



Imagen de David Hunt - blog.huntgang.com

Como configurar Tiras Inteligente **PIXELES** usando Vixen Lights y Arduino

Que es Vixen Lights?

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

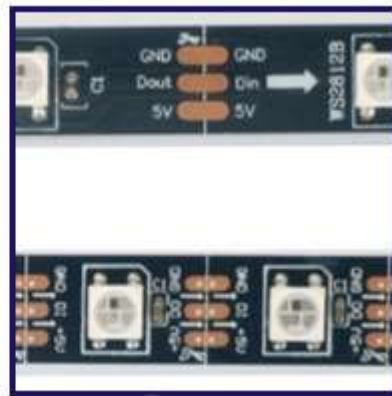
Lo puedes descargar en la siguiente liga <http://www.vixenlights.com/downloads/> Este tutorial está basado en la versión 3.4u2 64 Bit.

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 de 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

El código para el Arduino

Este ejemplo sale de un Tutorial de David Hunt - blog.huntgang.com que utiliza la librería FASTLED.

La librería la puedes descargar de <http://fastled.io> y colocarla en la librería de Arduino dentro de tus documentos.

Vixen Lights 3.x - Arduino Generic Serial for Addressable Pixels

Using this code is pretty straight forward, simply hookup your one wire (WS2811 or WS2812) data line to pin 6 of your Arduino and upload this code. Make sure you have properly installed the FastLED library from <http://fastled.io> Once you are done, simply power your Pixel strips from an external power supply. Next configure a Generic Serial Controller inside of Vixen Lights 3.x and add 3 x pixels for the number of channels. Configure the Generic Serial Controller to use 115200, 8, none, and 1. Then create your element and add "Multiple Items (1 x number of pixels)". Finally select your pixel elements and set them as RGB pixels before patching them to the controller outputs. You should now be ready to begin testing.

For a complete tutorial check out blog.huntgang.com

Created November 8th, 2014

By Richard Sloan - www.themindfactory.com

And David Hunt - blog.huntgang.com

Version 1.4

```
*/
// You must download and install the library from http://fastled.io/
#include <FastLED.h>
// Sets the maximum number of LEDs that this code will handle to avoid running out of memory
#define NUM_LEDS 300

// Sets the pin which is used to connect to the LED pixel strip
#define DATA_PIN 6

CRGB leds[NUM_LEDS];
```

```

void setup() {
  // Define the speed of the serial port
  Serial.begin(115200);
}

void loop() {
  // Set some counter / temporary storage variables
  int cnt;
  unsigned int num_leds;
  unsigned int d1, d2, d3;

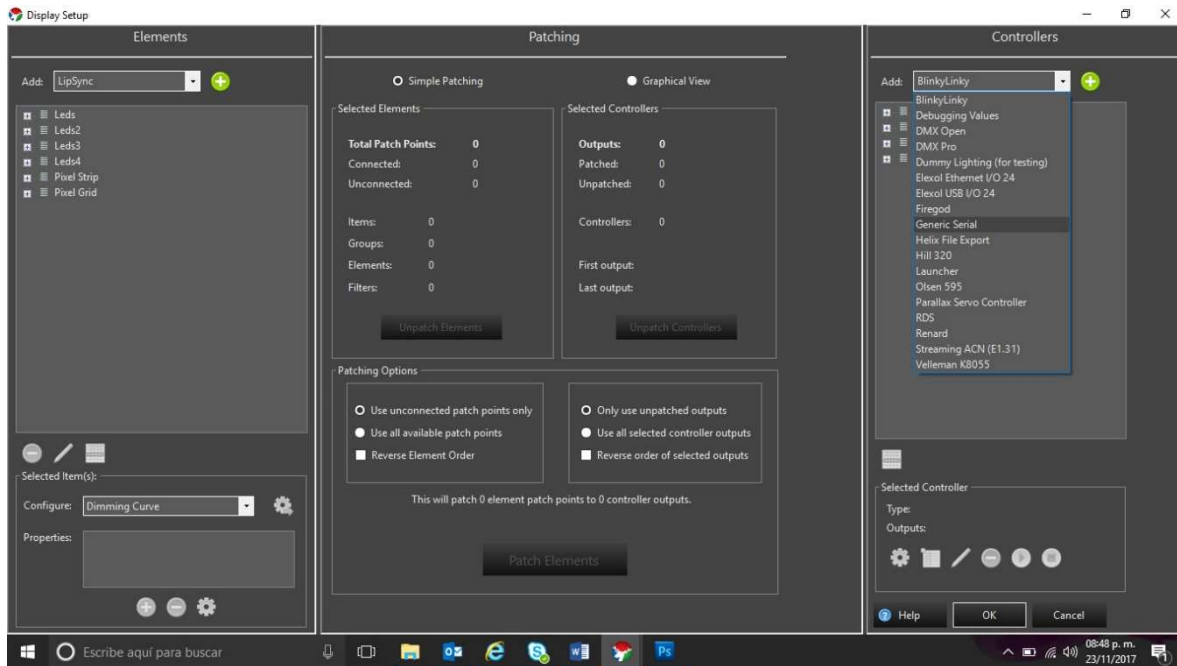
  // Begin an endless loop to receive and process serial data
  for(;;) {
    // Set a counter to 0. This counter keeps track of the pixel colors received.
    cnt = 0;
    // Begin waiting for the header to be received on the serial bus
    // 1st character
    while(!Serial.available());
    if(Serial.read() != '>') {
      continue;
    }
    // second character
    while(!Serial.available());
    if(Serial.read() != '>') {
      continue;
    }
    // get the first digit from the serial bus for the number of pixels to be used
    while(!Serial.available());
    d1 = Serial.read();
    // get the second digit from the serial bus for the number of pixels to be used
    while(!Serial.available());
    d2 = Serial.read();
    // get the third digit from the serial bus for the number of pixels to be used
    while(!Serial.available());
    d3 = Serial.read();
    // get the end of the header
    while(!Serial.available());
    if(Serial.read() != '<') {
      continue;
    }
    while(!Serial.available());
    if(Serial.read() != '<') {
      continue;
    }
    // calculate the number of pixels based on the characters provided in the header digits
    num_leds = (d1-'0')*100+(d2-'0')*10+(d3-'0');
    // ensure the number of pixels does not exceed the number allowed
    if(num_leds > NUM_LEDS) {
      continue;
    }
    // Let the FastLED library know how many pixels we will be addressing
    FastLED.addLeds<WS2812B, DATA_PIN, GRB>(leds, num_leds);
    // Loop through each of the pixels and read the values for each color
    do {
      while(!Serial.available());
      leds[cnt].r = Serial.read();
      while(!Serial.available());
      leds[cnt].g = Serial.read();
      while(!Serial.available());
      leds[cnt++].b = Serial.read();
    } while(--num_leds);
    // Tell the FastLED Library it is time to update the strip of pixels
    FastLED.show();
    // WOO HOO... We are all done and are ready to start over again!
  }
}

