SanDevices E681 Pixel Controller Operating Manual

Sep 24, 2011

Revision History:

Sep 24, 2011 original publication

Introduction

The SanDevices E681 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 E681s and/or E680s are used together with one or more pixel power supplies, one or more strings 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 lighting protocol.

The E681 installs near the pixels it controls and their power supply, and acts as the bridge between the pixels and the PC. The E681 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.

Feature Summary

Single PC board, approximately 4” x 7”, mount points compatible with CG-1500 enclosure

DMX data input via E1.31, 100mb, DMX over Ethernet, up to 4 universes, eliminates the need for multiple DMX ‘dongles’ at the controlling PC.

On-board individually-fused pluggable screw-terminal connectors for up to 16 pixel strings

Pixel strings powered from the board, no external pixel power wiring needed

Extremely versatile, with many programmable options, and many pixel types supported

Including 5-bit, 8-bit, and 12-bit pixels.

Multiple pixel types/voltages can be driven simultaneously

Simple programming via a web page

+5VDC output to power a small Ethernet switch

Signal levels selectable either 3.3 volts or 5 volts

May be powered from any pixel power supply from 5 to 24 volts

Specifications:

Input: E1.31, streaming DMX over Ethernet, **multicast format**, up to 4 universes, or a total of 2048 DMX channels. **The E681 does not support unicast E1.31.**

Output: 16 connectors for pixel strings. The board can control up to 680 individual pixels using up to 2040 DMX channels. (The limitation is the number of available DMX channels, and if some or all of the pixels are to be controlled in groups, then the total pixel count can be much larger.)

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 board. Each power inlet supplies power to 8 pixel strings. The E681 itself is typically powered from the right-hand (clusters 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 E681 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.

Mounting the E681

The pixel controller consists of a single circuit board measuring 6.8″ x 4.1″. There are (6) .125″ mounting holes (suitable for #4 screws). The coordinates of the 6 mounting holes (assuming the board is viewed horizontally, with pixel string connectors down, referenced to the top-left corner of the pc board are:

(X,Y):	(0.5″,0.6″)	(5.0″,0.6″)	(6.3″,0.6″)
	(0.5″,3.6″)	(5.0″,3.6″)	(6.3″,3.6″)

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

When selecting mounting hardware it is important that, if metal hardware is used, screw heads are sufficiently small to insure that they do not come into contact with any circuit board traces outside of the marked outline. The use of nylon mounting hardware is suggested whenever possible.

Since the E681 must be located near the pixel strings it controls, this will often mean mounting the unit outdoors. It is the user’s responsibility to provide a suitable enclosure for the E681 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, i.e., the large power terminal blocks J17 and J18 will face the bottom:

