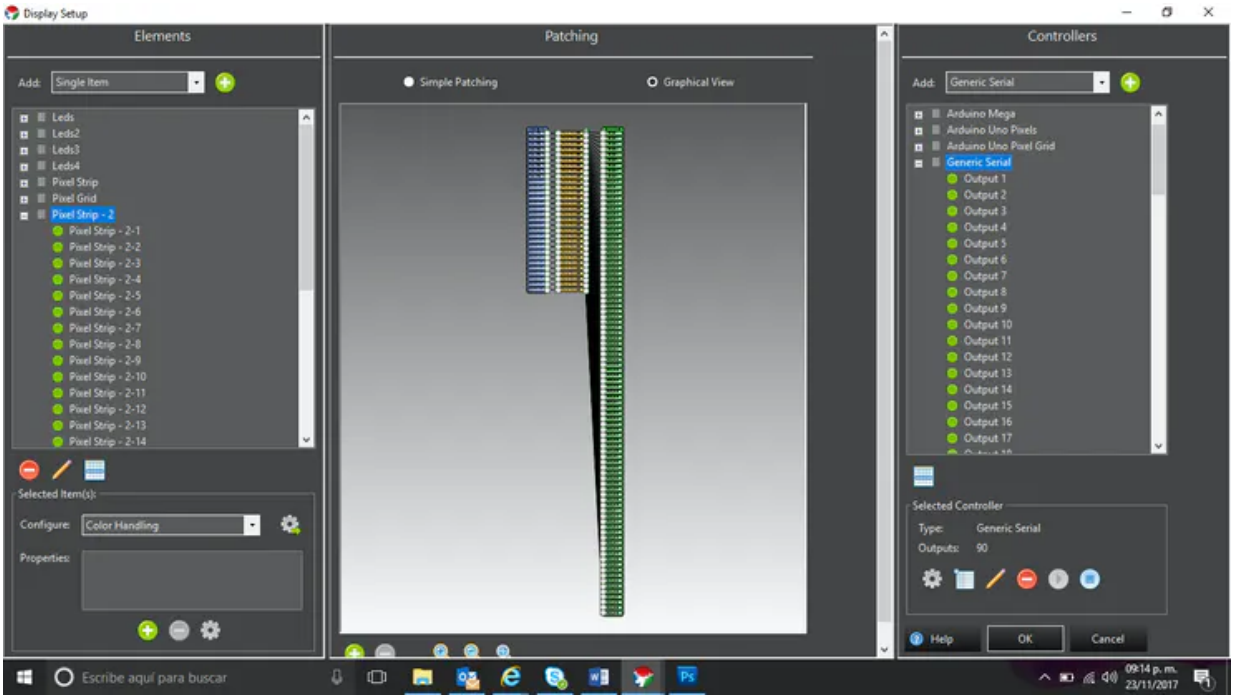




Paso 8.- Este es el paso final donde asignamos los elementos al controlador, para hacer este paso selecciona del lado izquierdo y el controlador genérico del lado derecho bajo el nombre que le hayas puesto. El número de puntos sin conectar debe ser el mismo. Lo único que queda es darle clic en el botón Patch Elements y ya está listo.



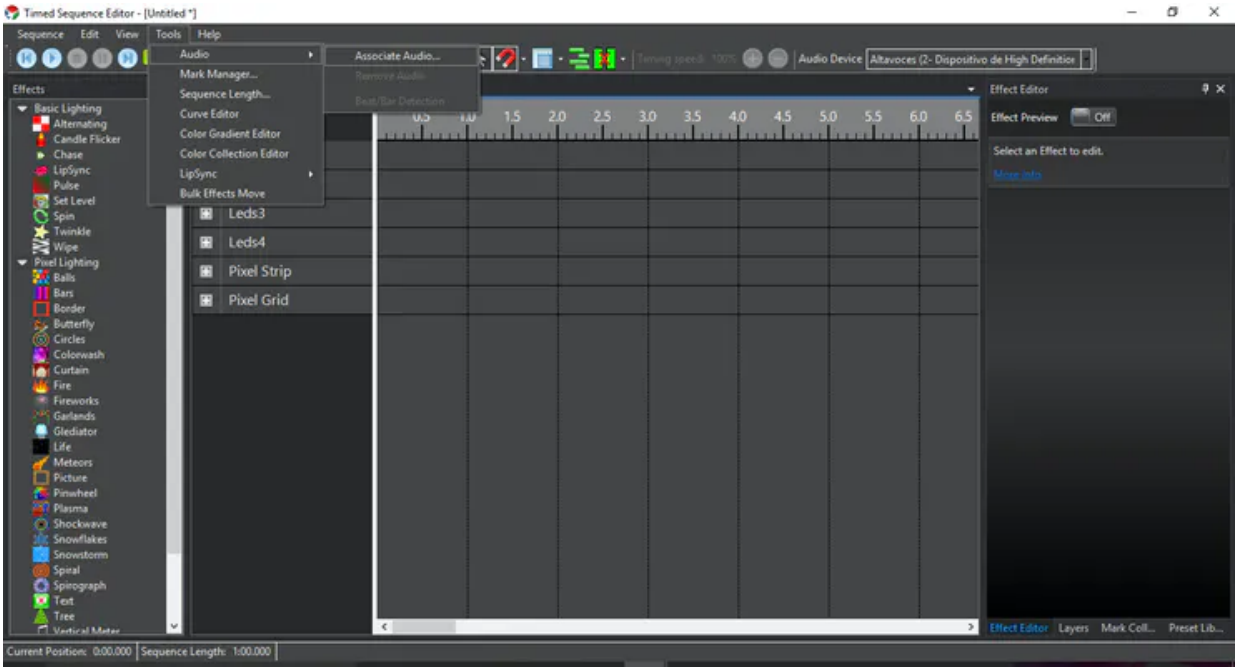
Paso 9.- Para comprobar que esta correcto deberás ver un ejemplo como el de la vista gráfica.



