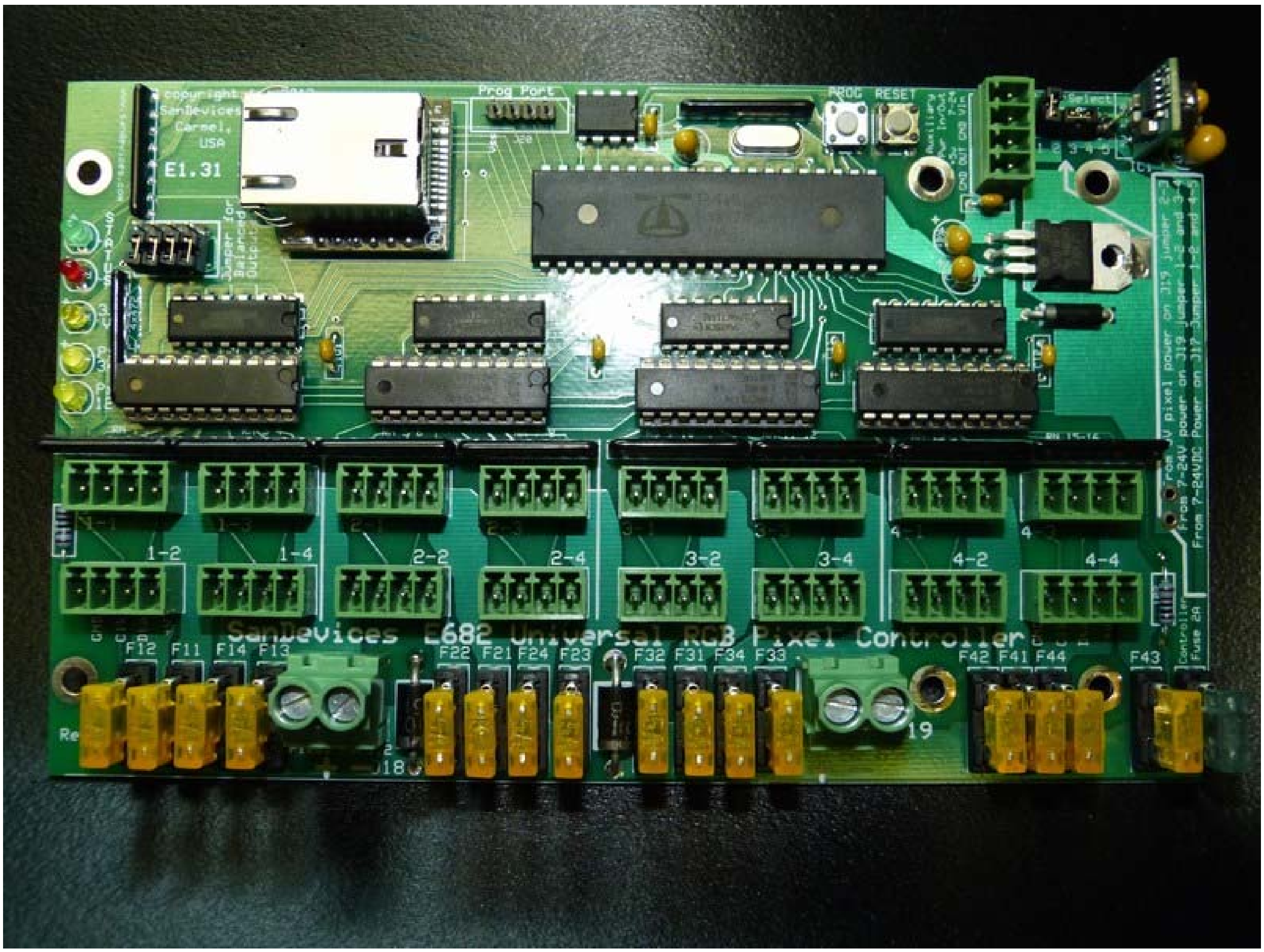
SanDevices E682 Pixel Controller with Version 4 Firmware Operating Manual

Sep 20, 2013

Revision History:
June 03, 2014
Jan 13, 2014
Sep 20, 2013 Updated to reflect version 4 firmware
Feb 20, 2013, minor format changes
Oct 21, 2012 original publication
Jan 12, 2013 updated for firmware version 2.050



Introduction

The SanDevices E682 is an evolution of the E68X product line, and is very similar to its predecessor, the E681. The E682 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 pixel controllers such as the E682 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, or E1.31 “DMX over Ethernet”, or Art-Net lighting protocols. In some cases a dedicated device that is capable of sending the E1.31 protocol may be used in place of a PC.

The E682 is usually installed near the pixels it controls and their power supply, and acts as the “bridge” between the pixels and the PC. The E682 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 170 pixels or 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 E682 via its Ethernet connection. The E682 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 on PC -> Network -> E682 -> Pixels

The E682 is available in kit form, as a fully assembled and tested board, or as part of a complete system including the E682, a power supply, enclosure, and cables (the PS1).
Like all SanDevices products, the E682 is manufactured in the USA.

Feature Summary:

The E682 is a single PC board, approximately 4” x 7”, the mounting hole pattern is compatible with the “new style” industry-standard Keptel or equivalent CG-1500 enclosure.

DMX data input via the SACN protocol, also known as E1.31, or DMX over Ethernet, using a 100mb Ethernet connection. The E682 presently support up to 7 universes of Multicast SACN channels, which is enough to control nearly 1,200 individual RGB pixels, or up to 12 universes of Unicast SACN or Art-Net, which is enough to control more than 2,000 pixels. The use of the SACN or Art-Net protocols allows multiple universes of lighting channels over a single Ethernet cable, and eliminates the need for multiple DMX ‘dongles’ at the controlling PC.

16 On-board individually-fused pluggable screw-terminal connectors for connecting up to 16 separate pixel strings or strips. No soldering is required. Fuses are standard “Mini ATO” plug-in fuses.

All pixel strings are powered from the E682, no external pixel power wiring is needed in most cases. Please note that some pixel configurations may require that some pixel power be supplied externally to the E682. This is typically needed if driving long pixel chains, or a large quantity of 5 volt pixels.

The E682 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 E682 can be configured to operate multiple pixel types (up to 4 simultaneously), and multiple pixel voltages (up to 2 simultaneously).

