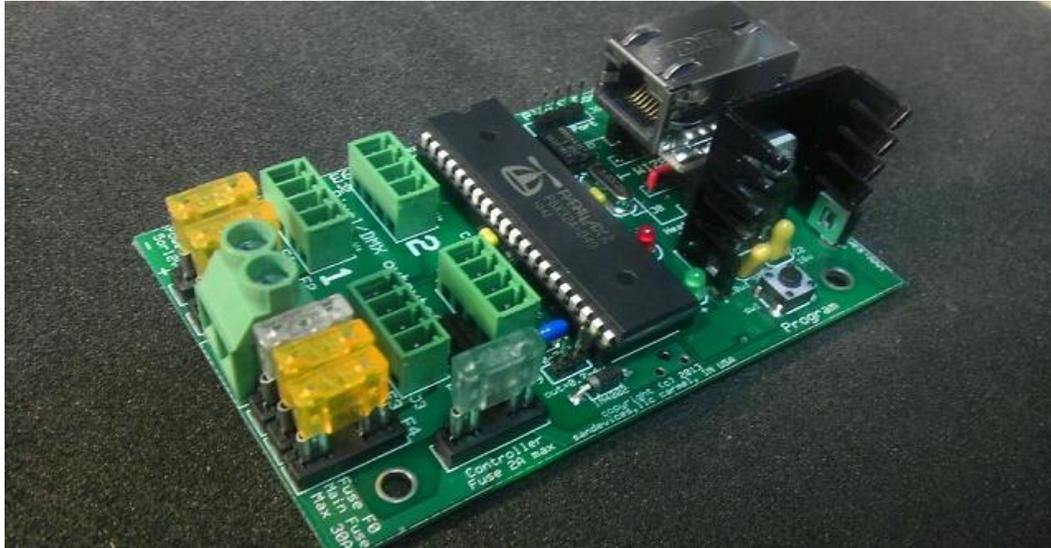




E6804 Pixel Controller Operating Manual



May 19, 2013; First Release

Introduction

The SanDevices E6804 is an evolution of the E68X product line, and is very similar to its predecessors, the E680, E681, and E682. The E6804 is a controller intended to be used as part of a system to operate a lighting display that consists of many individual RGB LED pixels. To form a complete system, one or more E6804s are used together with one or more pixel power supplies, one or more strings or strips of pixels, and a PC equipped with lighting display software that supports RGB pixels and is compatible with industry-standard SACN, also known as E1.31 “DMX over Ethernet” lighting protocol, or with Art-Net.

The E6804 is usually installed near the pixels it controls and their power supply, and acts as the “bridge” between the pixels and the PC. The E6804 receives the lighting intensity signals from the computer via a network (LAN) connection, and converts them into a form suitable for operating the actual pixels.

The lighting control software determines what DMX (intensity) values need to be sent to each color of each pixel. The PC software then forms this data into “packets”, each packet consisting of the current intensity value for each of up to 510 channels (3 channels for each pixel), and sends these packets out over its Ethernet port. The packets travel through your local network and eventually to the E6804 via its Ethernet connection. The E6804 then converts the DMX intensity values into multiple streams of data that are sent on to the various strings of pixels. The controller uses its configuration data, which you define via the controller’s web interface, to know how to reformat each piece of DMX data to route it to the proper pixel in the form that pixel will understand. So the path is:

Lighting Software(PC-Based) -> Network -> E6804 -> Pixels

The E6804 is available in kit form, as a fully assembled and tested board, or as part of a complete system including the E6804, a power supply, enclosure, and cables (the Pixel System 2).

Like all SanDevices products, the E6804 is manufactured in the USA.

Feature Summary:

The E6804 is a single PC board, approximately 2.5"x3.8", the mounting holes are on 2"x3" centers and will accept #6 hardware. This pattern is compatible with the "new style" industry-standard Keptel or equivalent CG-1500 enclosure, or similar enclosures with a 1" grid.

DMX data input is via the SACN protocol, also known as E1.31, or via Art-Net, using a 100mb Ethernet connection. The E6804 presently support up to 76 universes of E1.31 multicast, and up to 12 universes of E1.31 Unicast or Art-net.

SACN (E1.31) and Art-Net are industry-standard protocols that allow up to hundreds of thousands of lighting channels over a single Ethernet cable, and eliminates the need for multiple DMX 'dongles' at the controlling PC.

Note: at the present time Art-Net support for the SanDevices controllers is limited to handling ArtDMX packets. Additional Art-Net features such as remote configuration, and Art-Poll replies, are not supported.

4 On-board individually-fused pluggable screw-terminal connectors for connecting up to 4 separate pixel strings or strips. No soldering is required.

All pixel strings are powered from the E6804, no external pixel power wiring is needed.

The E6804 is extremely versatile, with many programmable options, and many pixel types supported, including 5-bit (software-expanded to 256 dim levels), 7-bit, 8-bit, and 12-bit pixels.

The E6804 can be configured to operate multiple pixel types (up to 4 simultaneously).

The E6804 is configured via a built-in web server.

The E6804 powers itself from the attached pixel power supply, either 5 or 12 volts.

In addition to pixels, any or all of the E6804 outputs may be configured to provide a wired DMX output or a Renard output.

The E6804 supports 8-bit dimming (256 intensity levels) for the LPD6803 pixel type.

Specifications:

Input protocols supported:

- E1.31 Multicast, up to 7 universes (up 1,190 pixels)
- E1.31 Unicast, up to 12 universes (up to 2,040 pixels)
- Art-Net, up to 12 universes (up to 2,040 pixels)

Outputs: 4 connectors for pixel strings, DMX, or Renard.

Power: A 2-wire screw terminal block is provided for connection to the pixel power supply, either 5VDC, or 12VDC.

The E6804 has a large number of programmable options to allow the board to be used in many different configurations. Programmable options are set using a web page. This page also displays operating statistics and the current configuration data.