```

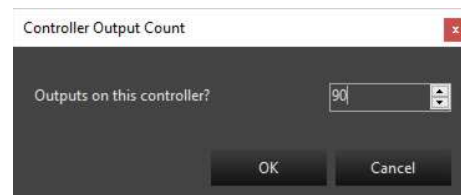
Configurando el Puerto Serial en Vixen

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

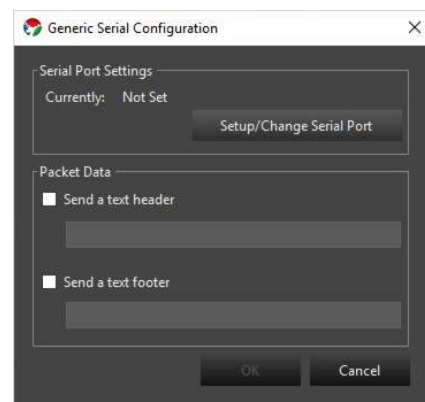
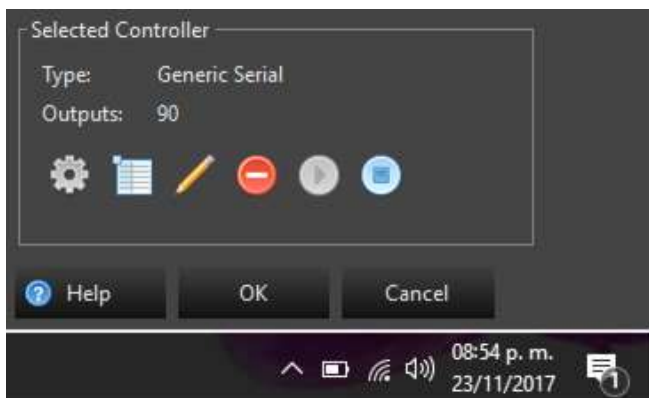
Paso 1.- Agrega un Generic Serial Controller del menú

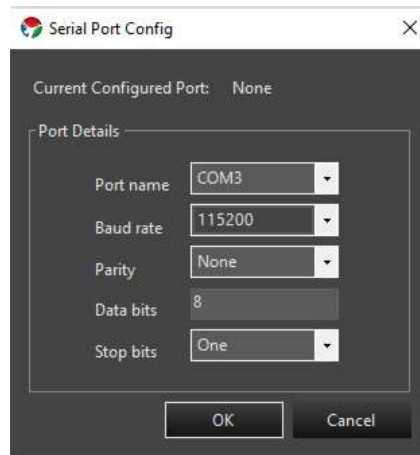


Paso 2.- Ahora definiremos el # de salidas de nuestro controlador, si en mi caso use 30 pixeles debo multiplicarlo x 3, es decir cada pixel tiene 3 leds en su interior, lo que da igual a 90 salidas.

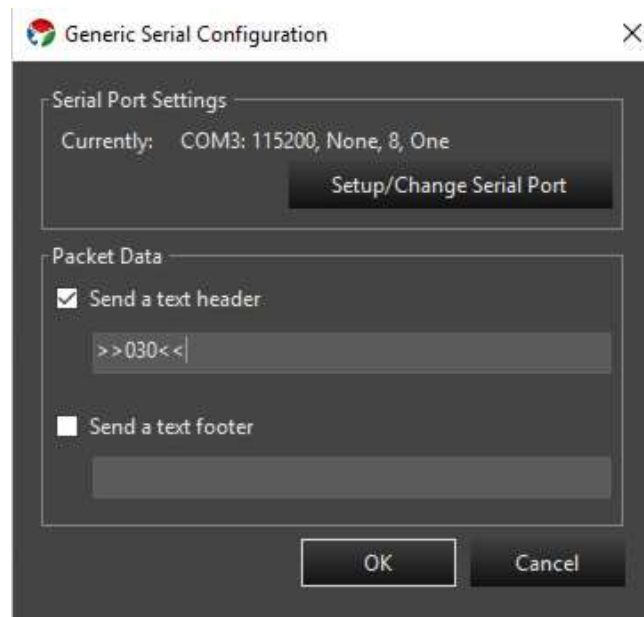


Paso 3.- En la parte inferior derecha dar clic en el engrane, ahora a configurar el puerto COM. Para hacer esto daremos clic en el puerto y seleccionamos el puerto que tiene asignado el Arduino, también configuraremos el velocidad de los baudios a 115200, el resto se queda como esta.