The E682 is configured via your web browser by accessing the built-in web server using standard web form tools (check boxes, buttons, drop-down lists, etc.

The E682 has a built-in regulated +5VDC output to power a small Ethernet switch.

The E682 is most commonly powered from the same power supply that operates the pixels, but may be powered by a small dedicated power supply (7-24VDC at 500ma) if desired.

Outputs, in groups of 4, can be selected for “balanced” 2-wire operation, to drive standard wired RS-485 DMX devices, or devices using the Renard protocol. In these cases the E682 acts as an E1.31 to DMX or Renard Bridge. In many cases this will eliminate the need for separate hardware (DMX dongles or bridges), to handle traditional wired DMX fixtures or Renard controllers.

The E682 supports 8-bit dimming (256 intensity levels) for the LPD6803 pixel type and 10-bit dimming for many “8-bit” pixels by slightly varying the pixel intensity on each refresh. This feature is available for any string length of TLS3001/CYT3005 pixels, and most 8-bit pixel types if string length is less than 55 pixels.

Specifications:

Data Input:
E1.31, streaming DMX over Ethernet, **multicast mode**, up to 7 universes, or a total of 3570 DMX channels.
E1.31, streaming DMX over Ethernet (SACN), **unicast mode**, up to 12 universes or 6,120 channels.
Art-Net, unicast mode, up to 12 universes or 6,120 channels.

Output:
16 connectors for pixel strings, pixel strips, or DMX/Renard outputs.

Power:
There are two connections for pixel power, allowing the use of two separate pixel power supplies. This also allows a mix of 5 volt and 12 volt pixels (for example) to be controlled by a single E682. Each power inlet supplies power to 8 pixel outputs and is rated to carry up to 32 amps. The E682 itself is typically powered from the right-hand (output groups 3 and 4) pixel power supply, but may also be powered independently by a power supply that is capable of providing 7-24 volts DC at about 500ma, connected to J19.

The E682 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, and “Renard”, and Native DMX mode. The E682 supports 8-bit dimming (256 intensity levels) for the 5-bit LPD6803 pixels and 10-bit dimming for many 8-bit pixels (for pixel string lengths less than 55 pixels).

Note that many common pixel types are “clones” of others. Some common examples are:
WS2811 = 1804
CYT3005 = TLS3001

Full Pixel Type List, Supported Pixel Types (E682)

Common Type	Also Equivalent To
WS2801	WS2803
LPD6803	D705, LPD1101, UCS6909, UCS6912
GE ColorEffects	
TM1804(fast mode)	WS2811, WS2812, TM1803, TM1809, TM1812, SM16715
SM16716	
Native DMX	HX512A, MY994X and standard wired DMX fixtures
Renard (57.6k)	Controllers that use the Renard protocol running at 57.6kbps
LPD8806	LPD8809
P981x	
TLS3001	TLS3002, CYT3005

Please note that the above-listed compatibility is believed to be accurate but not all listed pixel types have been tested to verify the compatibilities listed.

WARNING! IT IS UNSAFE TO USE THE E682, OR ANY OTHER STAND-ALONE PIXEL CONTROLLER WITH ANY PIXELS THAT ARE CONNECTED DIRECTLY TO AC POWER. THIS INCLUDES SOME STYLES OF “GEMMY” PIXELS.

ALL INSTALLATIONS MUST HAVE A SEPARATE ISOLATED POWER SUPPLY TO POWER THE PIXELS.

Mounting the E682:

The E682 consists of a single circuit board measuring 6.8” x 4.375”. There are (6) .150” mounting holes (suitable for #6 screws). The coordinates of the 6 mounting holes (assuming the board is viewed oriented horizontally, with pixel string connectors down, referenced to the top-left corner of the pc board as (0,0) are:

Rev 1.1 PCB:(X,Y): (0.15”,0.725”) (5.15”,0.725”) (6.15”,0.725”)
Rev 1.0 PCB (X,Y): (0.15”,3.725”) (5.15”,3.725”) (6.15”,3.725”)

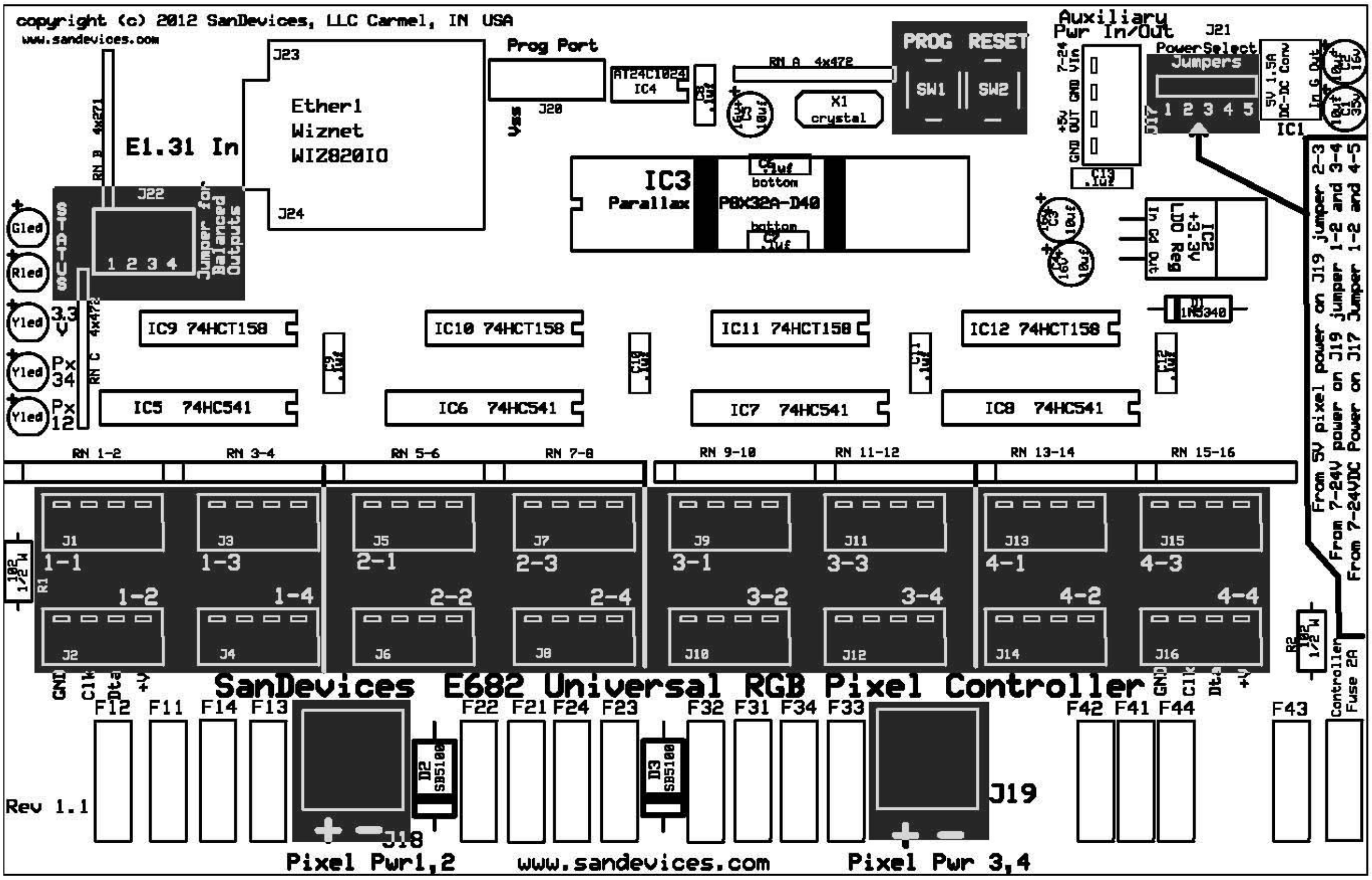
If mounting in a custom enclosure it is suggested that all 6 mounting holes be used. 4 of the 6 holes are used if mounting in a standard “CG-1500” enclosure, using #6 sheet metal screws.