Supported Pixel Types:

WS2801, WS2811, LPD6803, 180X, GE ColorEffects, TLS3001, CYT3005, 16716, 981X, 880X, Renard (57.6kbps), and Native DMX mode. The E6804 supports 8-bit dimming (256 intensity levels) for the 5-bit LPD6803 pixels.

Designations such as 180X mean that there are several pixel control chips in this family, they are all compatible with the E6804.

Note that many common pixel types are “clones” of others. Some common examples are:

WS2811 = 1804 and CYT3005 = TLS3001

The E6804 is intended for applications that require a pixel controller that is smaller and/or less expensive than the E682.

Mounting the E6804

The E6804 is a single circuit board with dimensions of 2.5” x 3’8”. There are 4 mounting holes on 2”x3” centers. The mounting holes will accept up to a #6 screw. If using metal mounting hardware, take care that the screw heads do not come into contact with PC board traces. The use of nylon hardware is suggested. The layout is such that all wiring can exit from the left side of the board. It is the user’s responsibility to provide a suitable enclosure for the E6804. The mounting hole pattern is command with many industry-standard enclosures such as those manufactured by Keptel.

Test Jumper

There is only one jumper on the E6804. It should normally be left in the ‘off’ position. The jumper should be installed when doing voltage tests on the E6804.

Fuses

The E6804 has 6 fuses. They are standard Mini_ATO fuses, and replacements can be found at locations that sell auto parts. Each pixel output has its own fuse, these are the yellow 5A fuses. There is also a Main Fuse, typically 25A, and a 2A fuse that protects the controller electronics.

Electrical Connections

Ethernet to PC or LAN

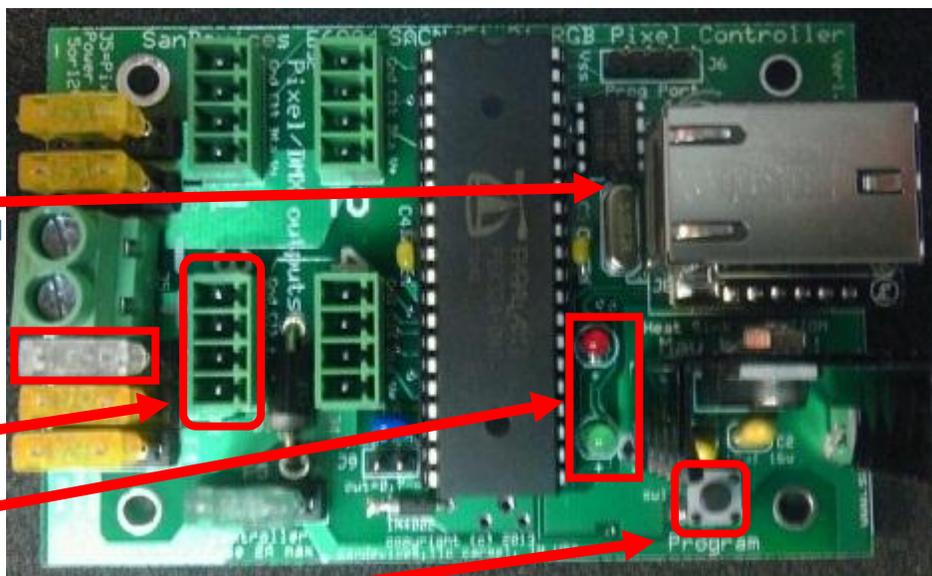
Pixel Power Input
5VDC or 12VDC.

Main Fuse

Pixel Connector #3

LEDs

Program Button



There are 3 types of connections to be made to the E6804:

Power Power to operate the E6804 and the attached pixels should be connected to the 2-position green screw terminal block at the left edge of the E6804. Pixel power may be 5VDC or 12VDC. The attached power supply should be capable of powering the full number of pixels to be attached to the E6804. As a general rule, figure about 3 pixels per watt.

Important: The power supply voltage MUST match the voltage requirements of the pixels to be used. Connecting a 12V power supply to 5V pixels WILL destroy them! The polarity (+ and -) designations are silk-screened on the E6804

immediately behind the terminal block. **ALWAYS INSURE THAT THE PIXEL POWER SUPPLY IS PROPERLY WIRED, + to +, and - to -.** While the E6804 does have circuitry built-in to protect against reversed-polarity power, this is not guaranteed to be successful in all cases. Please take the time to double-check that power supply polarity is correct.

Network The E6804's connection to the outside world is by Ethernet, using a standard CAT5 or CAT6 network cable. The Ethernet connection is used to send information to the E6804 to enable it to control the attached pixels (E1.31 or Art-Net) and is also used to configure the E6804 via its built-in web server. The E6804's Ethernet port will usually connect to your LAN (typically to an unused port on a router), but in some cases may connect directly to the controlling PC or to an Ethernet switch.

Pixels The pixels connect to the 4 4-position green pluggable terminal blocks on the E6804. These terminal blocks provide power to the pixel strings as well as data. The function of the 4 pins is silk-screened on the E6804 immediately behind each connector. The 4 pins are:

- Gnd The (-) side of the pixel power supply, also GROUND.
- Dta The Data signal to the pixels. All pixel types use the Data signal.
- Clk The Clock signal, only used by some pixel types.
- V+ The (+) side of the pixel power supply (fused) that provides power to the attached pixel string.

Note that when using an output type of DMX or Renard, the function of the Dta and Clk pins is different: Dta becomes DATA+, and Clk becomes DATA-. This allows for the connection of standard 2-wire "balanced" DMX or Renard devices.

The user must supply the wiring between the start of each pixel string and the E6804. It is generally recommended to use 18 gauge 3 or 4 conductor stranded wire (4-conductors are needed if the pixels need the CLOCK signal).