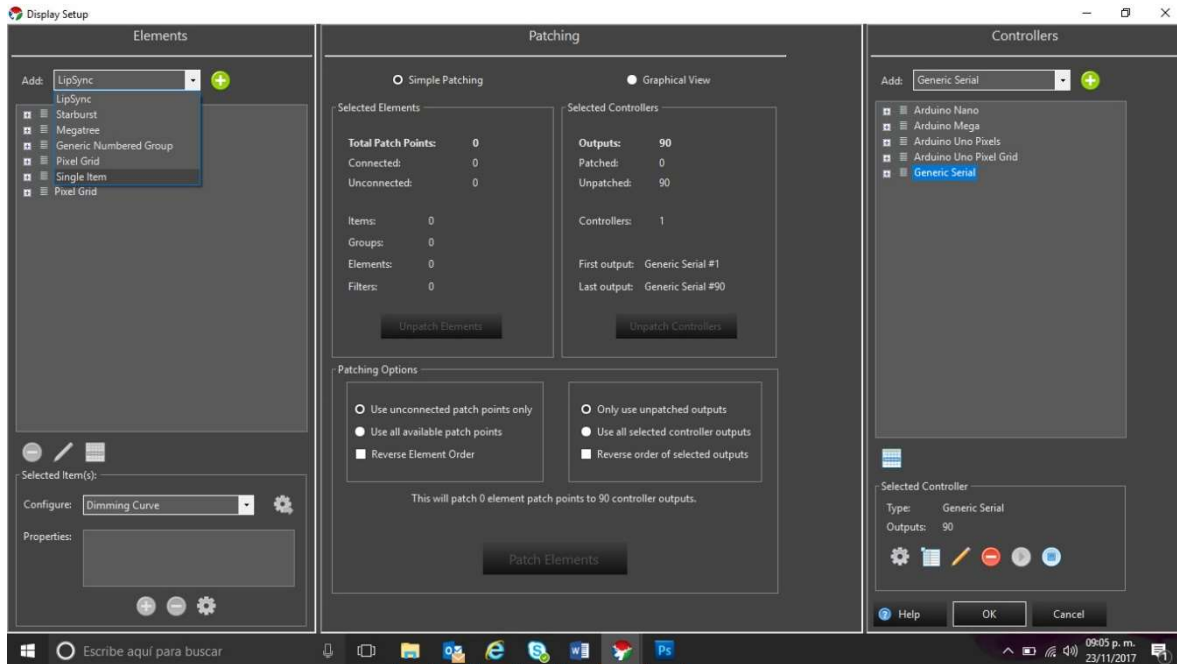


Paso 4.- Configuraremos el Texto de Encabezado, en mi caso es >>030<<, esto le indica al código instalado en arduino el número de pixeles que estará recibiendo, siempre el número se debe ingresar en valor de 3 dígitos por eso es importante colocar ceros al inicio, sino tendrás error en los pixeles.