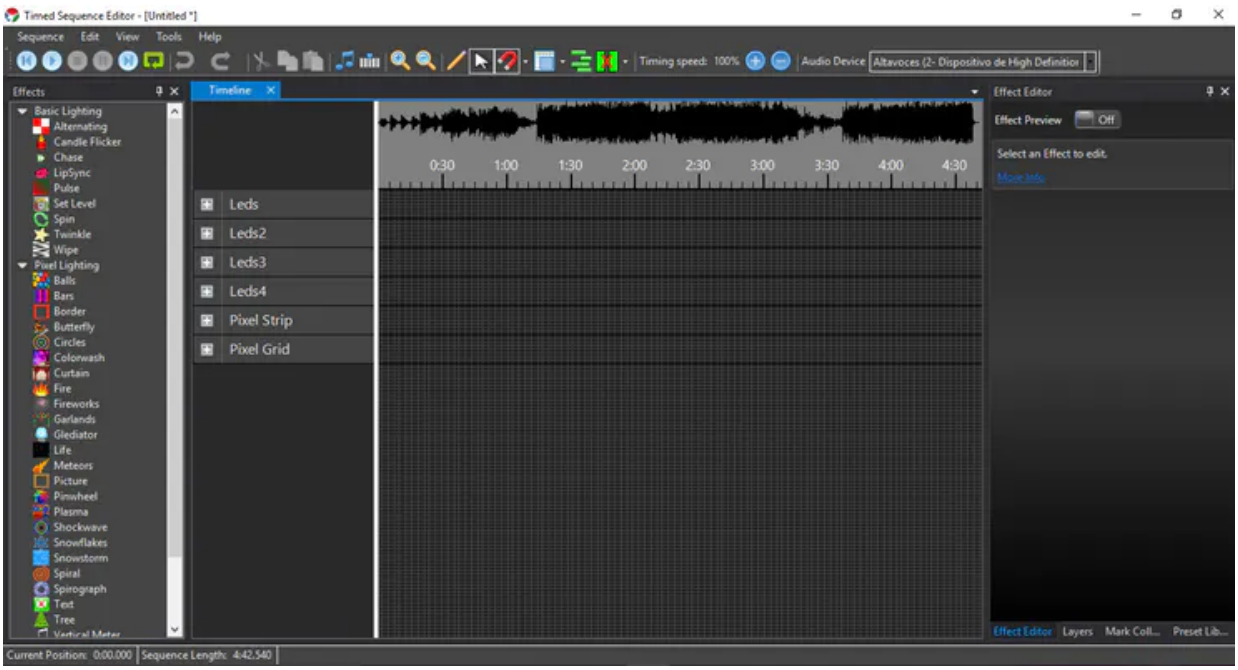
Crear mi Primera Secuencia

Paso 10.- Abrir Vixen y dar clic en New Sequence.....

