

# Troubleshooting Pixel Systems using the E682-E6804 Pixel Controllers

1<sup>st</sup> release 12/04/2016

This document is intended as a guide to troubleshooting problems with pixel displays using the SanDevices E682 pixel controller. It covers hardware issues, configuration issues, and issues re compatibility with the sequencing software being used. Although much of this information is E682-specific, the non-hardware portion would apply to the E6804 as well, and much of the general information would apply to any controller.

## Hardware Issues

This section will explore potential hardware issues with the E682 and E6804. There are a total of 7 LEDs on the E682, five on the board itself, and 2 on the Ethernet jack. The E6804 does not have the 3 YELLOW LEDs. These LEDs are useful for troubleshooting hardware problems.

### Not all Yellow LEDs are Lit (E682)

The 3 yellow LEDs should normally all be on, assuming that pixel power is wired to both sides of the board. The lower one is labelled "Px1,2" and indicates that power is connected to the left-side pixel power connector, J18. This connector provides pixel power to the 8 left-side pixel outputs. 1/1 through 2/4. If this LED is not lit it means that there is no power at J18.

The middle yellow LED is labelled "Px3,4". When lit it indicates that power is present at J19, the right-hand pixel power connector. This connector powers the 8 right-side pixel outputs AND, in most cases, powers the controller electronics also. If this LED is not lit it indicates that there is no pixel power at J19.

If either of these LEDs is off check your power supply and/or the wiring from the power supply to J18 and J19.

The top yellow LED is marked "3.3v" and indicates the presence of +3.3V power for the on-board electronics of the controller.

If this LED is not lit make the following checks. If you have a voltmeter you can verify that the indicated voltages are correct. All voltages are DC and measured referenced to pixel power ground, ie voltmeter black lead to the (-) side of J19 or J18.

Verify that the power selection jumpers (upper right-hand corner) are correct for the voltage being used. Note that it is possible to use 2 different power supply voltages on J18 and J19. The power select jumpers must ALWAYS be set based on the voltage on J19.

For 5V pixel power, jumper 2-3 only. For 12V pixel power jumper 1-2 and 3-4.

Make sure that the 2A fuse (lower right) is good. A visual inspection is not always reliable, checking with an ohmmeter is more reliable. If this fuse blows repeatedly, and the power jumpers are correct, the most likely cause is a shorted diode, D1. D1 is a black part with a silver band and is located beneath the 3.3v regulator. An ohmmeter can be used to test D1 for a short. With power off, meter on ohms scale, check resistance across D1, then reverse the meter leads and check resistance again. A shorted diode will show near 0 resistance on both tests. A good diode will show a higher resistance on one test and a lower resistance when meter leads are swapped. If D1 is suspected to be shorted, remove it from the circuit by de-soldering or cutting one lead.

NOTE: The controller will operate fine without D1 BUT YOU WILL NOT HAVE PROTECTION AGAINST OVER-VOLTAGE AT THE PIXEL POWER INPUT J19. For example, if you have the board jumpered for 5V pixel power, and you accidentally connect 12V power instead, D1 will protect the controller electronics from damage by blowing the 2A fuse. If D1 is not present and you connect +12V power to a board jumpered for 5V, it will likely damage the controller. So, if you operate without D1 present, use caution.

See if the 3.3v regulator (the part that's bolted to the PC board) is hot to the touch. If it is, it indicates a short circuit of the +3.3V power. This is likely not end-user repairable.

Voltage checks (E682):

	5V Pixel Power	12V Pixel Power	
J19+	+5V	+12V	check power supply and wiring to J19
2A fuse, upper pin	+5V	+12V	2A fuse bad or not properly installed
J21 PIN 3	+5V	+12V	
J21 PIN 4	n/a	+12V	no jumper between pins 3-4
J21 PIN 1	n/a	+5V	bad DC-DC converter (is it hot, if so +5V short)
J21 PIN 2	+5V	+5V	5V no jumper pins 2-3
3,3V regulator, top pin (same as J21 pin 2)			
3.3V reg, bottom pin	+3.3V	+3.3V	regulator hot? +3.3V SHORT

### **All Yellow LEDs lit, but Red and Green LEDs are both off (E682)**

Make sure that the EEPROM jumper is in place across the lower 2 pins (just to the left of the PROG button).

### **Red and Green LEDs not lit (E6804)**

Check 2A fuse.

### **All Yellow LEDs lit (E682), one or both Red/Green LEDs lit (E682 or E6804) but unable to access controller's web page**

Check the LEDs on the Ethernet jack. Make sure that you have an Ethernet cable connecting the E682 to a PC or router.