Note: Pixel string color codes vary tremendously. It is the responsibility of the user to insure that the proper connections are made between the pixel strings and the E6804. An incorrect connection or a short circuit could damage the pixels or the E6804.

Here are some guidelines for determining which wire has which function on a pixel string or strip.

Often the function is silk-screened on the pixel or strip. Common designations are GND (ground), +V, Vin, +5V (+power), Cin, Cout, Din, Dout, Clk, Dta.

The first step is to identify which end of the string is "IN" and which is "OUT". **The "IN" end of the string must connect to the controller.** If you see labels such as CI or DI, that is the "IN" end of the string. Likewise CO or DO would indicate the "OUT" end. Often there will be extra wires for +V and/or ground. This can be determined by visually checking to see if 2 wires go to the same point on the pixel or strip. These extra wires are not needed and should be taped to prevent an accidental short circuit.

Proper identification of the + and Ground wires is critical. A reverse polarity connection can easily destroy the pixel string. IF YOU ARE THE LEAST BIT UNSURE AS TO THE PROPER WIRING OF YOUR PIXEL STRINGS, PLEASE CONTACT THE PIXEL VENDOR FOR ASSISTANCE.

Incorrectly wiring pixels, either a short circuit or improper wire identification, is the single greatest cause of pixel damage. Don't guess! Incorrect power supply voltage is #2!

When using outputs as DMX or Renard, the power output may not be needed. For DMX and Renard you should always wire GROUND, DATA+ (Dta) and DATA- (Clk).

Initial Startup and Testing

For initial testing of the E6804 only 2 connections are required. You must connect a power supply that can provide between 5 and 12 VDC at a minimum of 300ma to the pixel power connector, and you must connect the E6804 to your LAN using a standard network cable. This is sufficient to allow you to access the E6804's web page. **It is recommended to do the initial testing without attaching any pixels.** In order to test with pixels you will need a power supply capable of powering the number of attached pixels as well as 1 or more pixel strings or strips.

A few seconds after applying power, the GREEN and RED LEDs will light. If not, please double-check that the power supply is correctly wired and provides the proper voltage. You should also see the GREEN and YELLOW LEDs on the Ethernet jack are lit (solid or flashing) if you have a working network connection.

The next section of this manual deals with the configuration of the E6804 using the built-in web server. This documentation is specific to firmware revision 4.026.

You will use the E6804's web page to configure controller. Among other items, you will tell it which outputs you are using, what type of pixels are connected to each output, how long each pixel string is, what color order the pixels use, and which addresses each pixel string should respond to.

Accessing the E6804 Web Server

This discussion assumes that the E6804 controller is connected via a LAN cable to your LAN, typically to an unused port on your router.

All configuration of the E6804 controllers is done via a web configuration page. The web page is displayed by typing the IP address of the controller into a web browser's address bar. In order to access the web page the current controller IP address must be known, and it must lie within the address range of the LAN to which the controller is connected.

The initial access to the web page presents a bit of a chicken-or-the-egg dilemma. In order to set the controller's IP address we need to be able to access the web page, but we may not be able to access the web page if the existing IP address isn't known, or isn't valid for our network.

All controllers ship with a default IP address of 192.168.1.206. Almost all small networks will use some variant of 192.168.x.y IP addresses. Of those, the 192.168.1.x range is the most common. Since the 192.168.1.x addressing scheme is common, it's usually easiest to first try accessing the controller at its default IP address of 192.168.1.206.

To do that, open your web browser and type 192.168.1.206 into the command bar (where you typically see the http:// addresses), and press ENTER. If you see the controller's web page displayed, then you can skip ahead to the next section, where you see an image of the web page.

If you're here, the 192.168.1.206 address didn't work. The most likely cause is that your network uses an address range other than 192.168.1.x. You can determine that with this simple procedure. Note that this is a one-time procedure. Once you've overridden the default IP address with one that's compatible with your network, we'll change the E6804s IP address to be one that's valid for your network to make future access to the web page simple.

The following procedure is good for Windows 7; other versions of Windows will use a similar procedure. In the text that follows, bold and capital letters are used to identify what you need to type or which screen button to 'click', you need not use caps when actually entering the commands:

Press the **WINDOWS** button to open the start menu.

Type **CMD** in the search bar and press **ENTER**.

A black window should open. In the black window type **IPCONFIG /ALL** then press **ENTER**. (That's IPCONFIG then a space, then a forward slash followed by the word ALL.)

Scroll up if necessary using the scroll bar to display the first part of the text that has appeared in the window. Look for an area called "**Ethernet Adapter Local Area Connection**". Within that area look for the line "**IPv4 Address**". Generally the number displayed will be:

192.168.x.y, where x will usually be 0 or 1, (sometimes higher) and y can be any number. **The third digit, X in the example, is what you need to know.** If X is 0, your LAN uses 192.168.0.X addressing. Other common IP ranges would be 2.X.X.X or 10.X.X.X. (The 169.254.X.X address range will be used if your PC is not connected to a router or is connected to a wireless router.)

Now that you know the address range used by your LAN, you can override the factory default IP address of the E6804 and force it to have an IP address that's within the range of your LAN. This is done by using the **PROGRAM** button on the E6804.

Using the information about your LAN's address range, pick the override you need to use from this table:

<u>Over-Ride Code</u>	<u>Purpose</u>
1	Display current IP address on LEDs, continue until restarted
2	Force a temporary IP address of 2.2.2.2

3	Force a temporary IP address of 10.10.10.10
4	Force a temporary IP address of 169.254.74.73
5	Force a temporary IP address of 192.168.0.206
6	Force a temporary IP address of 192.168.1.206.....

Note: Codes from 5 to 15 all force a temporary IP address of 192.168.X.206, where X is the over-ride code – 5. For example, over-ride code 15 forces a temporary IP address of 192.168.10.206, because 15-5=10.