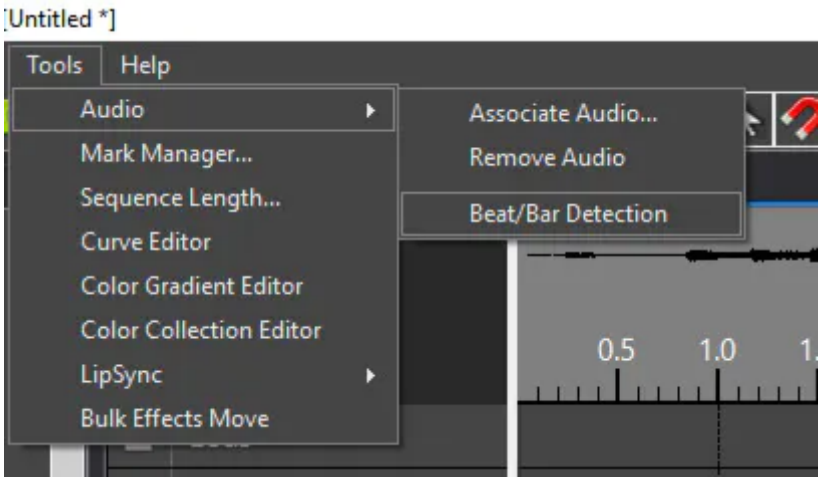
Paso 11.- Importar nuestro Audio, preferente utilizar formatos mp3

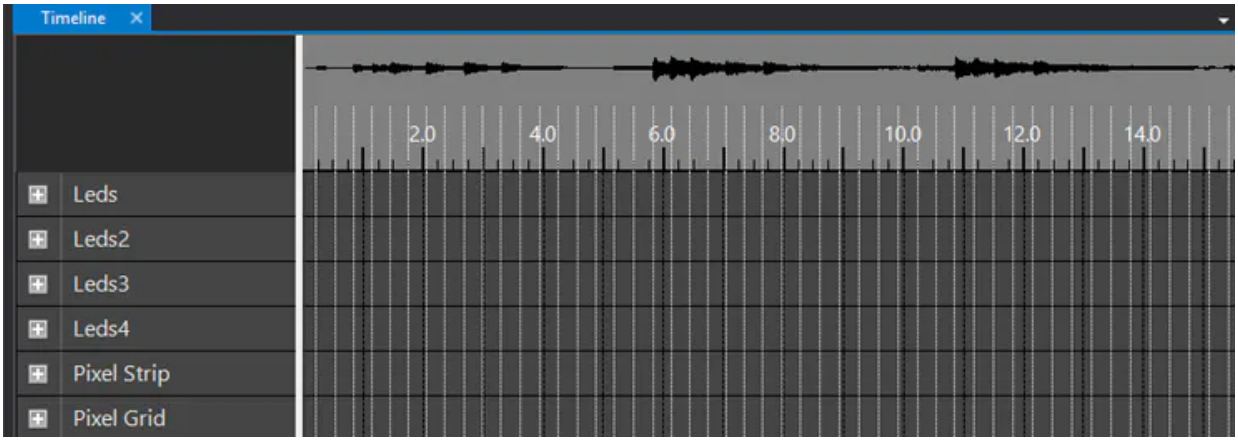


Paso 12.-. Así debe de verse nuestra pantalla, pueden notar que la onda de música está cargada en la parte superior, en la barra superior pueden encuentran dos lupas una de + y otra -, que sirven para alejar o aumentar la línea de tiempo que es el área donde estaremos trabajando.

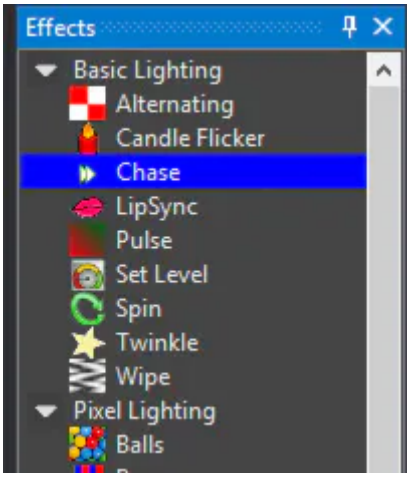


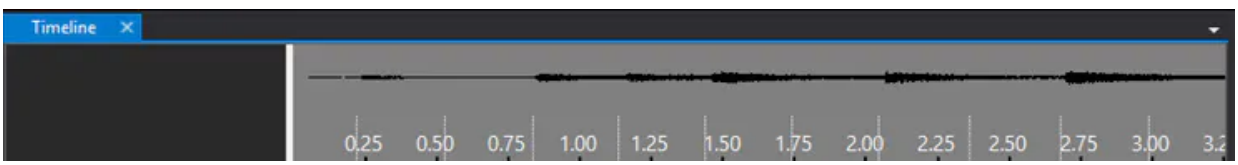
Paso 13.- Vamos a correr un proceso para detectar los Beats de la música, ya que nos ayudará a la hora de armar las secuencias, podrán observar en la imagen las líneas blancas alineadas a los Beats de la música.