Generally one LED will be on and one will be flashing. If neither is lit try another Ethernet cable and make sure the other end of the cable is connected to the Ethernet jack on a PC or router. If still no LEDs on the Ethernet jack, or if both LEDs are blinking on and off together, most likely the Ethernet module is defective.

### **All LEDs correct including LEDs on Ethernet jack, but can't access web page (E682, E6804)**

Often this is caused by the controller's IP address not being what you expect it to be, either because it was changed, or possibly corrupted by a firmware issue described below.

Do override 1. To do an override you will press and hold the PROG button. The RED and GREEN LEDs will start flashing on and off together about once per second. Wait until the LEDs come on for the specified number of times, then released the PROG button.

To do override 1, press and hold PROG until the RED/GREEN LEDs come on together for the first time, then release PROG.

Override 1 will blink out the controller's IP address on the LEDs. For each digit the green LED will come on. While it's on, the RED LED will blink out the number value of that digit. If the GREEN LED goes on and off with no flashes from the RED LED, that's a 0.

If the IP address is incorrect, either try accessing the controller at the IP address indicated by the LEDs, or use an override to force a temporary IP address and access the controller at the temporary IP address:

OVERRIDE 2 temp IP address is 2.2.2.2

OVERRIDE 3 temp IP address is 10.10.10.10

OVERRIDE 4 temp IP address is 169.254.74.73 (use this for a direct connection to a PC Ethernet jack)

OVERRIDES 5 thru 15: Temp IP address is 192.168.(override code-5).206 For example, if the controller is connected tyo your LAN and your LAN uses 192.168.1.xxx IP addresses, use override 6 to force a temporary IP address of 192.168.1.206.

Important notes concerning overrides and temporary IP addresses:

A temporary IP address is just that. It is used to allow access to the controller's web page when the controller's saved IP address is not within the range of the local network. A temporary IP address disappears as soon as the controller is reset or when another override is done. For example, you can't use override 1 to verify a temporary IP address because the temporary address will go away as soon as you do override 1.

When you use a temporary IP address, you should access the controller's web page and set the saved IP address to what you want it to be, then click UPDATE SYSTEM INFO. **After the next restart** you can access the controller at the saved IP address.

### **If the IP address shown when doing override 1 is 0.0.0.0 (E682, E6804)**

This is caused by a known firmware issue that affects all version 4 firmware other than 4.070-4.099.

Make sure that there is no active device sending E1.31 or Art-Net packets. Do override 18 followed by override 19 to clear out the controller's configuration memory, then use an override as described above to force a temporary IP address to access the web page. Make sure that all universe assignments are correct (universe selection area in middle of page). Also make sure that your sending software is not sending any Unicast or Art-Net universes to this controller's IP address that this controller is not configured to receive. Access the controller web page and restore the proper configuration settings.

Example: If you are using Unicast or Art-Net modes, and have universe selections set to universes 1 through 12, and your software sends universe 13 to this controller's IP address, it can cause corruption of configuration data, including the IP address. **A firmware update is available to correct this**, but in the meantime the work-around is to insure that your sending software does not send any universes that the board is not configured to receive.

## **IP Address is as expected but can't access web page (E682, E6804)**

Try doing a direct connection to the Ethernet port of a PC or laptop. Make sure that the Ethernet port on the PC is configured for DHCP (this is most common), and restart the PC. Use override 4 on the controller to force a temporary IP address of 169.254.74.73 and try accessing the controller's web page at that address.

Are you using a MAC or IOS device? There is a known compatibility issue with recent MAC/IOS browser versions. CHROME for MAC/IOS will work.

## **Able to Access Controller's Web Page, But Pixels Not Working Properly (E682, E6804)**

A pixel system is like the proverbial chain, many links, all of which must be working properly to get results. The sending software must be properly configured, the controller must be properly configured, and the pixels and power supply have to be good and wired properly.