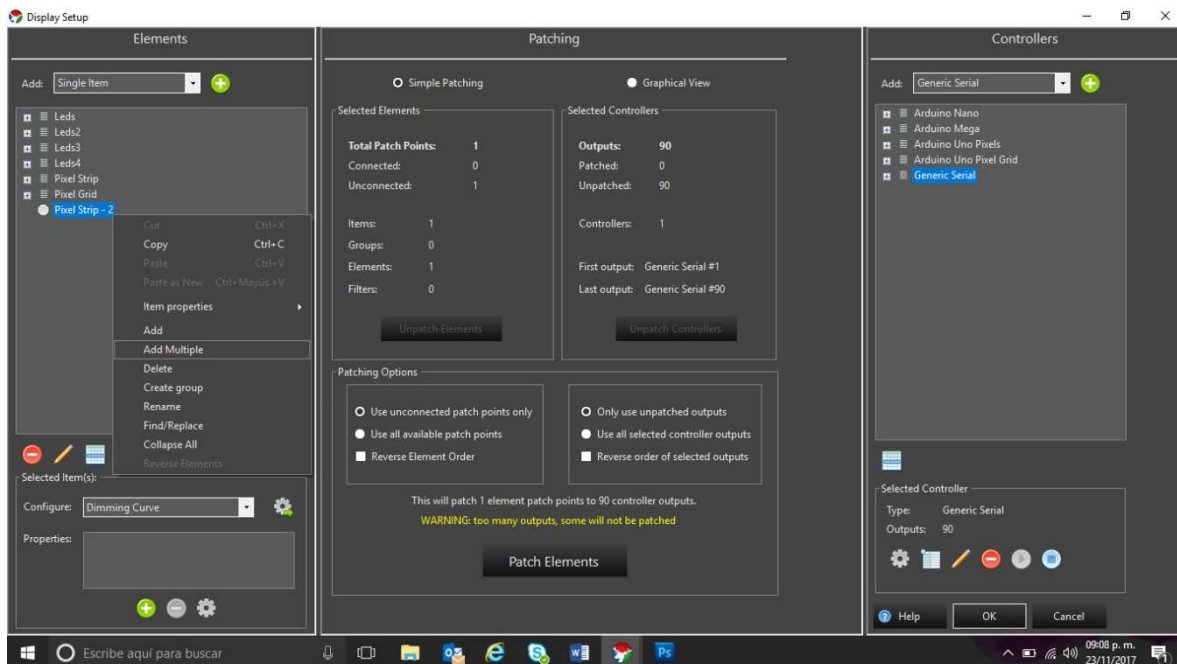


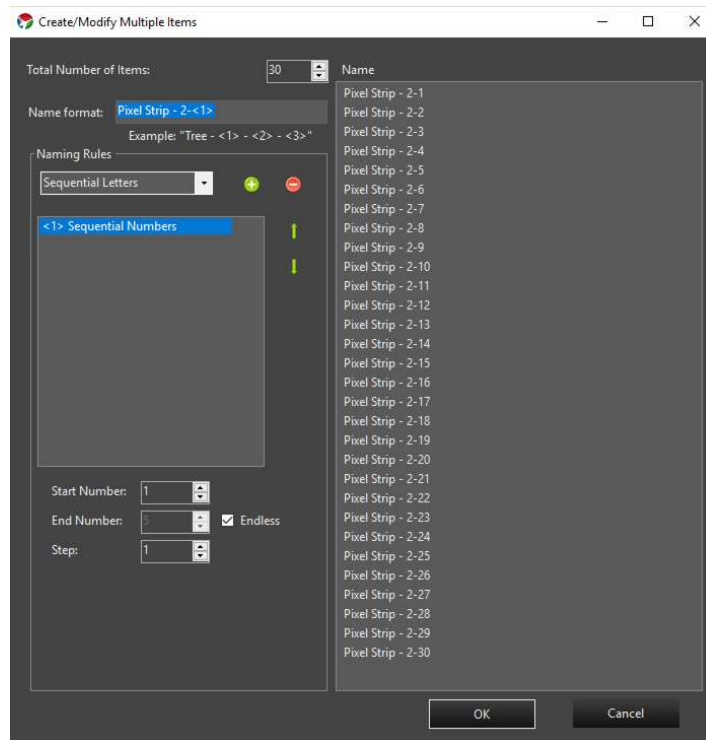
Configurar un Elemento para Pixeles

Paso 1.- 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.

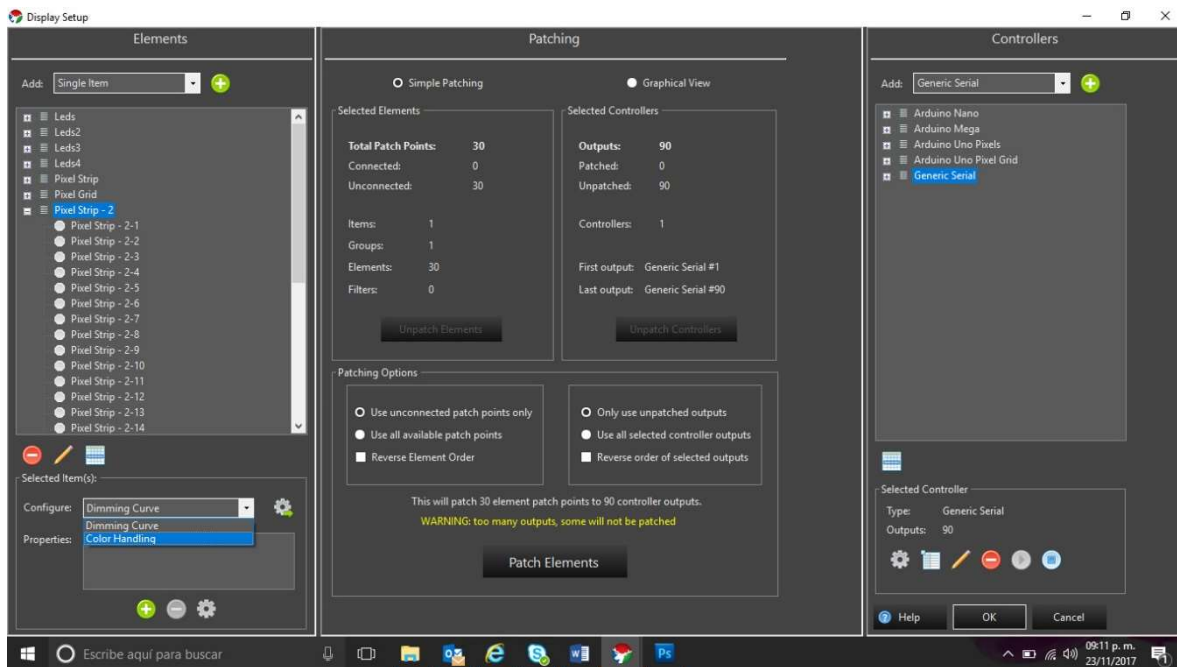


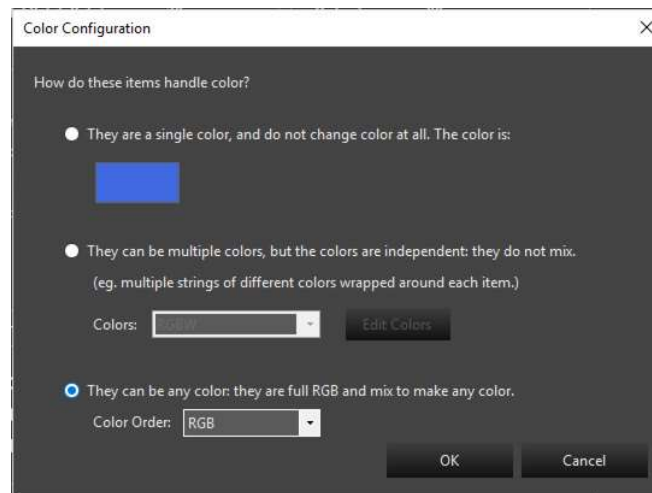
Paso 2.- Sobre el nombre de Pixel Strip que acabas de crear da clic derecho y selecciona Add Multiple. Para añadir todos los pixeles, seleccionaremos ítem numerados, define un nombre, (yo use Pixel Strip) y luego selecciona el número de pixeles que en mi caso con 30. Verás todos los nombres antes de dar clic OK.