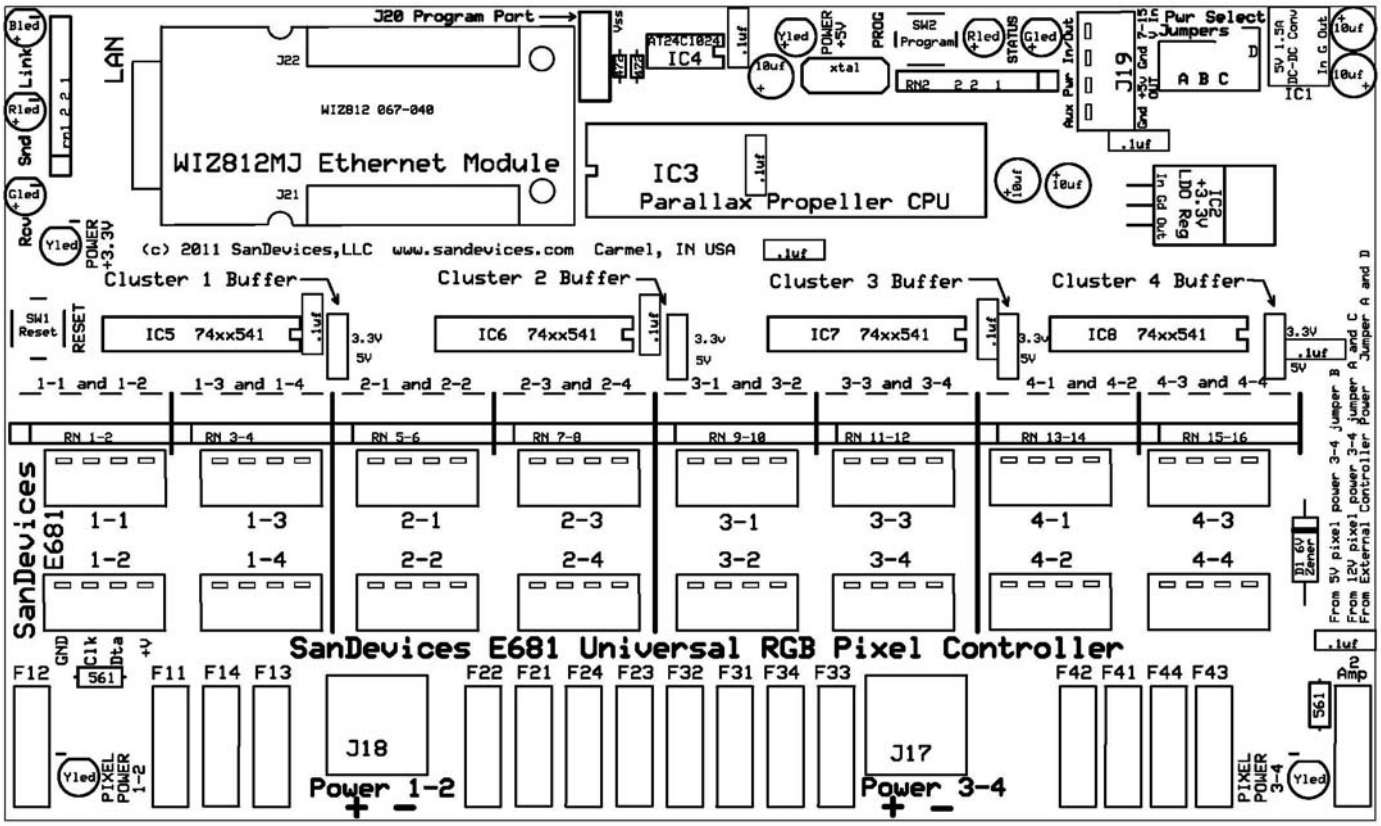


Figure 1: **E681 Circuit Board Layout**

Setting On-Board Jumpers

There are 5 sets of option jumpers on the E681. In the upper-right hand corner of the board are the **Power Select Jumpers**. There are 4 possible jumper positions, labelled A though D, and arranged in an L shape. There are three possible configurations of these jumpers depending on the power source for the E681. Typically the E681 will be powered from the pixel power supply attached to J17, since this eliminates the need for a separate power supply for the E681. If the E681 is to be powered from the pixel power source, then set power select jumpers as follows:

For 5 VOLT pixel power on J17, install jumper B ONLY

For pixel power voltages greater than 5V on J17, install jumpers A and C ONLY

Finally, if you elect to power the E681 from a separate power supply **install jumpers A and D only**. In this case connect the external supply to J19, the upper terminal is positive, and the 2nd terminal from the top is negative. The external power supply should be from 7 to 15VDC, and capable of supplying at least 500ma, or 1500ma if powering an ethernet switch from J19. The external supply should be current-limited or fused at no more than 2 amps.

The remaining 4 jumpers select the signal voltage level to the pixel strings. These are the 4 jumper locations with positions labeled 3.3V and 5V. In almost all cases these should be set to the 5V position.

LED Indicators

There are 9 LEDs on the E681. The Green, Red, and Blue LEDs in the upper-left corner indicate the status of the network connection; Receive, Send, and Link, respectively. The blue Link LED will be lit whenever the E681 is properly wired to an Ethernet switch or a PC. The green and red LEDs indicate receiving or sending data on the network.

The **red and green Status LEDs** to the left of J19 convey information about the state of the E681. Their usage is covered in a later section.