## **Wiring Pixels to the Controller (E682, E6804)**

Improper pixel wiring is a VERY common error. NEVER make assumptions about which wire is which based on wire color! Color codes change frequently, as though the manufacturer picks whatever colors they have the most of for each run. Pixel strings have an "IN" and an "OUT" end. **The "IN" end MUST be wired to the controller.**

Inadvertently wiring to the OUT end of a pixel string isn't likely to damage anything, it just won't work. Likewise reversing CLOCK and DATA won't hurt anything, it just won't work. **ANY OTHER WIRING ERROR HAS THE POTENTIAL TO DAMAGE PIXELS, CONTROLLER, or BOTH!. In particular the E6804 is not tolerant of a short circuit from a DATA or CLOCK line to + pixel voltage.**

Pixels will have either 3 or 4 wires. Note: Often a pixel string will have two connectors on the input side, a 3 or 4-wire connector and a separate 2-wire connector. The 2-wire connector is another set of +V and GROUND wires and is normally not needed when connecting to a pixel controller since these are duplicates of the +V and GROUND wires in the 3 or 4-wire connector. In the 3 or 4 wire connector there will always be DATA, +V, and GROUND. Some pixels use CLOCK as well. Reversing +V and GROUND can destroy a pixel string! Trial and error is not a good thing. Check with the pixel vendor if uncertain. Often, pixel strips will have the wire identification printed on the edge of the strip. Usually + or +V, - or GND, and DI for Data In, and CI for clock in (4-wire pixels only). If you see DO, that's data out and that means you're at the wrong end of the strip.

Pixel nodes often have wire identification printed on them as well, but it can be very difficult to see. A strip of scotch tape applied to the pixel body will often make it easier to see through the body to make out the legends printed on the circuit board.

## Checking Controller Configuration (E682, E6804)

Each controller can receive up to 7 or 12 universes of pixel data depending on the receive mode selected. 7 universes for Multicast E1.31, 12 universes for Unicast E1.31 or ArtNet. In general, initial setup and troubleshooting is easiest if you use Multicast mode. Also set TIMEOUT to 0.

In the center of the web page you select which universe numbers the controller will receive. By default, controllers are shipped with universes 1 through 12 selected. 1 through 7 are useable for Multicast E1.31. This is fine for most users, but if you have a display that uses multiple controllers, then each controller will likely have a different set of universes.

When setting universe numbers, it is generally easiest to begin with the lowest universe number that controller needs to receive, then fill in the rest of the slots with the next 11 universe numbers in order, even if they will not be needed.

Do not use the same universe number in more than 1 slot as this can cause confusion when setting pixel start addresses.

**Note Re Art-Net Universe Numbers:** E1.31 does not allow a universe number of 0 but Art-Net does. Before the addition of the ArtNet protocol, 0 was considered to be an invalid universe number. For compatibility, Art-Net universes are offset by 1. In other words, an Art-Net sender sending Art-Net universe 0 would be seen by the controller as universe 1.

Verify that each output group has the proper number of outputs enabled, and has the correct length and pixel type.

Be sure to click the appropriate 'SAVE' box after making changes to each output group.

Now turn on test pattern 15 and click "UPDATE SYSTEM INFO". This is an RGB chase (RGB then OFF for newer firmware) of 5 consecutive pixels per color.

If the basic controller configuration is correct, and if the pixels are good and wired correctly, all pixels should be displaying the RGB chase sequence.

Note: The most recent firmware (4.051 thru 4.099) displays this sequence as RED, GREEN, BLUE, OFF. The OFF segment allows you to use this pattern to verify that the RGB color order is correct. If you see a different color order, change the RGB Color Order configuration entry for that output group to get the correct order. Also note that for the color order to be correct, the channel number of the START ADDRESS of each output group must be 1, 4, 7, 10, etc, in other words, some multiple of 3, plus 1. This insures that the 3 channels of a pixel will not be split between 2 separate universes.

Once it is known that the pixels light properly when displaying a test pattern, the next step is to see how they respond to actual lighting data from the sequencing software.

**You must turn the Test Pattern OFF (set to 0) for the controller to respond to lighting data from the sequencing software.**

Three basic things must happen in order to control the pixels from external data.

- 1) The receive mode (Unicast E1.31, Multicast E1.31, or ArtNet) of the controller must match the mode used by the sending software.
- 2) The controller must be configured to receive the universes that the sending software is using.
- 3) Controller pixel addresses must match sending software pixel addresses.

For initial testing or troubleshooting, Multicast mode is selected. It is easiest to configure because the sending software does not need to know the IP address of the controller, as long as universe numbers match, they will talk.

If the sending software is running, when you turn off test patterns you should see pixels respond to the lighting data from the sender.

If that doesn't happen then we need to isolate the problem.

### **Pixels never turn completely off but continue to glow dimly (E682, E6804)**

Check the power supply voltage. Some 12V pixels are very picky about voltage and will act up if the power supply voltage is even slightly high. NEVER increase power supply voltage to compensate for voltage drop.