As an example, if you determine that your LAN uses 192.168.0.x addressing, then you would use override 5 to force an IP address of 192.168.0.206 because that is within the address range used by your LAN. If your IP address is of the form 2.X.X.X or 10.X.X.X or 169.254.X.X you would use over-ride codes 2, 3, and 4 respectively.

Overrides are done by use of the **Program** button on the E6804. Use this procedure:

Determine the override code you need to use per the discussion above, it will be in the range of 2 to 15.

Once the controller is powered up and the LEDs are lit, press AND HOLD the **PROGRAM** button.

After about 2 seconds you will see the RED and GREEN LEDs flash on and off together at the rate of one flash per second. Count the flashes and release the PROGRAM button when the lights flash on for the number of times equal to your override code. With our previous example, a LAN using 192.168.0.X addressing, we determine from the table that we need override 5. So we would hold down PROGRAM until the LEDs come on for the 5th time, then release it.

After you release the **PROGRAM** button the LEDs will come on and stay on. Now you should be able to enter the override address into your browser's address bar and you should see the controller's web page.

If you are unable to use the above procedure successfully, there is an alternate method to access the controller page. Unplug your computer's LAN cable from your router and wire it directly to the Ethernet jack on the controller. **Reboot the computer** and then use over-ride code 4 on the controller. You should then be able to access the controller's page at 169.254.74.73.

Note: The IP address created by the over-ride is temporary. Once you are able to access the controller's web page, change the IP address to an address that is within the range of your LAN, save by clicking **Update System Information**, then restart the controller.

There is an additional use of the override; that is to display the current IP address of the controller. This is useful if, for whatever reason, do not know the IP address that the controller is set to. You can use override #1 to display the IP address on the LEDs.

Over-ride code 1, IP address display

Use over-ride code 1 to determine the IP address of the controller when it is not known. Press and hold the PROGRAM button until the red and green LEDs come on together for the first time, then release (over-ride code 1). The LEDs will now display the current IP address. For each digit, the green LED will come on, then the red LED will flash a number of times corresponding to the value of that digit. Then the green LED will go off. This process repeats until all 12 digits of the IP address have been displayed, and the entire process repeats continuously until the controller is restarted by pressing the PROG button again, or by pressing the RESET button (all but E6804), or by interrupting power to the controller. For the factory default IP address of 192.168.1.206, you would see the following pattern on the LEDs:

Brief pause with both LEDs off, then green ON, then red flashes 1 time, then green off (1)

Brief pause with both LEDs off, then green on, then red flashes 9 times, then green off (9)

Brief pause with both LEDs off, then green on, then red flashes 2 times, then green off (2)

(slightly longer pause, then next group begins)

Once the current IP address is known (factory default is 192.168.1.206), if it is within the address range of your LAN just enter that IP address in your browser's address bar. If it is outside of the range of your LAN use the override procedure described above.

Once you have successfully accessed the E6804's web page, you will see a screen similar to the following:

SanDevices SACN/E1.31/Art-Net RGB Pixel Controller Model E6804

System Information:

IP Address	Subnet Mask	MAC Address	Up-Time	Receive Mode	Timeout	Test Pattern	Gamma Value	
192 168 1 206	255 255 255 0	4A:49:4D:B0:B9:ED	0000:00:16	<input checked="" type="radio"/> Multicast E1.31 <input type="radio"/> Unicast E1.31 <input type="radio"/> Art-Net	0	0	1.0	Update System Information

Firmware Version: 4.026 Firmware update status: None Tried [Update Firmware](#)

Universe Selection and Packet Statistics:

Universe	Multicast, Unicast or Art-Net Universes							Unicast or Art-Net Only Universes					Update Universe Numbers
	1	2	3	4	5	6	7	8	9	10	11	12	
Packets Received	428	428	427	0	0	0	0	0	0	0	0	0	0
Sequence Errors	5	5	5	0	0	0	0	0	0	0	0	0	0
Invalid Packets	0	0	0	0	0	0	0	0	0	0	0	0	0

Output Configuration:

Outputs	Outputs In Use	Output Type	Length Pixels	Group Size	Color Order	Start Address Universe	Start Address Channel	End Address Universe	End Address Channel	Reverse	Zigzag Every	Null Pixels	Refresh Rate	
1	<input checked="" type="checkbox"/>	WS2801	50	1	RGB	1	1	1	150	<input type="checkbox"/>	0	0	212	Update
2	<input checked="" type="checkbox"/>	GE ClrE#	50	1	RGB	1	151	1	300	<input type="checkbox"/>	0	0	20	Update
3	<input checked="" type="checkbox"/>	1804/2811	50	1	RGB	1	301	1	450	<input type="checkbox"/>	0	0	245	Update
4	<input checked="" type="checkbox"/>	TLS3001	50	1	RGB	1	451	2	90	<input type="checkbox"/>	0	0	81	Update

[REFRESH PAGE](#)
[RESTART CONTROLLER](#)
0 [SYSTEM COMMAND](#)

This discussion pertains to firmware version 4.026 Version 4 firmware replaces the old command line interface used in earlier firmware versions with a more modern interface that uses HTML form elements including drop-down lists, buttons, and check boxes. Overall the interface is much more intuitive and the user no longer has to rely on a search of the documentation to learn the purpose and/or syntax of various commands.

In general, you will make configuration changes by entering the desired value(s) using entry boxes, checkboxes, buttons, and drop-down lists. Then you will click the corresponding **UPDATE** button for the changes to take effect. Please note that there are several **UPDATE** buttons; **when you make changes in one area of the screen, you must press the proper UPDATE button, before moving on to make changes in another area.** In particular, each pixel output has its own configuration line and its own **UPDATE** button. So when configuring outputs, you would make all changes to output 1, then click that line's **UPDATE** button. Then move on to output #2 and repeat the procedure.