The **4 yellow POWER LEDs** indicate the presence or absence of different power voltages on the E681 and are used for troubleshooting. The 2 yellow LEDs at the bottom edge of the board 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 remaining 2 yellow LEDs indicate the presence of 5 volt and 3.3 volt power on the controller. If one or both of these LEDs are not lit the controller is not receiving power:

5V LED lit and 3.3V LED lit	Normal operation
5V LED lit and 3.3V LED not lit	Failure of the on-board power supply
Both 5V and 3.3V LEDs not lit	The E681 is not receiving power.

If both LEDs are not lit, then check the following. If the E681 is being powered from pixel power, and the right-hand pixel power LED is not lit, then check the pixel power source. If the pixel power LED is lit, then most likely fuse F17 is open. **Check for proper configuration of the power select jumpers**, then replace F17. If the E681 is being powered from an external power supply through J19, verify that this supply is functioning and is connected properly.

Pluggable Resistor Networks

There are 8 pluggable resistor networks on the E681 located just above the upper row of pixel string connectors. Each resistor network applies to 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 resistor networks must be in place for the E681 to operate.

Fuses

The E681 will generally be shipped with either 4 amp or 5 amp fuses installed in the 16 pixel string fuse locations. These fuses protect the pixel string wiring in the event of

a short circuit. 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 1st string in cluster #2.

The right-most fuse position is for the power line 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.

Making Connections to the E681

There are 3 basic types of connections required for operation of the E681: 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 E681 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 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. 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 J17 powers the pixels on connectors 3-1 through 4-4, and usually supplies power to the E681 itself as well.

One power supply can power both sides, using a jumper wire between the 2 positive terminals, or you can use two separate power supplies. Figure a maximum current requirement of about 3 amps per typical 50-pixel string at 5 volts. **To power a full load of 16 50-pixel 5 volt 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.** The negative side of the power supply should be connected to earth ground.

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 module is capable of driving pixels of 2 different voltages at the same time. **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.** In other words, don’t plug a 5 volt pixel string into a connector that is wired for 12 volts, 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 E681 contains circuitry to protect against a reversed

connection, it is still possible to damage the controller and/or pixels with improper wiring. **Double-check for correct pixel powering before turning on the power.**

The pixel power supply must be located near the E681, typically within a foot or two. This may mean that the power supply will be mounted outdoors. It is the responsibility of the user to insure that the power supply and the E681 are properly protected from moisture.

Network Connection:

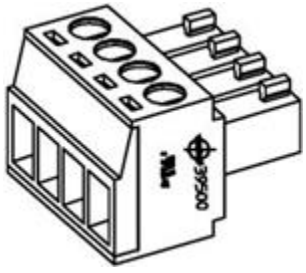
The Ethernet jack on the red Ethernet module must be connected to your LAN, or directly to the LAN card of the PC that will be controlling the pixels. The E681 LAN port is capable of running at 100mb/sec, and is auto-sensing. A crossover cable is not required for a direct connection to a PC.

Pixel Connections:

The pixel strings plug into the 16 4-pin euro-style connectors labeled 1-1 through 4-4. In general, pixel strings can be plugged and unplugged while the system is powered up, but it is recommended, to reduce risk of damage to pixels, that pixel connection/disconnection be done only while the system is off. **If using the GE ColorEffects strings, they will not function properly if plugged in while the E681 is powered up.**

Connecting Pixel Strings to the E681

Mating 4-pin screw-terminal connectors are supplied with every E681. In some cases the module may be supplied with waterproof cable assemblies as well. It is the user's responsibility to make the proper connections from the pixel strings to the connectors on the E681. The mating connectors are illustrated below:



Depending on the type of pixels being used, either 3 or 4 wires will be required between the pixel strings and the E681. All pixels have +V, 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 should be used. It is recommended that the distance from controller to pixel strings be kept less than 20 feet.

You MUST know the correct color-code used by your pixel strings. **This can and will vary from manufacturer to manufacturer**. **Wire color is NOT a reliable indication of function.**

If in doubt please contact the pixel vendor for clarification. **Incorrectly wiring pixel strings can destroy them.**

If you are using the white Sandevices-supplied 4-wire waterproof cable assemblies the color code for those cables is as follows:

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

If you look at the board, beneath the 1-2 pixel string connector, the function of each pin position 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.