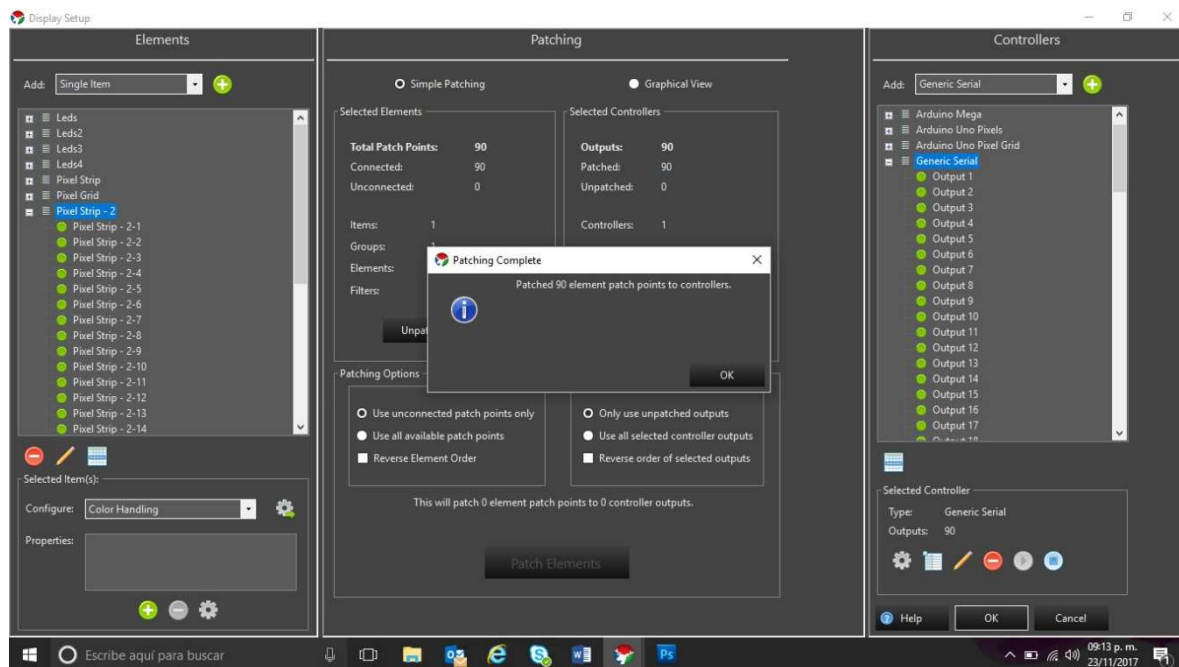


Paso 3.- Ahora seleccionamos el Pixel Strip y configuraremos las propiedades del Color Handling, que se encuentra en la esquina inferior izquierda en un cuadro que dice Configure:, da clic y selecciona Color Handling. Seleccionaremos *“They can be any color: they are full RGB and mix any color”*.





Paso 4.- 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 5.- Para comprobar que esta correcto deberás ver un ejemplo como el de la vista gráfica.

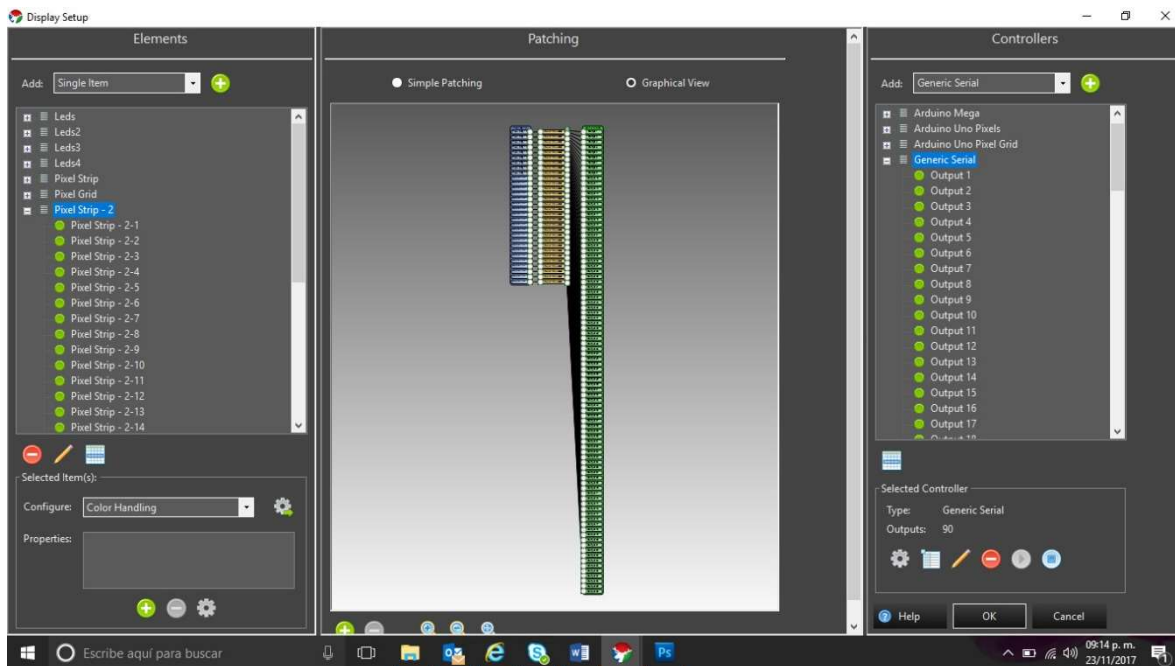
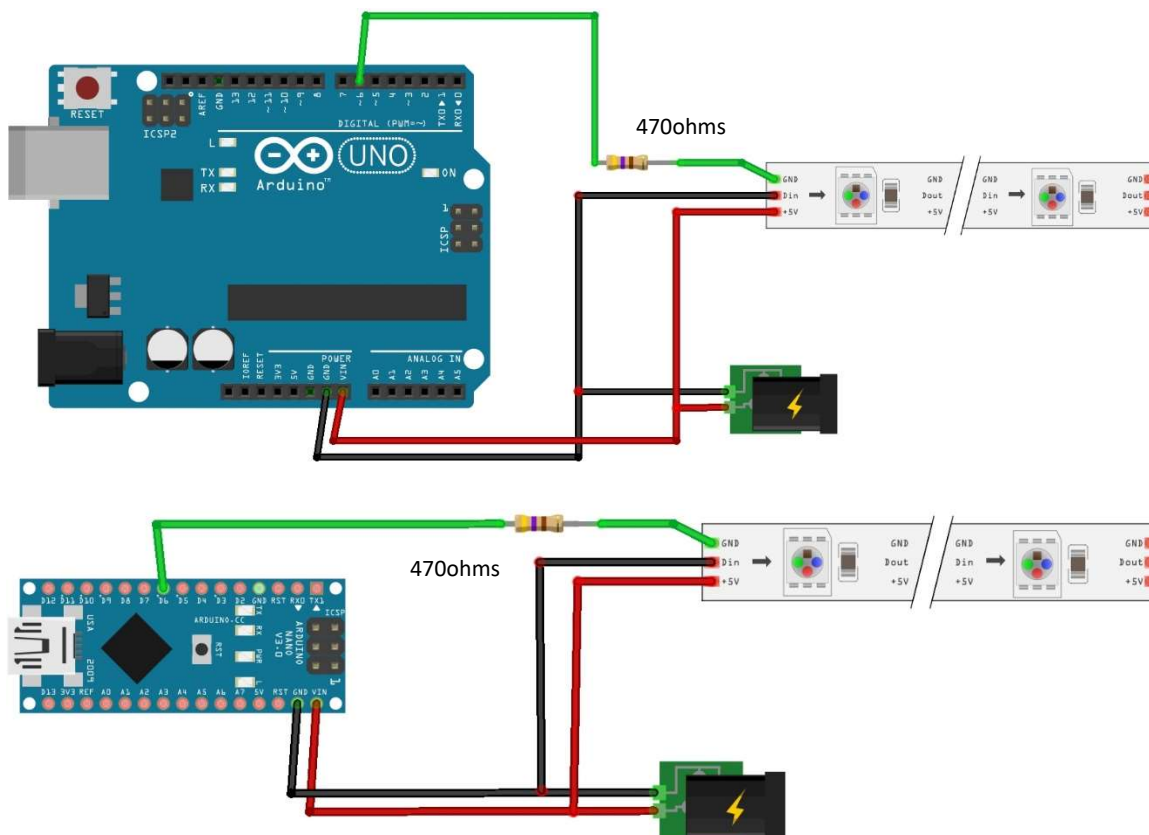


