MIDI Color Organ Testing

MIDI to PWM LED Array using World Semiconductor WS2803

Two 18 Channel PWM Constant Current LED Drivers | Three Octave MIDI Driven LED ARRAY | Info on connecting WS2803, LEDs, Arduino

Arduino code
MIDI_WS803 MIDI_WS2801.ino

```
// MIDI_WS2803
// By Thomas Olson
// teo20130301.01
// MIDI IN to WS2803 LED Array
// WS2803 pin4 CKI -> ws2803_clockPin (digital pin 5)
// WS2803 pin5 SDI -> ws2803_dataPin (digital pin 6)
// WS2803 pin2 IREF = Rext to GND
// WS2803 pin6-23 = OUT0-17

// MIDI IN - digital pin 7
// MIDI OUT - digital pin 8

// This program takes MIDI Note Velocitys from Channel 0
// and applies them to a linear LED array where Velocity values are PWM values.

// This test uses two WS2803 in series to represent three octaves.


#include <SoftwareSerial.h>
SoftwareSerial midiSerial(7,8); // instantiate MIDI serial port
```

Shown are two WS2803 18-channel PWM LED constant current drivers connected in series. Since each WS2803 can drive 18 LEDs, two WS2803 conveniently represent three octaves of notes. One WS2803 is driving 18 separate blue LEDs. The other WS2803 is driving a 30W RGB LED module. Each color of the RGB LED is connected to six PWM channels in parallel.

An Atmeg328 or Arduino serial port is used to read the MIDI data into an array. MIDI note data includes velocity data representing 127 levels of how hard a key was pressed.

Three octaves of that array are continuously sent to the WS2803 PWM LED drivers. Since each LED channel of the WS2803 can provide 256 brightness levels the MIDI velocity data is doubled before sending it to the WS2803.

The video first shows all the notes being pressed one after the other, followed by an attempt to show the brightness change when hitting some keys harder and softer.

Also, it can be seen that pressing consecutive notes that are tied in parallel to the RGB LED will make that color brightness increase.