When selecting mounting hardware it is important that, if metal hardware is used, screw heads are sufficiently small (mounting holes are sized for #6 pan-head screws) to insure that they do not come into contact with any circuit board traces outside of the marked outline.   This applies to mounting screws as well as stand-offs on the bottom side of the board. The use of nylon hardware, particularly for standoffs, is suggested.

Since the E682 must be located near the pixel strings it controls, this will often mean that the unit is mounted outdoors. It is the user’s responsibility to provide a suitable enclosure for the E682 and pixel power supplies that will protect these items from direct exposure to moisture.

When selecting a mounting location, keep in mind the need for adequate clearance to allow routing of wires to the pixel power connectors, and for routing the network cable. When referring to parts locations on the board, using the following illustration, the assumption is that you are looking at it “right-side up” as shown in the illustrations, i.e., the large power terminal blocks J18 and J19 will face the bottom.

Figure 2: **E682 Circuit Board Layout**



(The highlighted areas in the above illustration indicate user-configurable jumpers or connection points)

It is recommend that you read this manual in its entirety, plan your pixel layout once you have familiarized yourself with the capabilities of the E682, and once you have that plan in mind, proceed with the setup of the E682.

Setting On-Board Jumpers:

Although most of the E682 configuration is done via the web page, there are a few options that are selected with jumpers. There are 2 sets of option jumpers on the E682. In the upper-right hand corner of the board are the **Power Select Jumpers, designated as J21**. There are 5 terminal pins numbered 1 through 5. There are three possible configurations of these jumpers in this area, depending on the power source for the E682. Typically the E682 will be powered from the pixel power supply attached to J19, since this eliminates the need for a separate power supply. If the E682 is to be powered from the pixel power source (most common case), then set power select jumpers as follows:

- **If the pixel power supply connected to J19 is 5 volts, install a jumper between pins 2 and 3 ONLY**
- **For pixel power voltages greater than 5V on J19, install a jumper between pins 1 and 2, and a jumper between pins 3 and 4.**
- If you elect to power the E682 from a separate power supply **install a jumper between pins 1 and 2, and a jumper between pins 4 and 5.**

In this case connect the external supply to J17, the upper terminal is positive, and the 2nd terminal from the top is negative. The external power supply should be from 7 to 24VDC, and capable of supplying at least 500ma, or 1500ma if powering an Ethernet switch from J17. The external supply should be current-limited or fused at no more than 2 amps.

The standard configuration that E682s are shipped from the factory is for 12V power (jumpers 1-2 and 3-4).

The E682 contains circuitry to protect the controller's electronics if a voltage of greater than 5 volts is connected when the board is jumpered for 5 volt power. This will result in the 2A fuse blowing. If the 2A fuse is found to be open, double-check the power select jumpers. The proper configuration of these jumpers is shown on the E682 silk-screen at the extreme right edge of the board.

Caution: If using 5V pixel power, and the power supply voltage is slightly high, the controller will appear to work even if jumpered for 12V. This operation will not be reliable however. It is important to ALWAYS make sure that the on-board power jumpers are correct for the pixel power supply voltage being used.

The other set of configuration jumpers is highlighted in the board layout at the upper left and is labeled J22. These jumpers, one per output cluster, are installed to enable the "2-wire balanced" output feature, explained later in this manual.

LED Indicators:

There are 5 LEDs on the E682, located in a line along the left edge. The **red STATUS LED will be on during normal operation. The green STATUS LED generally indicates the presence of received SACN/E1.31 data packets, or that a test pattern is enabled, and will usually light during normal operation, or flicker rapidly in when a test pattern is enabled.** The 3 **yellow POWER LEDs** indicate the presence or absence of different power voltages on the E682 and are used for troubleshooting. The 2 bottom yellow LEDs indicate the presence of power on the two pixel power connectors. If either of these LEDs is not lit, it indicates that no pixel power is being supplied to the corresponding pixel power input. The upper yellow LED indicate the presence 3.3 volt power, this is the power source that operates the E682's electronics. If this LED is not lit the controller is not receiving power, either the power supply voltage is not present, (this is most likely the case if the "PX 3,4 Pwr" LED is also out), or a blown controller fuse (if the "PX 3,4 Pwr" LED is lit).

Pluggable Resistor Networks:

There are 8 pluggable resistor networks on the E682 located just above the upper row of pixel string connectors. Each resistor network affects the 2 string connectors immediately beneath it. These resistor networks may be changed to a different value in certain situations, depending on pixel type, and/or type and length of pixel wiring. These networks may also need to be changed for outputs that are driving DMX devices. The resistor networks must be in place for the E682 to operate. The standard "as-shipped" value for these resistors is 270 ohms (part number containing "271". Smaller resistor values may be needed in cases where the distance between the E682 and the start of the pixel strings is unusually long.