Diagrama de Conexión de Arduino UNO y Arduino NANO

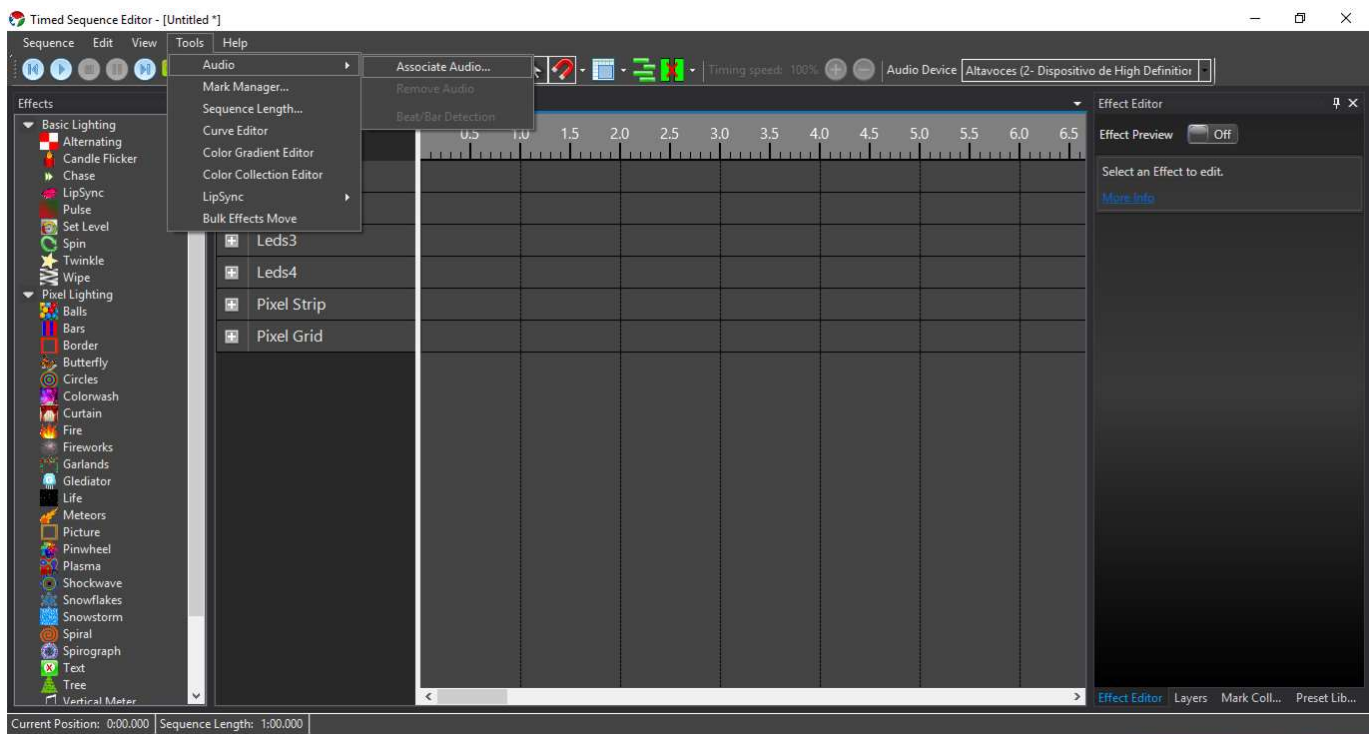


Crear mi Primera Secuencia

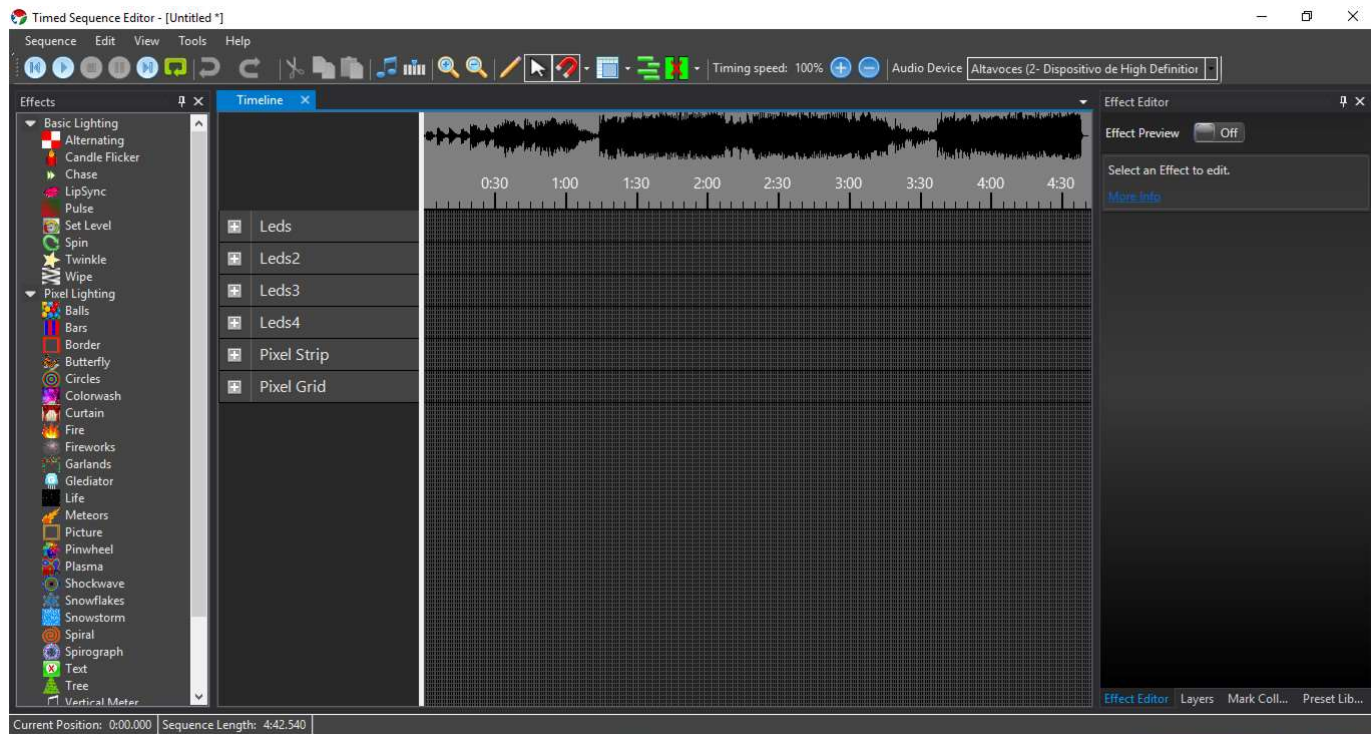
Paso 1.- Dar clic en New Sequence...



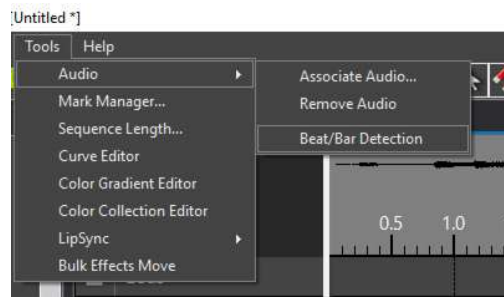
Paso 2.- Importar nuestro Audio, preferente utilizar formatos mp3