### **All pixels light bright white even when not receiving data (E682, E6804)**

This is most often caused by an incorrect pixel type selection.

### **4-wire pixels don't work with one or more output groups (E682)**

Insure that the "Balanced Output" jumpers are NOT installed for output groups configured for 4-wire pixels. These jumpers should only be installed for outputs configured for DMX, Renard, or for 3-wire pixels when being used with a balanced line receiver module at the pixel string (typically used for very long wire runs).

### **Some pixel strings fail to operate properly with symptoms such as flickering (E682)**

This is most often caused by degradation of the data signal as it travels down the wire connecting the controller to the pixel string. It can also be caused by timing variations between otherwise "compatible" pixel types. If running firmware version 4.051 and using type 2811 or compatible pixels, try version 4.058 to see if this cures the flicker problem.

When running relatively long wires between controller and pixel, meaning more than 20 feet or so, the data signal from the controller can be degraded to the point where it can't operate the pixels reliably. There is a resistor in series with each DATA and CLOCK output to protect the output IC from damage if the DATAS or CLOCK line is shorted to pixel voltage. The value of this resistor is a tradeoff between wire length driving capability and amount of protection offered to the IC.

The stock value of 270 ohms is conservative. IC failure is unlikely even with a prolonged short of CLOCK or DATA to +12V pixel voltage. For driving longer wire lengths, better results can be obtained by switching these resistors to a lower value. They are pluggable networks and each network controls the 2 outputs immediately beneath it.

### **Some pixel strings fail to operate properly with symptoms such as flickering (E6804)**

There is a jumper on the E6804. This jumper should be removed for normal operation. Some pixels may have issues with this jumper installed.

### **Failure of a single output to light pixels (E682, E6804)**

The fuse is the likely suspect if a single output fails to operate. Each output is fused with a 5 AMP "Mini ATO" style fuse. Replacements are available at any location that sells automotive parts. Please note that for each output group on an E682, the fuse order from left to right is 2-1-4-3, NOT 1-2-3-4. The output number corresponding to each fuse is silk-screened on the circuit board near each fuse. A visual inspection is NOT a reliable indicator of a good or bad fuse. Swapping with a known good fuse, or testing with a voltmeter from presence of + pixel voltage on the pixel connector is

more reliable. Other possibilities include (E682) a bad resistor network or a bad line driver IC. Each network operates the 2 outputs immediately beneath it and each driver IC operates the 4 outputs immediately beneath it. These parts are socketed and user replaceable.

(E6804) single output failure, when fuse is known good, likely requires CPU replacement.

## **No pixels light when receiving data from sequencing software (E682, E6804)**

The first thing to check is to see if packets are being received by the controller. The easiest way to do this is to periodically click the REFRESH PAGE button on the controller's web page and look at the values displayed near the center of the page called "Received Packet Count".

You should see the packet count jump up on every universe that the sequencing software is sending when you refresh the page. Typically you can expect to see about 40 packets per second on each universe.

If you don't see incoming packets then either the sequencing software is not sending any, or it's sending the wrong packet type, or the universe numbers aren't correct. As a test, you can try changing the controller to Unicast mode and see if packets are received then. If so, it means the sequencer is sending Unicast instead of Multicast.

One of the advantages of using Multicast mode for debugging is to be able to use a tool called SacnView. It's a free download, just Google "download SacnView.exe". No install needed, just double-click the sacnview.exe file to run it.

SacnView will show you who is sending E1.31 universes, and what data is being sent. It can also send universes itself.

If the controller isn't receiving packets from your sequencing software, you can try sending a universe with SacnView to see if the controller receives that. Run SacnView then click the red 'antenna' icon to open a universe sender. It will default to Multicast mode and universe 1. Change the universe number if you need to. Select 'sine wave' as the type of data to send (you may need to scroll down the list of buttons) then click START.

Now your controller should receive these packets in universe 1 or whatever other universe you may have selected. Any pixels addressed to this universe should start doing a slow ramp up from off to full white then back to off and repeat.

You should see the packet count for this universe jump when you refresh the page.