Fuses:

The E682 uses standard automotive-style "Mini-ATO" fuses. Replacements are readily available at any auto parts store. The E682 will generally be shipped with 5 amp fuses installed in the 16 pixel string fuse locations. Some installations may requiring that these fuses be replaced by fuses of a larger value. These fuses protect the pixel string wiring in the event of a short circuit. The 17th fuse is 2 amps, and protects the E682's electronics.

Note that a short circuit at the very end of a pixel string may not blow the fuse, since in many cases the resistance of the pixel wiring itself limits the current flow from such a short to a value below the rating of the fuse.

Each of the 16 left-hand fuse positions is designated with the number of the string it protects. Fuse "F21", for example, protects the 1$_{st}$ string in cluster #2. **These are NOT in order, check the legend by the fuseholder to make sure you are checking the proper fuse. The fuse order, from left to right, for each output group is: 2, 1, 4, 3.** The right-most fuse position is for the power to the controller circuitry. This fuse should be 2 amps. The most likely cause of blowing this fuse is incorrectly setting the power select jumpers.

<u>Do NOT replace a blown 2A fuse with a 5A fuse. This will defeat the protection circuitry on the E682 and could lead to component damage.</u>

Making Connections to the E682:

There are 3 basic types of connections required for operation of the E682: One or two pixel power supplies, an ethernet connection to the local LAN, and the pixels themselves. In certain situation there will also be a connection to the AUX POWER connector, J19, described later.

Pixel Power Supply Connections:

The E682 must be used in conjunction with an appropriate pixel power supply. The specifications of the pixel power supply will depend on the type and quantity of pixels being driven. 5 volt and 12 volt pixels are the most common. If using 2 different voltages of pixels simultaneously, you will need either 2 separate power supplies, or a single supply that can supply both voltages.

Power for the pixels is connected to the two large 2-terminal screw blocks on the bottom edge of the board, designated J18 and J19. The left-hand terminal is marked positive (+) and the right-hand terminal is negative (-). There is a legend silk-screened on the board below the terminal blocks.

Power to the left-hand connector, J18, supplies power to the pixel strings connected to connectors 1-1 through 2-4. Power to J19 powers the pixels on connectors 3-1 through 4-4, and usually supplies power to the E682 itself as well.
One power supply can power both sides, either by running 2 sets of + and – wires from the power supply, one to each terminal block (preferred), or by connecting power to one terminal block and installing a jumper wire between the 2 positive terminals. Alternatively the E682 can be powered by 2 separate power supplies, one driving each 'side' of the board. To know the current requirements of the pixel power supplies, you need to know the power requirements of each pixel and the total number of pixels. A good rule of thumb is about 3 amps per 50 pixels (5V pixels) or 3 amps per 100 pixels (12V pixels). These numbers are guidelines only, a specific pixel types may differ from these values.

To power a load of 16 5V 50-pixel strings, you should have a power supply rated at a minimum of 50 amps, and use short runs of 12 gauge wire or larger between the power supply and the board. As a safety measure, <u>the negative side of the power supply should be connected to earth ground by wiring a jumper between the power supply's GROUND terminal and the V- terminal.</u>

Note: Typically, RGB pixels will operate off of 5 volts or 12 volts, occasionally 24 volts. **It is important that you match your power supply to the requirements of your pixels**. The E682 is capable of driving pixels of 2 different voltages at the same time. **<u>It is very important, when using a dual voltage setup, that you do not plug pixels into a power supply that supplies more voltage than they are rated for</u>.** In other words, don't plug a 5 volt pixel string into a pixel connector that is wired for 12 volt pixels, as the pixel string will most likely be destroyed. If in doubt please contact SanDevices for assistance in selecting a suitable power supply.

Use care when connecting the pixel power supply to insure that the polarity is correct, + to +, and – to -. Although the E682 contains circuitry to attempt to protect against a reversed connection, the effective performance of that protection depends on the power supply being able to shut down in the event of a reverse-polarity type short-circuit condition. Some power supplies, particularly if there is significant voltage drop in the power wiring, may not draw enough current to sense the condition as a short and in that case it is still possible to damage the controller and/or pixels with improper wiring.

- **Double-check for correct pixel power wiring before turning on the power.**
- **Be certain that wiring from controller to pixels is not shorted. A short-circuit in the pixel wiring, particularly in the case of 12V pixels, could damage the controller or the pixels.**

<u>The fuses on the E682 will not protect the wiring from the power supply to the E682.</u>

Because pixels are low voltage devices, the pixel power supply must be located near the E682, preferably within the same enclosure. This usually means that the power supply will be mounted outdoors. It is the responsibility of the user to insure that both the power supply and the E682 are properly protected from moisture. Also, when using this equipment outdoors, **<u>ALWAYS power the pixel power supply from a ground fault-protected receptacle.</u>**

<u>It is strongly suggested that the 120VAC power input terminals of the power supply be protected from accidental contact to protect against the possibility of electric shock.</u>

Network Connection:

The Ethernet jack on the Ethernet module must be connected to your LAN, or directly to the LAN card of the PC that will be controlling the pixels. The E682 LAN port is capable of running at 100mb/sec, and is auto-sensing. A crossover cable is not required for a direct connection to a PC. Further discussion of network connection will be found later in this manual.

Pixel Connections:

The pixel strings plug into the 16 4-pin “euro-style” pluggable terminal blocks labeled 1-1 through 4-4. In general, pixel strings may be plugged and unplugged while the system is powered up, but it is recommended, to reduce risk of damage to pixels, that pixel strings only be plugged and unplugged while the system is off. **If using the GE ColorEffects strings, they will not function properly if plugged in while the E682 is powered up.**

Connecting Pixel Strings to the E682:

A complete set of mating 4-pin screw-terminal connectors are supplied with every E682. It is the user’s responsibility to make the proper connections from the pixel strings to the connectors on the E682. The mating connectors are illustrated below:

Depending on the type of pixels being used, there will be either 3 or 4 wires between the pixel strings and the E682. All pixels have +V (power), ground, and DATA connections. Some pixels have a CLOCK connection as well. For runs of up to 10 feet or so, 20 gauge wire is suitable, beyond 10 feet, 18 gauge wire or larger should be used. It is recommended that the distance from controller to pixel strings be kept as short as possible. There are 2 factors that limit the maximum distance, voltage drop, and degradation of the signal on the data and clock wires. The maximum allowable distance between the E682 and the pixel strings will vary depending on pixel type and the type of wire used. Up to 15 feet or so should be fine, often significantly longer distances are achievable, but beyond 20 feet or so the configuration should be tested before the final installation is done. Symptoms of the cabling being too long are dimming or color-shifting of the last pixels in the string (usually an indication of excessive voltage drop, often correctable by using heavier wire), or flickering, generally an indication of corruption of the clock or data signals, sometimes correctable by using a different type of wire, by using “null” pixels, or by using smaller value resistors in the pluggable resistor networks.