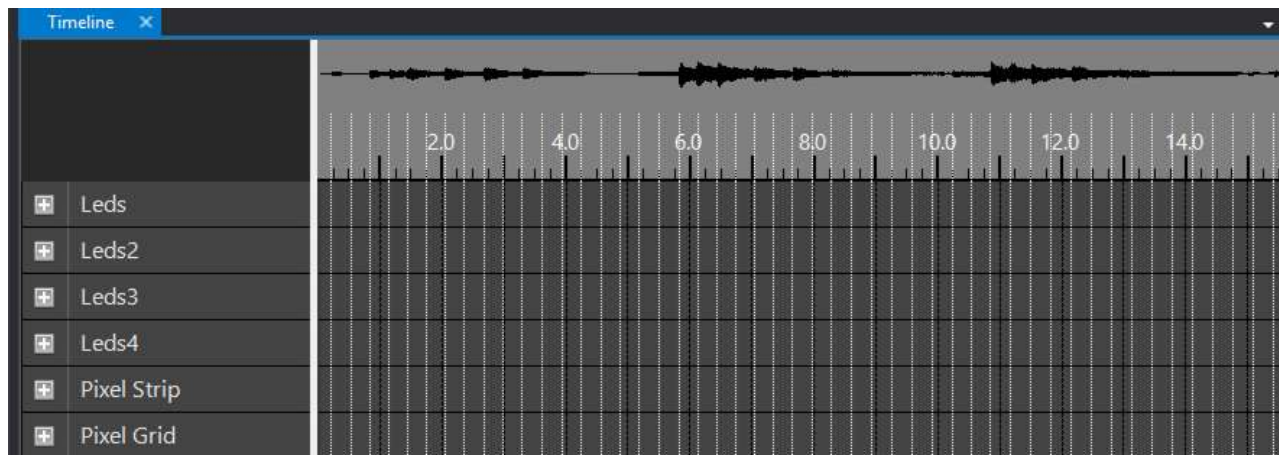


Paso 3.- Así debe de verse nuestra pantalla, pueden notar que la onda de música está cargada en la parte superior, en la barra superior pueden encontrar dos lupas una de + y otra -, que sirven para alejar o aumentar la línea de tiempo que es el área donde estaremos trabajando.