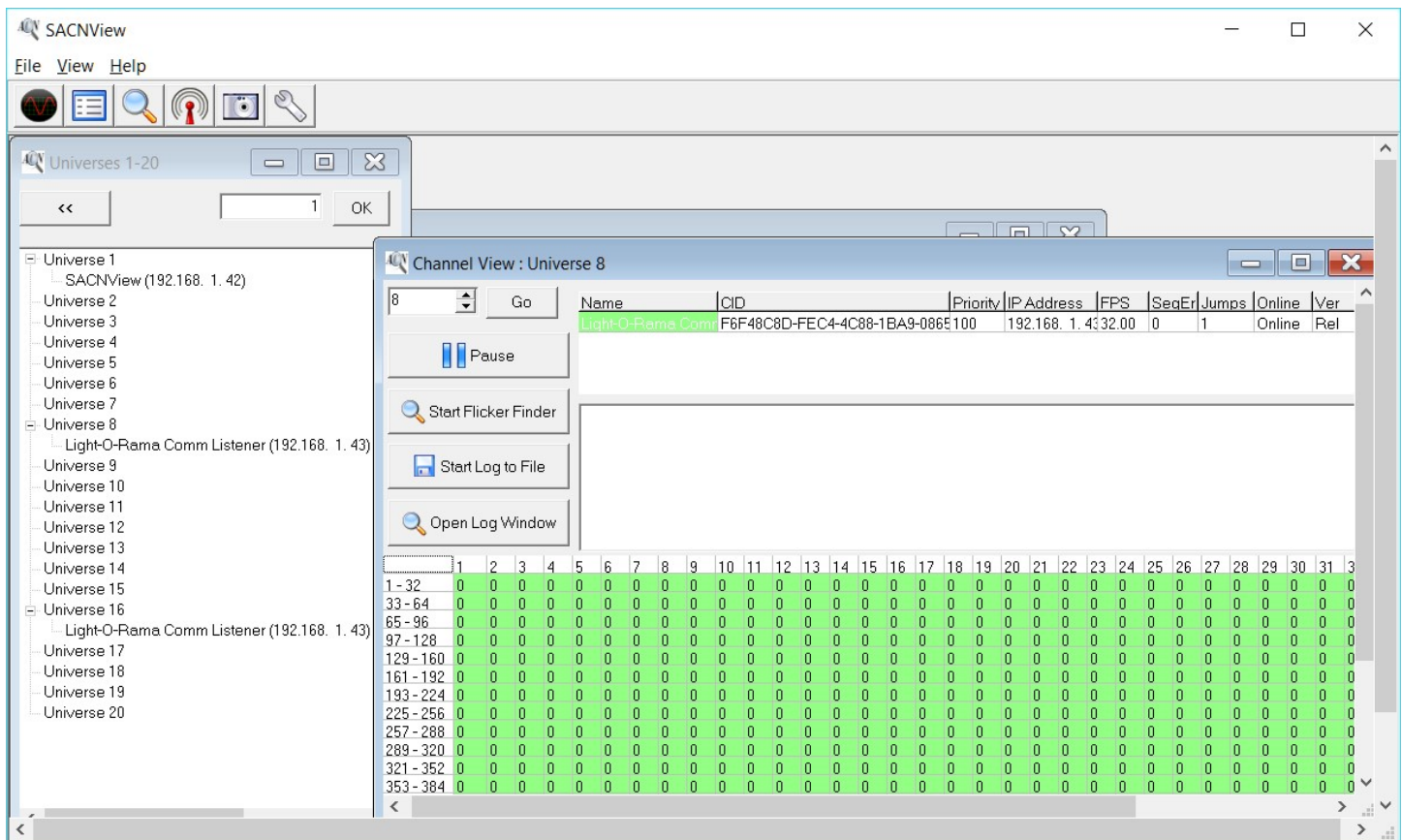
If this test is successful it tells you that the controller is properly configured to receive E1.31 data.

## **Pixels light using SacnView, but not from sequencing software data (E682, E6804)**

SacnView's Universe Viewer feature can be used to display information about who is sending as given E1.31 universe and what data is being sent. Note in the SacnView screen capture below that SacnView sees another instance of itself (on another PC) sending universe 1 sine wave pattern, and sees LightORama on another PC sending universes 8 and 16.

Double-click a universe number in the "Universes 1-20" window to open a Channel View window for that universe. Click the + sign next to a universe number to display the sender info for that group of 8 universes.

In this example, if we were expecting to light our pixels on Universe 8 from LOR, the pixels would not light because LOR is sending all 0s in its packets.



## Pixels light from sequencing software data but not correctly (E682, E6804)

There can be several causes and effects related to this. If the correct pixels are lighting but the colors are wrong, then it's likely either an incorrect RGB color order in the controller configuration or a discrepancy in the start addresses of 1 or 2 channels, perhaps caused by crossing a universe boundary, discussed below.

If the wrong pixels are lighting then it's almost certainly an addressing issue.