The web page is divided into 3 main sections: **System Information**, **Universe Selection**, and **Output Configuration**.

System Information

The **System Information** section allows the user to view and/or change the following system settings:

IP Address: This is the IP address that the controller will have **when it is not over-ridden at startup**. This IP address is used to access this web page, and is also used as the destination address if using E1.31 Unicast mode or Art-Net (described later). The IP address is entered as 4 separate numeric values. The controllers are shipped with a default address of 192.168.1.206. If your LAN uses 192.168.1.x addressing, you should be able to access the controller simply by entering that IP address into the URL bar of your browser. If your LAN uses a different address range, there is an override mechanism, described earlier in this manual that allows you to use the **PROGRAM** button on the controller to force a different IP address at startup. Typically this is used to force an IP address in the range of your LAN, then you can access the web page, and change the IP address so that is within the address range of your LAN. To change the IP address simply click on all of the 4 IP address boxes and enter the desired values. When finished you must click the **Update System Information** button to save your changes. In order for the new IP address to take effect, you must then restart the controller, by pressing the **RESTART CONTROLLER** button at the lower left. Note that **ONLY an IP address change requires a restart**, all other changes take effect immediately when the corresponding **UPDATE** button is pressed.

If you used the override procedure to access the controller, you should change the IP address now to one that is compatible with your network so that you can access the controller in the future without using the override. You will see a line at the top of the screen that shows the temporary IP address that is in use because of the override:

System Information:

Temporary IP Address is: 192.168. 1.206

In many cases this will be a good choice for the permanent IP address. You would want to use a different address if your LAN already has another device at this address, such as another pixel controller. In general, just use the next consecutive address. Enter

*the 4 parts of the desired IP address in to the 4 **IP Address** boxes. Verify that all parts of the IP address have been entered correctly. Once they are all correct, click on the **Update System Information** button. Wait a few seconds then click the **Restart Controller** button. Wait a few more seconds until both LEDs come back on, then you should be able to type the new IP address into your browser to regain access to the controller's web page.*

The controller's **Subnet Mask** is shown next. This is set automatically by the controller based on the IP address in use.

The controller **MAC Address** and controller **Up-Time** items are display only items, and cannot be changed. **Up-Time** simply shows the elapsed time since the controller was last restarted.

Receive Mode Receive Mode is an item that can be changed by clicking on the appropriate button to change the receive mode from Unicast E1.31 to Multicast E1.31 or Art-Net. At this time Art-Net support is limited to reception of ArtDMX packets directed on the controller's IP address. ArtPollReplies are not supported. In general, it is suggested to use Multicast E1.31 as your first choice. The other modes are more specialized, and will be described later.

Timeout may be set to any value in the range of 0-99. If non-zero, and if incoming data stops, all of the controller outputs will be turned off after the selected number of seconds of delay. If set to 0 this feature is disabled.

Test A test pattern may be enabled by setting a value greater than 0. A value of 0 disables test patterns, a non-zero value enables a test pattern. **When test patterns are enabled the controller will not display any received data. When test patterns are enabled the web page will take longer to render, so it is suggested to keep test pattern turned off while working with the web page.**

Test Patterns: 1-3 All pixels lit RED, GREEN, or BLUE respectively

- 4-6 A bright pixel chases from start to end leaving dim pixels behind, RED, GREEN, and BLUE
7-10 Not used
11-29 A pattern of n RED pixels followed by n GREEN pixels followed by n BLUE pixels that chases through the entire range of pixels, where n is the test pattern number – 10. **For example, test pattern 15 is a group of 5 pixels of each color (15-10=5).**

(Please note that the colors you observe may be different than those listed, this simply means that your pixels don't use the standard R->G->B color order. Not to worry, this can be fixed later.)

Gamma Value may be set to any value in the range of 1.0 (no correction) to 3.0 (maximum correction) in increments of 0.2. Gamma correction is used to improve the dimming characteristics of some pixel types to make them have a more natural-looking dimming characteristic, more gradual at the low end similar to the dimming characteristics of incandescent bulbs.

After making any changes to any items in the **System Information** area, you must click the **Update System Information** button for the changes to take effect. **With the exception of a change to the IP address, all changes become effective immediately and do not require a restart of the controller.**

Universe Selection and Packet Statistics

This area is where you define the list of universes that the controller will respond to. The set of universes needed will depend on how your pixels or other devices are defined in the software package that sends out the E1.31 packets. This could be LOR, LightShowPro, Madrix, etc. **Later in this manual there will be a discussion that explains many of these terms, and talks about the design of a pixel system. If you're new to pixels, you can skip over the information here that you're not familiar with, and then after reading the material later in the manual you will have a better understanding of how these configuration settings work.**

Based on the settings that you use to address your pixels in your sequencing software, you will know how many universes, and which universes, need to be received by this controller. Typically the universe numbers are entered in order from lowest to highest. Unused universes may be left at 0. Valid universe numbers are 1-63998. There should be no duplications in the universe numbers used, in other words **don't enter any universe number in more than one box.** Universe numbers are entered simply by clicking on the appropriate entry box and entering the desired value. When finished, click the **Update Universe Numbers** button to save the changes. Universe number changes become effective immediately. **Note that the selections here simply determine which set of incoming universes the controller will receive. In the next section we will assign those universes to control specific pixel strings.**

This section of the page contains 3 additional items of information for each universe: **Packets Received, Sequence Errors,** and **Invalid Packets.** These are display-only items. Packets Received lists the total number of packets received for this universe since the controller was restarted. **Sequence Errors** shows the number of packets that were missed, or skipped over. Typically this value will be rather small, but not necessarily 0. For example, the controller will tend to miss some incoming packets when the web page is being updated, although this is typically not visible because the packets come in so quickly. **Invalid Packets** indicates the count of packets that were not proper E1.31/SACN packets. This should rarely have a non-zero value.