CAUTION:

You MUST know the correct color-code used by your pixel strings. This can and will vary from manufacturer to manufacturer, even from lot to lot. Wire color is NOT a reliable indication of function.
If in doubt please contact the pixel vendor for clarification.

Incorrectly wiring pixel strings can and will destroy them. Not only must you identify the function of each wire, but you must properly identify the START (input) end of the pixel string.

Incorrect identification of pixel connections is the #1 issue when users report non-working systems. It cannot be emphasized strongly enough that you MUST know which wire is which signal and you MUST know which end of the pixel strings is IN and which end is OUT.

If you are using the black 4-wire waterproof cable assemblies from SanDevices or from Ray Wu, or pixel strings that have those cables pre-attached, the color code for those cables **SHOULD BE** as follows:

RED +V	pixel power
GREEN	Data
BLUE	Clock (not used on 3-wire pixels)
BLACK	Ground

If you look at the E682 circuit board, beneath the 1-2 pixel string connector, the function of each pin position on the pixel connectors is identified. From left to right, the pins are:
Gnd Ground.
Clk This is the clock line, used on some but not all pixel types.
Dta This is the data line. It is required for all pixel types.
+V This is the (typically) +5V or +12V power from the terminal block to the string.

If your design layout calls for longer controller-to-pixel distances, the “Null Pixel” feature may be a work-around. This involves the installation of extra pixels in the cable run between the E682 and the pixel string. Since each pixel regenerates the data signals, this is an effective technique if the issue is one of data corruption. Other possible solutions are heavier wire from controller to pixels, or smaller values of the pluggable resistor packs. If the issue is voltage drop, possible solutions are heavier wire between pixels and the E682, or locating the pixel power source closer to the pixels.

Obviously, you need to be 100% sure that your wiring is correct before plugging in a Pixel string or strip. It is suggested, if you have the capability to do so, to check voltages at the ‘string’ end of the connecting cables that you make up to verify that the proper voltage is on the proper wire, before splicing to the pixel string itself.

- **Incorrect wiring of a pixel string may cause irreversible damage to the pixels.**

Using the Auxiliary Power Connector, J17:

J17 is a multi-purpose connector. Typically it is not needed, but it can be used for 2 optional features. If you choose to power the E682 from its own power supply, rather than from the pixel power source, you would connect that power supply to J17. +V to the top pin, and GROUND to the 2nd pin from the top. This power supply should provide from 7-24 volts DC, and be rated at a minimum of 500ma. This supply should be externally current-limited or fused at no more than 2 amps. **If the E682 is powered in this manner, Power Select Jumpers must be installed between pins 1 and 2, and 4 and 5.** The second use of J17 is to supply a +5VDC output. This would typically be used to power a small Ethernet switch located near the E682. Having a small Ethernet switch near the controller can simplify the network wiring since, when using multiple controllers, it eliminates the need to run an Ethernet cable from every controller back to a remote switch. The ability to power the Ethernet switch from the E682 simplifies wiring, since it eliminates the need for a separate switch power transformer, and the 120VAC wiring to it. You must of course choose an Ethernet switch that runs off of +5VDC, at a maximum of 1 amp. The +5V power output is the 3rd terminal from the top of J17, and ground is the bottom terminal. See the screened legend on the board.

Note that J17 can be used for both functions simultaneously, but in that case be sure to use an external power supply rated at a minimum of 1.5 amps since it will be powering both the E682 and the connected Ethernet switch.

J17 Pinout is as follows (top to bottom):

+V In external power input, +7 to +24 volts DC, if powering the E682 from its own power supply (not generally required)
GROUND

+5V Out This is a power out, to provide +5V, typically used to power a small Ethernet switch that is mounted in the enclosure with the E682.

GROUND

Using the Remote Port (J25), available on Rev 1.3 boards only:

J25 is available on Rev 1.3 boards to allow connection of a remote PROG pushbutton and/or a remote green status LED. This allows accessing the “override” feature and status checking by mounting a pushbutton and/or LED on the outside of the E682’s enclosure. Printed legends are on the PC board that define the function of each pin.

Initial Startup and Testing

If you have pixels of one of these types: 2801, 1804/2811/2812, GE, or TLS3001, you can do a very simple test to insure that the controller is functioning and that your pixels are properly wired:

<u>Make sure that the power select jumpers are set properly for your pixel voltage.</u>

Connect a power supply with the proper voltage for your pixels, and enough capacity to power at least 1 test string (typically 5 amps) to the E682. Connect the power supply (-) to the (-) on J18 or J19. Connect the (+) power supply lead to the (+) terminals on J18 **<u>AND</u>** J19 (use 2 + wires or 1 + wire and a jumper between the 2 + terminals).

Turn power on to the E682. Verify that all 3 yellow LEDs are lit. After a couple of seconds, the red and green LEDs should light also, with the green LED flickering. If the 3 yellow LEDs do not light check your power wiring and the setting of the power jumpers. Turn power off.

Plug a (PROPERLY WIRED!) pixel string into one of the pixel output connectors as follows based on the pixel type:

Type 2801 use any connector from 1-1 thru 1-4
Type GE pixels, use any connector from 2-1 thru 2-4
Type 1804/2811/2812, use any connector from 3-1 thru 3-4
Type TLS3001/CYT3005, use any connector from 4-1 thru 4-4

Re-apply power to the E682. The on-board LEDs should light as described above. If the board is working properly (and if the test pattern was enabled at the factory) you should see a single bright pixel chase through your string, leaving dim pixels behind. This pattern will repeat every couple of minutes. Please note that it takes time for the test pattern to complete its initial cycle. Depending on which connector your pixels are plugged into, it may take a minute or more for the test pattern to appear.