**For each string of pixels, the addresses assigned to those pixels by the controller must match the addresses used by the sequencing software.** In the sequencing software this is often a 2-step process, where channels are initially given a sequential channel number, then later those channel numbers are mapped to E1.31 universes.

## Pixel addressing considerations specific to E682 and version 4.xxx firmware

I will explain how addresses are assigned by the E682 firmware because the E682, running version 4 firmware, does have some pixel string addressing limitations. **Because of this, it's easiest to set the addressing the controller first, then set the sequencing software addressing to match.**

In version 4 firmware, E682 outputs are addressed in groups of 4. In other words you specify the start address of the first pixel of the first output of the group, and the controller assigns sequential addresses to all pixels in the group.

Example: Group 1, 4 outputs in use, 50 pixels each. Start address is Universe 1 channel 1 (I will use a-b notation where a=universe number and b=channel number within that universe from 1 to 510), or 1-1.



Output 1/1 gets 150 channels (50 pixels X 3 channels per pixel), so it's assigned channels from 1-1 through 1-150. Output 1/2 gets assigned the next 150 channels, or 1-151 through 1-300. Output 1/3 then gets 1-301 through 1-450. Finally the last output of the group, output 1/4 gets assigned channels 1-451 thru 2-90.

Now two things are happening with that last output. First we are crossing a universe boundary because we have used up all of the available channels in universe 1. Second that universe boundary occurs at channel 510. Even though DMX universes can be 512 channels long, the convention when using RGB pixels is to only use 510 channels to avoid splitting the channels of a single pixel across a universe boundary. This is an important point because you need to make sure that your sequencing software is doing the same, only using 510 channels of each universe. If the sequencing software uses all 512 channels, addressing will get out of sync by 2 channels when a universe boundary is crossed and this will have the effect of making the pixels on the next universe display the wrong colors.

Another consideration with version 4 firmware on an E682 is that there is only one LENGTH parameter that defines all 4 outputs of a group. That's fine for a matrix or megatree where all strings will be the same length, but can be problematic for applications such as outlining windows or rooflines where many different string lengths may be needed.

What has to be done, is to set LENGTH to the length of the longest string in the output group. For example, let's say we have a group with 4 outputs that are 60, 50, 40, and 30 pixels long. Length must be set to the largest value, 60. The controller will assign  $60 \times 3 = 180$  channels to each output as follows, assuming we are using the 2<sup>nd</sup> output group and we start at address 2-91 (this makes our addresses start where the first output group ended):

Output 2/1 will be assigned addresses 2-91 through 2-270 and will use all of them for its 60 pixels.

Output 2/2 will be assigned addresses 2-271 through 2-450 but will actually only use 2-271 through 2-420 for its 50 pixels. Addresses 2-421 through 2-450 are not used.

Output 2/3 will be assigned addresses 2-451 through 3-120 but will only use addresses 2-451 through 3-60 for its 40 pixels. Addresses 3-61 through 3-120 are not used.

Output 2/4 will be assigned addresses 3-121 through 3-300 but will only use addresses 3-121 through 3-210 for its 30 pixels. Addresses 3-211 through 3-300 are not used.

Again we have crossed a universe boundary in this group. The important point though is the addressing gaps created by the varying string lengths on this output group. The addressing in the sequencing software must take this into account so that its addresses will match. This can be done by creating 'dummy' pixel strings to use up the channels corresponding to each of the gaps.

So in our sequencing software we might have something like this:

String 2/1 60 pixels

String 2/2 50 pixels

Dummy string 1 10 pixels

String 2/3 40 pixels

Dummy string 2 20 pixels

String 2/4 30 pixels

Note that we don't need a dummy string following string 2/4 because we can start our next output group at address 3-211 (the last address actually used by string 2/4 +1) so no addresses are wasted.

A couple of things can be done to minimize wasted addresses. First, try to group strings with similar lengths in the same output group. Second, make the shortest string of the group the last output of the group when possible since no addresses are wasted by the last output of the group.

Having said all of that, **version 5 firmware does allow the length and start address as well as color order and pretty much everything except pixel type, to be specified independently for each of the 16 outputs**, making life much easier for folks using differing string lengths.

Once you understand how the addresses are assigned in the E682 (making a list of each output and it's address range may be helpful), then it's a matter of going to the sequencing software and verifying that each element uses the same address range there that it does on the controller.

### **Pixel addressing specific to E6804**

The E6804 allows independent length and address settings for each of the 4 outputs, so it is not subject to the limitations discussed above.