Important: The first 7 universes can be used for pixel control data of any type: Multicast E1.31, Unicast E1.31, or Art-Net. The last 5 universes cannot be used for Multicast E1.31. If you use Multicast E1.31 you must only assign the 1st seven universes to your pixel strings (see Output Configuration, below).

Output Configuration

This is where the individual controller outputs are configured. Outputs are labeled 1 through 4, matching the numbers on the E6804 itself. Each output has its own **Update** button which must be clicked for the changes to be saved and to take effect. Changes take effect immediately when the Update button is clicked, no controller restart is needed.

Remember that changes will not take effect until the appropriate Update button is clicked, and will not be saved if a different Update button is clicked. So, when configuring outputs, configure all of the items for one output then click that Update button to save those changes. Then move on to the next output and repeat the procedure.

Outputs In Use This box is checked if the corresponding output is in use. Outputs that are marked as in use will be assigned a range of DMX addresses if the specified Length in Pixels is greater than 0. This box should be checked for each pixel output that will be used.

Outputs	Outputs In Use	Output Type
1	<input checked="" type="checkbox"/>	WS2801
2	<input checked="" type="checkbox"/>	LPD6803
3	<input checked="" type="checkbox"/>	WS2801
4	<input checked="" type="checkbox"/>	GE ClrEff
		1804/2811
		16716
		LPD880x
		981x
		TLS3001
		DMX
		RENARD57k

REFRESH PAGE

RESTART CONTROLLER

0 SYSTEM COMMA

Output Type is a drop-down selection list that allows you to choose the type of pixel (or other device) attached to this output. As of this time the following pixel types are supported: 6803, 2801, GE Color Effects, 1804/2811, 16716, 880x, 981x and 3001. Non-pixel output types supported are Renard (57.6kbps) and DMX.

Length in Pixels is where you enter the number of pixels that make up the pixel string or strip connected to this output. For output types **Renard** or **DMX**, the value entered here would be the number of channels of output desired divided by 3, since each 'pixel' uses three channels. As an example, for a full universe of DMX output, (actually 510 channels), you would specify a length of 170. This is because 170 pixels would use 510 channels.

Group Size allows pixels to be controlled in groups rather than individually. This feature would typically be used either to reduce the number of channels needed or to simplify programming of the pixels. When the group size is other than 1, instead of pixels being controlled individually, groups of pixels are controlled. For example, with a string length of 50 pixels and a group size of 5, the string would be controlled as 10 groups of 5 pixels and would use only 30 channels (10x3) instead of 150 channels (50x3).

Color Order is a drop-down that allows the specific color order of the pixels connected to this output to be specified. This can be helpful if the particular pixel string or strip does not use the standard R->G->B color order. If the pixels do not light with the expected colors because they use a color order other than RGB, changing the color order to the proper sequence (often by trial and error), will correct the problem.

Group Size	Color Order	Start Address	
		Universe	Channel
1	RGB	5	1
1	RGB	6	91
1	RGB	8	1
1	<ul style="list-style-type: none"> RGB RBG GRB BRG GBR BGR 	9	1

Start Address Each output that is marked as in use, and has a length greater than 0, is assigned a sequential block of DMX addresses. The length of this address block is (Length in Pixels) X 3, because each pixel requires 3 channels, one for each of its primary colors; red, green, and blue. Note that if pixel grouping is used the number of channels needed will be reduced accordingly.

The operator can configure the starting address for each output. All pixels on that output are then assigned consecutive addresses beginning with the specified start address. The necessary channels may span across one or more universe boundaries, depending on the start address and the number of channels needed. When the

end of a universe is reached (channel 510 is the last channel used in each universe), channels will be assigned from the next selected universe based on the order of universe selection (see Universe Selection and packet Statistics).

In most cases the selected universes will be assigned sequentially. For the first controller the starting universe will often be Universe 1. If using Multicast for example, the first controller would typically be configured for universes 1 through 7.

If a different group of universes is selected, this will affect how channel addresses are assigned. For example, assume that the first 5 selected universes (from left to right) are 1, 3, 5, 7, and 9. If we define output output 1 as having 200 pixels, then this output needs 600 total channels. Assume that we start this output at Universe 1 Channel 1. Since there are only 510 useable channels per universe, the last 90 channels will be assigned from Universe 3, since Universe 3 follows Universe 1 in the universe selection list.

The selection of the start address for an output is done by entering the starting Universe (from a drop down list which will only show the list of available universes), and a starting channel, a numeric entry from 1 through 508.

Although not specifically required, it's best to start every output on a channel number of 1 or 1 plus a multiple of 3 (4, 7, 10...508). The example below reflects the selected universe group consisting of universes 5 through 16.

Color Order	Start Address		End Address	
	Universe	Channel	Universe	Channel
RGB	5	1	6	90
RGB	6	91	7	180
RGB	7	1	8	150
RGB	9	1	9	510
	10			
	11			
	12			
	13			
	14			
	15			
	16			

You can see in this example, that a starting address of Universe 5 Channel 1 produced an ending address of Universe 6 (the next universe in line) channel 90. This is because the output needs 600 total channels, 510 from Universe 5 and 90 from Universe 6. The ending channel is calculated automatically by the controller, only the starting address can be entered.

Note: Start Address and End Address values are NOT displayed for any output that is not enabled, or has a length of 0.

Reverse	Zigzag Every	Null Pixels	Refresh Rate	
<input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	212	<input type="button" value="Update"/>
<input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	20	<input type="button" value="Update"/>
<input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	245	<input type="button" value="Update"/>
<input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	81	<input type="button" value="Update"/>

Reverse The Reverse check-boxes allow you to indicate that the associated string is 'backwards', in other words the first pixel to light will actually be the last pixel of the string. An example where this feature would be useful is a roofline, where the controller is mounted at the mid-point, with one pixel string (say #1) running to the left and another (say #2) to the right. Without the reverse feature, if you began lighting pixels in order, you would actually be lighting pixels

beginning at the center of the roof, working left to the left end, then jumping back to the center and then working to the right. By selecting the check-box to indicate that the first string is reversed the pixel sequence will be as it should be, from left to right with no jumps.