If this test is successful it tells you that the E682 is working, your pixel wiring is correct, and your power supply and power wiring are correct.

The next test will allow you to access the E682's web page. This is essential in order to be able to configure the many options of the E682.

The E682 is a network device. It needs to be attached in one of 2 ways. If you have a 'wired' Ethernet network, the E682 is usually attached to an unused Ethernet port on a network router or switch. If your PC uses wireless networking, (nothing plugged into the PC's Ethernet jack) you will probably use a direct Ethernet connection from the E682 to the Ethernet jack on your PC. In either case any standard CAT5 or CAT6 Ethernet cable will work, it does not have to be a 'crossover' cable.

Every Ethernet device has an IP address, which is used to enable communication between the various devices on the network. IP addresses are usually assigned automatically, almost always by your router. IP addresses may also be set manually, this is known as a STATIC IP address. **<u>The E682 ALWAYS uses a static IP address.</u>**

Each network encompasses a range of IP addresses. IP addresses are ALWAYS shown in the format a.b.c.d where a, b, c, and d, are numbers ranging from 0 to 255.

In order for your PC to be able to access the web page of the E682, the E682 must have an IP address that lies within the range of addresses used by your network.

Most small network have IP addresses of the form 192.168.X.Y. X will be the same for all devices on the network, but Y will be different. The most common value for X is '1'. In some cases it can be another value, with '0' and '2' being the next most common.

Since '1' is the most common X value in small networks, the E682 ships with a default IP address of 192.168.1.206. For many small networks this address will work "out of the box" and need not be changed.

One exception is if your PC uses a wireless network connection. In this case the wired Ethernet jack on the PC will typically have an address such as 169.254.X.Y.

You will need to follow the procedure below that applies to your network.

#1) You use a WIRED network, and the Ethernet jack on your PC connects to your router or to an Ethernet switch.

In this case the E682 should connect to an unused port on your router and will need to have an IP address that matches the address range of your network. Since the 192.168.1.X network is so common, it's often easiest just to try it and see if it works:

Plug an Ethernet cable from the E682 to an unused router LAN port.

Apply power to the E682 (no pixels needed for this). After a few seconds all 5 LEDs should be lit, green may be flickering.

Open your web browser and type **192.168.1.206 into the address bar** (where you see the http://www-type entries) then press **ENTER**.

If you see a web page pop up that says SanDevices E682 at the top, you are in business and can move ahead to the "configuration" section. If not, the most likely cause is that your network uses something other than 192.168.1.X addresses.

The first step is to determine which IP address range your network uses, then we will use the Over-Ride function to force the E682 to use an address in that range.

To determine the IP address range of your network, do the following:

Press the **WINDOWS** key to bring up the START menu.

In the command box (may say search programs and files) type **CMD** then press **ENTER**.

This should cause a black window to open. Type this into the black window:

Ipconfig /all   (that's i-p-c-o-n-f-i-g (one word) SPACE then forward slash(/) then the word all) then press **ENTER.**

This will display a bunch of text in the black window. Use the right-side scroll bar to scroll up to the top. Look for a section called "Ethernet adapter Local Area Connection".   A few lines below that look for a line labelled IP Address or Ipv4 Address. To the right you will see a number of the form (usually): 192.168. (then 2 more numbers). The first of those "2 more numbers" is the one we are interested in. For example if you see 192.168.0.101, the number we're looking for is the 0.

Now we are going to force the E682's IP address to be in the range of your network. Take the number from above (usually 0 or 2 thru 10, if it was a 1 the first method we tried should have worked) and add 5 to it. **This is the override number that we will use. Do an Override (described a few pages ahead as <u>Using the Override Function</u>) using that override number (should be 5, or 7 to 15).**

After you do this the E682 will (temporarily) have an IP address that is within the range of your network and has a last number of 206. (note: replace the X in this example with your network number that you got from the black screen above, ie 0, 2, etc). Open the web browser and type in an address of 192.168.X.206 where X is the 3rd portion of your PC's IP address as determined above. For example, if the "ipconfig" procedure showed that your PC had an address of 192.168.0.101, then the 3rd digit is 0, you would have used override 5, and you would now type in 192.168.0.206 in your browser's address bar then press **ENTER.**

Now do you see the E682's web page? If so, success, please jump ahead to Configuring, if not please re-read and repeat the instructions in this section.

#2) You use a wireless network for your internet connection, and the Ethernet jack on your PC is not used.

In this case you will connect the E682 directly to your PC's Ethernet jack using an Ethernet cable. Typically, an unused PC Ethernet port will be given an IP address of 169.254.X.Y when Windows starts. Therefore we will use an Override to force the E682's IP address to be within this range. Perform the OVERRIDE function (described below) using an override code of 4.

Now, go back to the browser address bar and type this in: **169.254.74.73** and press **ENTER**.

Do you see the E682's web page? If so, success, please jump ahead to Configuring, if not please re-read and repeat the instructions in this section.

Using the OVERRIDE function

Read this procedure through completely before doing it. You will need to know the **OVERRIDE NUMBER** that you want to perform, generally 1 thru 15:

1) Press and hold the PROG button on the E682.
2) After a few seconds the red and green LEDs will begin flashing on and off together at the rate of about 1 flash per second. Wait until the LEDs flash the number of times equal to your override number, then release the PROG button.

Example: If your override number is 5, press and hold PROG, and count the flashes. When the LEDs come on for the 5th time, release PROG.

This is a list of all override numbers and their purposes:

1 Display the E682's currently stored IP address on the LEDs, repeat until powered off.
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…..

Overrides 7 through 15 operate similarly to 5 and 6 in that they force an IP address of 192.168.X.206 where X is the override number-5.

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, and save it by clicking **Update System Information**, then restart the controller.