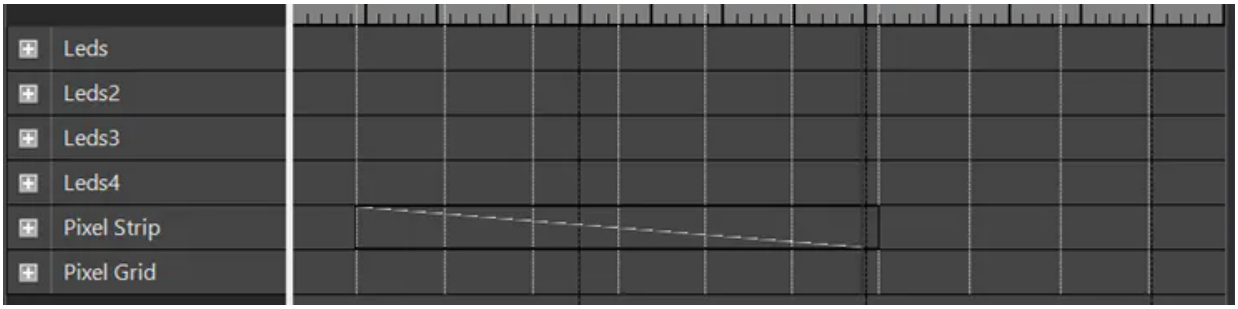




Paso 14.- De nuestro lado izquierdo hay un menú de Effects, Basic Lighting, Pixel Lighting, ambos menús puedes ser utilizados con Pixeles, vamos a dar clic en Chase

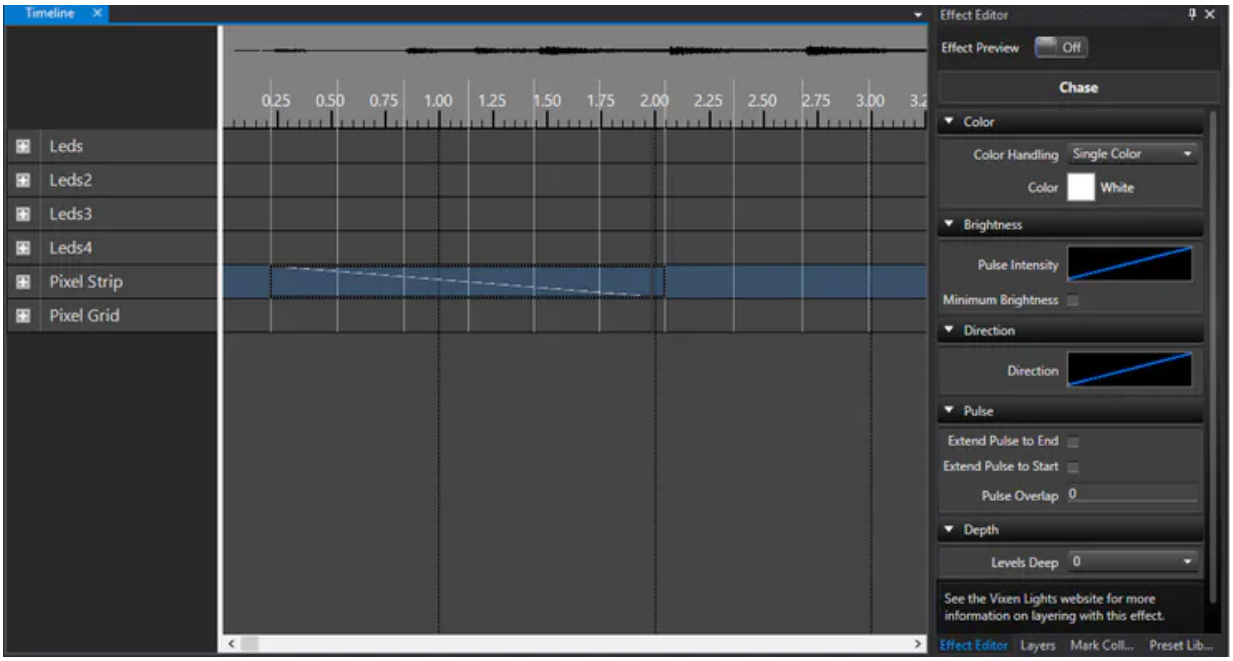






Nota: En este ejemplo veremos cómo le encienden con un efecto de seguimiento los 30 pixeles.

Paso 15.- Seleccionamos el efecto en la línea de tiempo, del lado derecho se activará un menú de configuración del efecto, donde podremos modificar, intensidad, dirección, color, etc.



Paso 16.- Da clic en PLAY, diviértete, hay mucho tutoriales in YouTube.

Nota: Una vez que conectes tu Arduino y abras Vixen Light deben empezar a parpadear los leds Rx, esto indica que Arduino está esperando recibir información por el puerto Serial.