Zigzag The Zigzag feature is another feature that can simplify the pixel programming by allowing the pixel addresses to flow in a more logical order. As an example, say you have a matrix of 200 pixels arranged in 20 columns of 10 pixels. If you start at the bottom left you would string pixels 1-10 from bottom to top in the first column, and then pixels 11-20 from top to bottom in the 2nd column, and so on, basically zig-zagging up and down. When done, if you lit pixels in order from the lowest address to the highest, you would see a pattern that started at the lower left and zig-zagged up and down across the matrix. When programming the pixels, it's often easier if the pixels light in a more natural order. In this example, we would use the zigzag feature to tell the controller that the pixels reverse direction every 10 pixels. Now, lighting sequential pixels produces the desired effect, the first column lights from bottom to top, then the 2nd column from bottom to top, etc.

Null Pixels Null pixels are pixels which are ignored by the controller and never lit. The most common use of null pixels is to allow a longer length of wiring between the controller and the start of a pixel string. Because the pixel control signals aren't designed to travel over long distances, it may not be possible to use wire runs longer than about 20 feet. If the particular installation requires a longer wire run than normal, this can often be accomplished by inserting one or more extra pixels in the wire run between the controller and the pixel string. Each of these pixels will regenerate the control signals, enabling them to be run for another 20 feet or so. As an example, if we needed to have a 100 foot wire run between the controller and a pixel string, we might use 4 null pixels, one at 20 feet, 40 feet, 60 feet, and 80 feet. Even though the total run was 100 feet the maximum distance between pixels is only 20 feet because of the null pixels.

Another use for null pixels is when you don't need the full length of your string. Say you have a 50-pixel string but only need 48 pixels lit. If it's more convenient to have the unused pixels at the start of the string rather than at the end, set the string length to 48 and set null pixels to 2.

Refresh Rate This is a displayed value that shows the approximate refresh rate of the pixels connected to this output. It is affected by pixel type and string length.

Special Purpose Buttons

There are 3 special buttons at the bottom of the page, and one number entry box.

Refresh Page simply refreshes the page without changing any configuration. Since the web page never updates by itself, you would use the **Refresh Page** button, for example, to see updated packet statistics, or updated system up-time.

Restart Controller simply does a reset of the controller, the same as would happen if you removed then reapplied power. It would most commonly be used after a change of the IP address, since an IP address change does not become effective until the controller restarts.

The **System Command** button is used to perform some infrequently used functions, based on a numeric value that is entered into the number box to its left. There are 3 system commands presently defined:

- 10 Erase system configuration memory. This will wipe out all configuration data with the exception of the IP address.
- 20 Set test default configuration. This function sets all outputs for type 2801 pixels, 50 pixels per string.
- 30 Set as-shipped defaults. This sets 4 different pixel types on the 4 outputs.

As an example, to return a board to the factory default configuration (other than IP address) you would enter 10 in the system command box and **click System Command**. Then enter 20 and click **System Command**. Then enter 30 and click **System Command**.

What exactly is a Pixel, and some Pixel System Design Considerations

A pixel system is just that, a system. As a minimum it consists of some sequencing software. If you're background includes experience with non-pixel displays you're probably familiar with sequencing software. It's a program, usually running on a PC, where you design your "show". Often a combination of lights and music, but sometimes lights alone. This is where you define what lights (or pixels) will be lit during different segments of your show.

If you're familiar with sequencing software then you're also familiar with the concept of channels. For non-pixel displays, a channel is assigned to each display element; perhaps a string of lights, or a flood light, or a lighted fixture such as a star or mini-tree. The goal of every serious display designer is to have individual control over smaller and smaller pieces of the display.

Pixels takes that concept to its ultimate level. We are no longer limited to say, turning on a string of red LEDs. Now we can not only turn each LED in the string on and off independently of the others, we can set its intensity and set its color. A pixel is in fact a multi-color LED (actually 3 tiny LEDs placed very close together, one red, one green, and one blue), along with some circuitry that gives it some smarts. The circuitry that makes up each pixel is able to look at a string of 1s and 0s coming down the data wire, determine which 1s and 0s belong to it, and translate those 1s and 0s into any color and any brightness level.

OK, let's back up. We discussed the sequencing software. When we add pixels, we are adding a whole bunch of channels, because each and every pixel requires 3 channels, one for each of its primary colors. Before pixels, our sequencing software would send out our lighting commands in several possible ways. The Light-O-Rama software is designed primarily to control LOR controllers, over an LOR network, using an LOR 'dongle' that plugs into your show PC. Other software might output channels in the DMX format, or the Renard format, again using a dongle in the PC. While the approaches are different, they are similar in many respects. In every case they use serial data, a fast stream of sequential 1s and 0s, to control many output channels over a single wire. By many, we are talking typically up to hundreds of channels. DMX for example is an industry standard that allows control of up to 512 channels over a single wire. That sounds like a lot, but in terms of pixels it will only let us control up to 170 individual pixels. It's easy to see that a large pixel display using DMX would be a wiring nightmare.

Coming to the rescue, the lighting industry defined ways to get many more channels on a single wire by using Ethernet. Ethernet is the network standard used by all PCs, it's the familiar oversize-phone-plug cables that we call CAT5 or CAT6. (Please note that LOR uses the same type of cable but it is NOT ethernet).

So, how many channels can we send over an Ethernet cable? How about a few hundred thousand? OK, so that solves the wiring problem in two ways. Not only do we no longer need a dongle (or multiple dongles), because our PCs already have Ethernet, we also only need one Ethernet cable regardless of the number of channels we need.