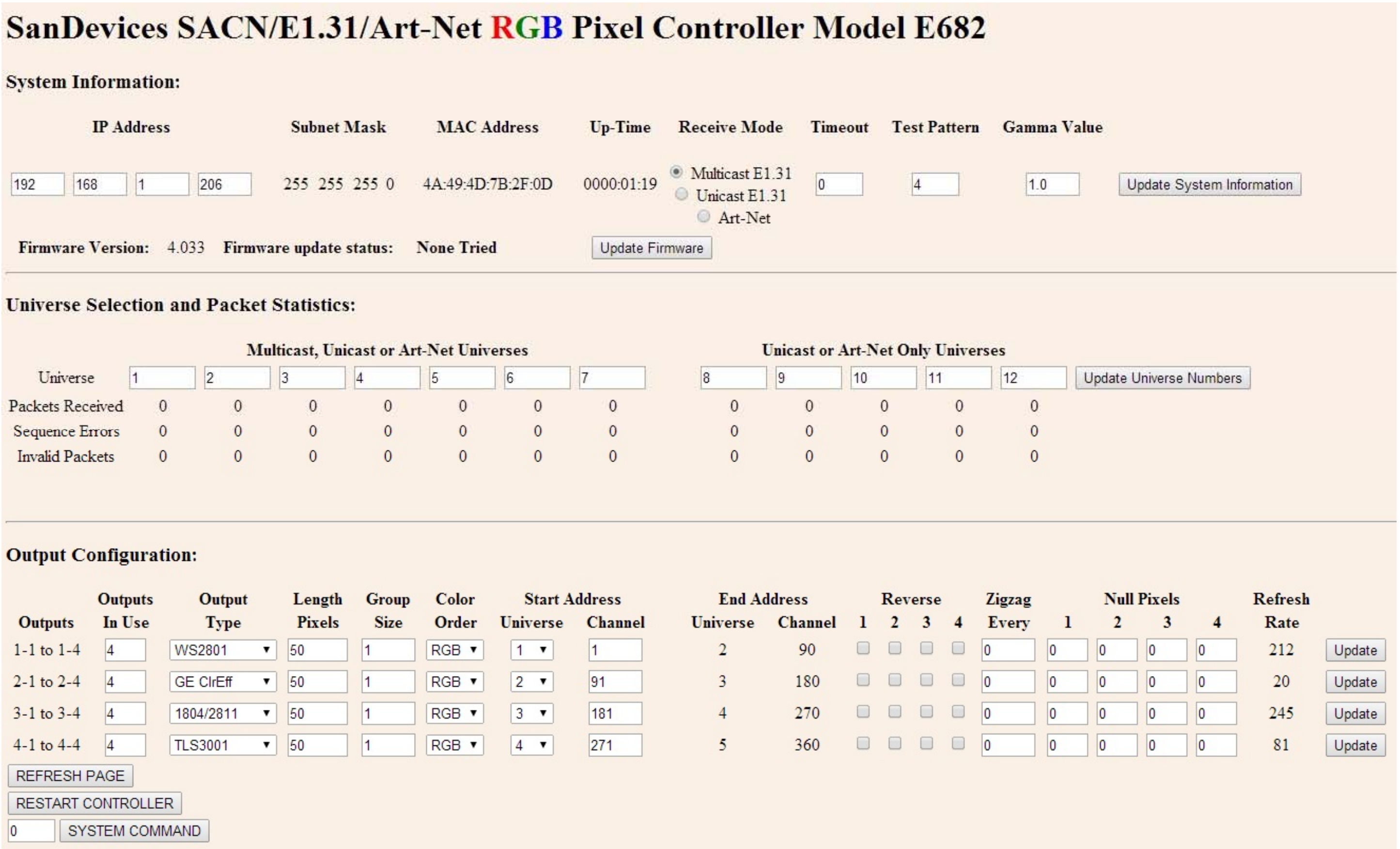
Use over-ride code 1 to determine the IP address of the controller when it is not known. Press and hold the PROG button until the red and green LEDs come on together for the first time, then release (over-ride 1). The LEDs will now display the current IP address. For each digit, the green LED will come 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, by pressing the RESET buttons (for boards that have a RESET button), 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.

Configuring the E682 to operate with your pixels:

Once you have successfully connected to the E682's web page (possibly using an override) you should see a web page similar to this:



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 group has its own configuration line and its own **UPDATE** button. So when configuring outputs, you would make all changes to output group 1, then click that line's **UPDATE** button. Then move on to output group 2 and repeat the procedure.

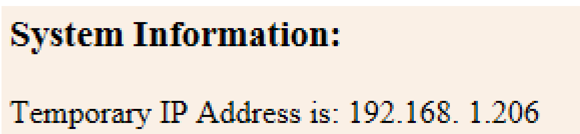
The web page is divided into 3 main sections: **System Information**, **Universe Selection**, and **Output Configuration**.

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 you 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 over-ride mechanism, described earlier in this manual that allows you to use the **PROG** 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's web page, 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. If an override was used, 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:



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, if that is the case, just use the next consecutive address. If you have 3 E682s for example, the first one can be 192.168.1.206, the 2nd can be 192.168.1.207, etc. Enter the 4 parts of the desired IP address into 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.

Once the IP has been set, or if you don't need to change the IP address, continue here:

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. ArtPollReplys, and broadcast Art-Net packets, 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.

IMPORTANT NOTE RE FIRMWARE VERSION 4.026: There are known bugs in the Art-Net implementation with firmware versions prior to 4.033. If you are running version 4.026 please do NOT select the Art-Net Mode. If you need Art-Net and have a firmware version earlier than 4.033 please contact SanDevices for assistance.

Timeout may be set to any value in the range of 0 or 2-99. 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. There is a known issue with 4.033 firmware that will cause pixel flicker if timeout is set to 1.

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 (in some cases several seconds 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 presently 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. Gamma correction is ALWAYS applied to pixel type TLS3001, and will also be applied to 8-bit pixels if the string length is 55 or less. To disable gamma correction set the value to 1.0.

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 should be set to a value other than 0, that does not duplicate any other universe number. Valid universe numbers are 1-63998. There should be no duplications in the universe numbers used, in other words **don't enter any one universe number into 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.**

IMPORTANT NOTE RE FIRMWARE VERSION 4.026: Version 4.026 contains a bug that prevents universe numbers greater than 511 from displaying properly. This problem affects the display only, the number is in fact properly saved and will work as entered, it just won't display properly if it's higher than 511. This has been corrected with version 4.033.

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.