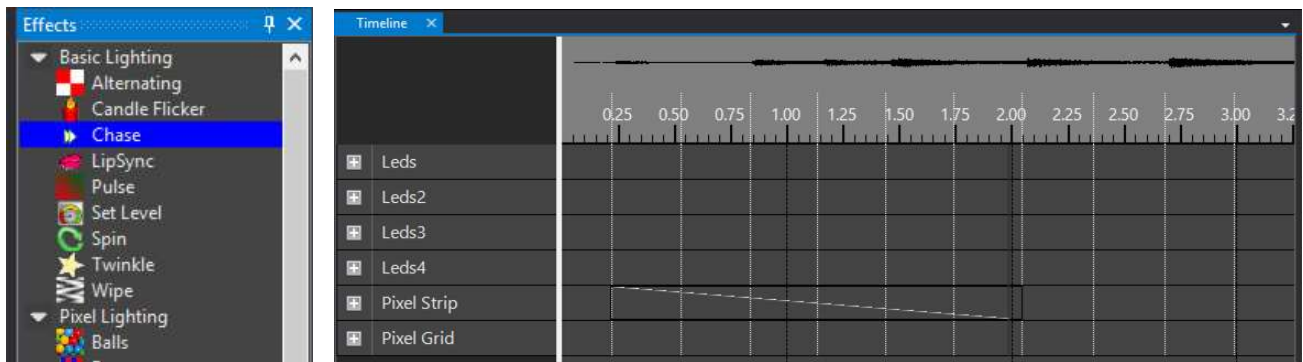


Paso 4.- Vamos a correr un proceso para detectar los Beats de la música, ya que nos ayudará a la hora de armar las secuencias, podrán observar en la imagen las líneas blancas alineadas a los Beats de la música.



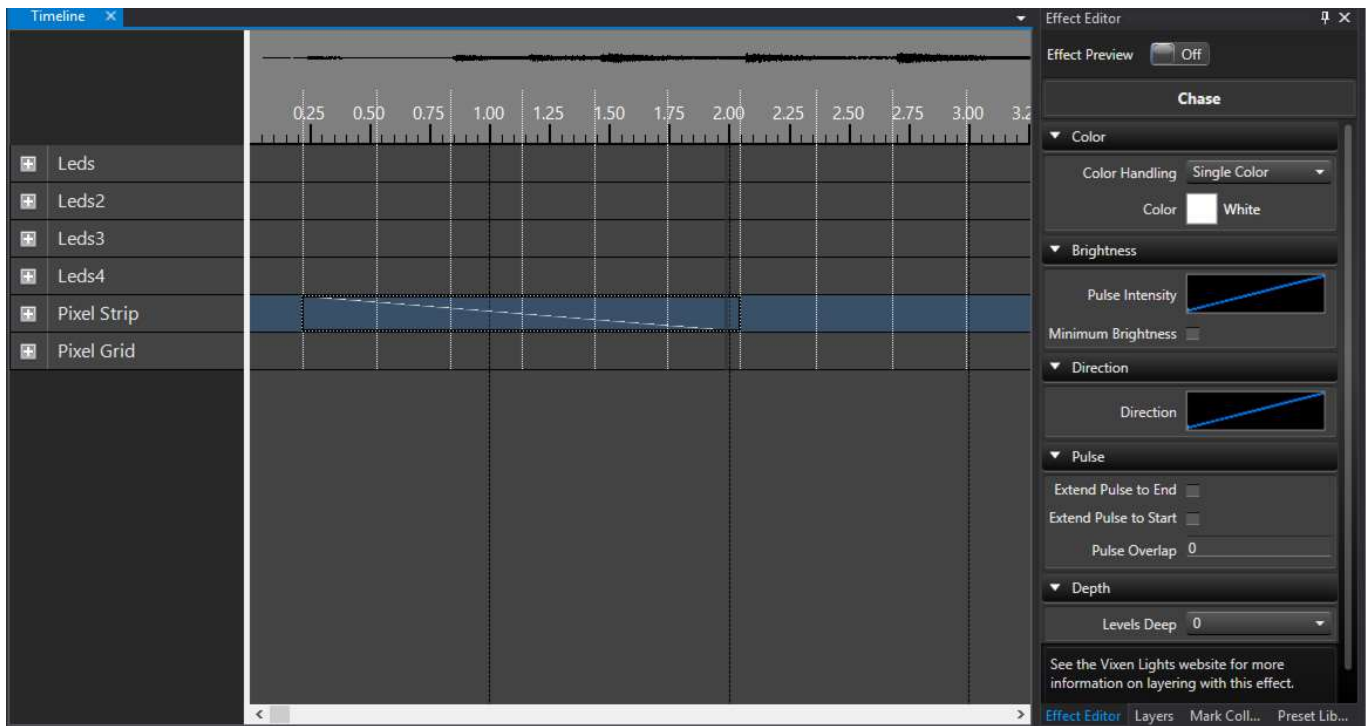


Paso 5.- De nuestro lado izquierdo hay un menú de Effects, Basic Lighting, Pixel Lighting, ambos menús puedes ser utilizados con Pixeles, vamos a dar clic en Chase



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

Paso 6.- 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 7.- Diviértete, hay mucho tutoriales in YouTube.

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