What we do need though is a controller. The sequencing software has the responsibility to convert out channels to a form that can be sent out over the Ethernet. There are 2 standards. The most common is known as SACN, for Streaming ACN, and it's also known as E1.31. The second is called Art-Net. E1.31 is the newer of the two and in general it's easy to implement. While Art-Net has some additional capabilities, it has additional complications as well.

So, our PC and our sequencing software are going to be shooting out our lighting commands as Ethernet data, in the form of packets, using either E1.31 or Art-net. The concept of a packet really isn't all that important, but in simple terms one packet carries up to 512 channels of lighting information. This, by the way, equates to what could be sent over one traditional wired DMX circuit. In order to keep everything sorted out, each group of up to 512 channels is called a universe. Depending on the total channel count your PC will be sending out packets for one or more universes. One universe of E1.31 or Art-Net has the same capacity as that of a single traditional wired DMX network.

This is where the pixel controller comes in. The pixel controller's job is to look at all of the 1s and 0s coming down the Ethernet cable, sort out which ones are lighting data, sort out which universes of lighting data it needs to keep, and eventually to rearrange those 1s and 0s into several different streams to send out to the pixels. And do some crunching along the way so that whatever pixel type is attached to the controller sees its data in the format that it needs to see it in.

Finally we need a power supply to power the pixels, and we need the pixels themselves. So, the complete pixel "system" looks like this:



Pixel System Design

Often it's best to start small, get your feet wet so to speak, then expand your system. As a minimum you will need (in addition to the sequencing software and PC that you probably have already), a pixel controller, a power supply, and at least one string of pixels. "Playing" with a small configuration is a good way to get to really understand how all of the pieces work together.

Eventually though, you'll be ready to design a full display. Once you have a feel for the number of pixels you will be using, and where they will be placed, you can determine the number and type of pixel controllers that you will need. For a fairly small display, 4 strings or less, located fairly close to each other, you could use a single E6804 controller. An E682 can handle up to 16 strings and is often used for larger display objects such as mega-trees. If you will have multiple pixel display elements and they are spread out, you will probably want to have a pixel controller in each location to keep the wire length between controllers and pixels as short as possible.

Keep in mind that the capacity of a single controller can be limited in several ways. There's a limit to how many physical strings you can plug in, there may be a limit as to total power consumed by all of the pixels, and there can be limits based on the total number of pixels vs. available addresses.

You also need to choose a pixel type. There are many variations. Most common are pixel strings, much light LED light strings, or pixel strips, a flat flexible strip about ½" wide that has pixels embedded along its length. With pixel strips the pixel spacing is fixed, although there are typically several spacings to choose from. With pixel strings you can often have them custom built with the spacing that you need.

In general you will need one pixel power supply per pixel controller, in a voltage that matches the voltage needs of the pixels that will be connected to that controller (almost always either 5 volts or 12 volts).

Once you have a general feel for the number and length of pixel strings, and the number, type, and placement of controllers, you can decide how to assign addresses to your pixels.

Addressing begins in the sequencing software, this is where each and every pixel is assigned a group of 3 addresses. This is of course in addition to the channel needs of your non-pixel display items. For pixels, these addresses are expressed as a universe number and a channel number. Each universe can control up to 170 pixels, and to use all of them would require 510 channels. If you have more than 170 pixels you will be using more than one universe.

The specifics of mapping pixels to channels is dependent on the sequencing software used. In general it's easiest to assign them sequentially. Pick a logical starting point in your display and assign the 1st string of pixels to begin at universe 1 channel 1. Usually it's easiest to just assign everything in order, but you may want to leave some gaps at points in the addressing if you know you will be adding additional pixels in the future and you will want them to have certain addresses. In general there's no penalty for skipping over addresses, but you don't want to go crazy and, for example, assign a separate universe to every string of pixels. The reason for that is that the pixel controllers have a limit as to how many different universes they can listen to. If you put every string of 50 pixels on its own universe, and you wanted to put 16 strings of pixels on a controller, you would have a problem because a single controller can't listen to 16 separate universes. On the other hand if you want to assign 3 50-pixel strings per universe, then skip ahead to the next universe for strings 4-6, that would be fine.

Once you have a plan in mind for assigning channels to pixels, then you will have the information you need to configure the controllers. You also need to decide which protocol you're going to use to send the data from the PC to the pixel controllers.

In some cases you won't have a choice. If your sequencing software only supports Art-Net, then you obviously have to choose Art-Net. If you want to use E1.31, but your software only supports Multicast E1,31, then you will have to choose that.

Let's talk a bit about the 3 protocols. Art-Net and Unicast E1.31 both send their lighting packets from the PC directly to a specific controller, using the controller's IP address. The advantage is that traffic on the network is minimized. The disadvantages are that you have an extra step in the configuration process, because the sender needs to know the IP address to send each universe to. Also it's usually not possible to send the same universe to more than one controller. This might be a consideration for example if you wanted all of your pixel addresses to run contiguously without gaps. Say one controller's pixels end in the middle of a universe. Well, to be gap-less the next controller would have to start

up in the same place, and that's not possible with a unicast protocol because each universe can only go to one controller.

Multicast E1,31 on the other hand uses a different approach. Every data packet is sent to the entire network, so the sender doesn't need to know the IP addresses of the pixel controllers, and if necessary several pixel controllers can be setup to respond to the same universe. The disadvantage is that there is more overall network traffic because every lighting packet goes to every controller. This is usually not an issue unless you have a partially wireless network. A large volume of lighting data packets can overwhelm a wi-fi channel. The other potential disadvantage of Multicast is that the E68X controllers are not able to listen to as many Multicast universes as they are Unicast universes. This only becomes an issue if you want to go beyond 7 universes (1190 pixels) per controller. If you do, you will need Unicast E1.31 or Art-Net.

-to be continued-