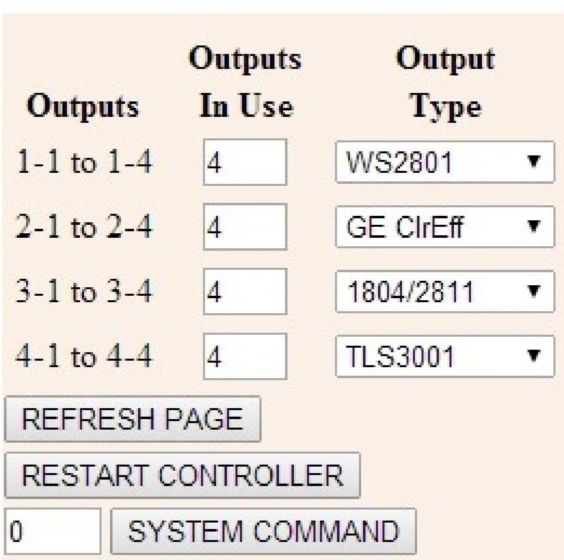
Important: The first 7 universes can be used for pixel control data of any type: Multicast E1.31, Unicast E1.31, or Art-Net. The last 5 universes cannot be used for Multicast E1.31. If you use Multicast E1.31 you must only assign the 1st seven universes to your pixel strings (see Output Configuration,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

Output Configuration

Many of the pixel configuration options of the E682 are configured in groups of 4 outputs, or "output groups".

This is where the individual controller output groups are configured. Output groups are labeled 1 through 4, matching the numbers on the E682 itself. In other words, Output Group 1 consists of the 4 outputs labelled 1-1 thru 1-4. Each output group 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 group then click that Update button to save those changes.** Then move on to the next output group and repeat the procedure.



Outputs In Use This is the number of outputs (0 thru 4) that is in use in this output group. 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 value should reflect the actual number of pixel outputs in this group that will be used, and outputs must be used beginning with the first output in the group. In other words, if outputs in use is set to 2 for output group 3, it means that outputs 3-1 and 3-2 will be used, but 3-3 and 3-4 will not.

Output Type is a drop-down selection list that allows you to choose the type of pixel (or other device) attached to this output group (all outputs within an output group will have the same pixel type). 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 group. 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.

If you connect strings of different lengths to an output group, set the **Length in Pixels** based on the size of the longest string. In general, it is most efficient to keep the strings in an output group as similarly-sized as possible.

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.

Length	Group	Color	Start Address	
Pixels	Size	Order	Universe	Channel
50	1	RGB ▼	1 ▼	1
50	1	RGB ▼	2 ▼	91
50	1	RGB ▼	3 ▼	181
50	1	RGB ▼	4 ▼	271

Start Address Each output group 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 X (Number of Outputs In Use), 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 group. All pixels on that output group 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 (not typical, but possible to do). If we define output group 1 as having 200 pixels with 2 outputs in use, then this output needs 1200 total channels (200X2X3). Assume that we start this output at Universe 1 Channel 1. Since there are only 510 useable channels per universe, the next 510 channels will be assigned from Universe 3, since Universe 3 follows Universe 1 in the universe selection list, and the last 180 channels will be assigned to universe 5, since it follows universe 3 in the universe 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.

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.

Color	Start Address		End Address	
Order	Universe	Channel	Universe	Channel
RGB ▾	1 ▾	1	2	90
RGB ▾	2 ▾	91	3	180
RGB ▾	3 ▾	181	4	270
RGB ▾	4 ▾	271	5	360

Note: Start Address and End Address values are NOT displayed for any output group that has no outputs in use , or has a length of 0.

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.

Reverse				Zigzag	Null Pixels				Refresh	
1	2	3	4	Every	1	2	3	4	Rate	
☐	☐	☐	☐	0	0	0	0	0	212	Update
☐	☐	☐	☐	0	0	0	0	0	20	Update
☐	☐	☐	☐	0	0	0	0	0	245	Update
☐	☐	☐	☐	0	0	0	0	0	81	Update

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 as 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 at 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**.

Note: The E682s web page is ALWAYS static, in other words it NEVER updates by itself. It will stay as-is until you either hit refresh type in a command or hit "refresh" which is the same as re-typing the previous command. The times and statistics shown on the page are as of the last time the page was displayed.

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:

Sequencing Software → Ethernet Port on PC → Ethernet Cable to Controller → Pixel Controller --------→ Pixels

Pixel Power Supply

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.

2-Wire Balanced Outputs:

One of the new features of the E682 is the ability to provide 2-wire balanced outputs. This feature would typically be used for outputs that are selected as type DMX when those outputs will be driving standard wired DMX fixtures.

The E682 is also able to provide Renard-compatible outputs as well using the Balanced Output feature Balanced output mode may also be used with 3-wire pixels in conjunction with a balanced to unbalanced converter at the pixel string to enable running longer cable distances between the E682 and the pixels.

This feature is enabled on a per-output group basis by installing the corresponding jumper in the J22 area. When a jumper is installed, the function of the output pins for those 4 outputs is changed as follows:

Normal Use	Use with J22 Jumper Installed
DATA	DATA+
CLOCK	DATA-
GROUND	GROUND
+V	+V (often not used)

The pinout to connect to a standard wired DMX 'XLR' connector would be:

GROUND	XLR Pin 1	;signal ground
DATA+	XLR Pin 3	;data+
DATA-	XLR Pin 2	;data-
+V	not used	

When a cluster is configured for Native DMX, each output of that cluster becomes a separate DMX universe output. The channel numbers corresponding to each "string" are mapped to the first n channels of that output, where n is the defined string length X 3.

For example, if Cluster 4 is defined as chip type “Native DMX”, and we set cluster 4 to have 2 String(s) of length 170 pixels, then outputs 4-1 and 4-2 will each output a 510-slot DMX universe.

For best results, the cluster(s) being used for DMX should have the Balanced Output jumpers in place so that the output becomes a standard 2-wire ‘balanced’ DMX signal.

This feature can also be used in conjunction with a balanced receiver (located at the pixel string), to drive 3-wire pixels over much longer distances than are achievable with the standard unbalanced pixel data signal. This is limited to 3-wire pixels because the CLOCK signal is not available in this mode.

Please note that this type of output will not have the full drive capability of purpose-built DMX hardware. This feature is intended primarily to drive limited numbers of DMX fixtures over relatively short distances (say tens to hundreds of feet rather than thousands). It may also be necessary to substitute pluggable resistor packs of a different value for optimum performance in balanced mode. The proposed configuration should be tested to insure proper operation prior to installation.