The maximum possible distance between board and 1st pixel will vary with pixel type and timing. If pixels exhibit flickering, try shortening the length of wire between controller and pixels if possible. For those that need longer cable runs, the E681 has a feature called Null Pixels. Just insert a single extra pixel in line near the controller, and repeat if needed to reach the desired driving distance. After the 1st pixel, you should be able to go 10-20 more feet between these extra pixels. Then the board can be configured to 'know' how many extra pixels are wired into each string, and the software automatically ignores these, and starts lighting the string at the first real pixel. This feature is described in more detail later.

A future software revision will support a programmable timing option, that will allow you to run pixel strings at slower refresh speeds (still much faster than dmx), where longer string separation is needed, for those pixel controller types that support variable speed operation.

Obviously, you need to be 100% sure that your wiring is correct before plugging in a string, I suggest checking 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 string itself.

Incorrectly wiring of a pixel string may cause irreversible damage to the pixels.

Using the Auxiliary Power Connector, J19

J19 is a multi-purpose connector. If you choose to power the E681 from its own power supply, rather than from the pixel power source, you would connect that power supply to J19. +V to the top pin, and GROUND to the 2nd pin from the top. This power supply should provide from 7-15 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 E681 is powered in this manner, Power Select Jumpers A and D ONLY must be installed.**

The second use of J19 is that it supplies a +5VDC output. This would typically be used to power a small Ethernet switch located near the E681. 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 E681 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 J19, and ground is the bottom terminal. See the screened legend on the board.

Note that J19 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 E681 and the Ethernet switch.

Initial Startup

E681s are shipped with a test pattern enabled. This allows you to verify the proper operation of the E681, pixel power supply, and pixels, without requiring a network connection or a source of E1.31 lighting data.

When powered up with one or more pixel strings attached, the test pattern will ‘chase’ a single bright solid-color pixel through a background of dim pixels. The sequence runs through each of the 680 possible pixels, and will take 30 seconds or so to complete a cycle. **Note that the “as-shipped” default configuration is generally for type 2801 pixels. If your pixels are of another type you will have to re-configure the E681 to the proper pixel type before the test pattern feature will work.**

To configure the controller or operate pixels with ‘live’ data requires a network connection. The recommended procedure is to supply power and a LAN connection to the E681, access the web page to configure the E681 for your particular arrangement of pixels, then connect pixel strings and test.

There are 9 LEDs on the board. The red, green, and blue LEDs in the upper-left are for the ethernet module, BLUE for LINK, RED for SEND, and GREEN for RECEIVE. These LEDs always indicate activity on the network. The RED and GREEN status LEDs along the top edge near J19 are used for displaying E681 status, and for selecting options at system start-up. During normal operation, the green status LED will flicker whenever a DMX data packet is received, and the red status LED will light whenever the web server is running. You can only access the E681 via a web browser when the red LED is lit. References to the red and green LEDs in the following text always refer to the red and green LEDs near J19, not the network SEND and RECEIVE LEDs.

There are 2 pushbuttons on the E681. The pushbutton toward the left edge of the board is RESET, pressing it at any time resets the controller. The button along the top edge of the board is the PROGRAM button, it is used to perform certain operations at start-up, such as forcing the web server to start, or forcing specific network modes.

LED Activity During Startup

During startup, after a few seconds of delay, the green status LED will flash one or more times to indicate the current network configuration:

1) Currently saved IP MODE is DHCP, and an IP address was obtained from a DHCP server.

2) Currently saved IP MODE is Static, and the last STATIC IP address saved into memory page 0 was used.

3) IP address was obtained from DHCP, because it was forced by the pushbutton.

4) IP address was set as the last saved STATIC IP address, because it was forced by the pushbutton.

5) Currently saved IP mode is DHCP, but no DHCP server responded, so our IP address has been set to 169.254.74.73.

7) IP address was forced to DHCP by the pushbutton, but no DHCP server responded, so our IP address was set to 169.254.74.73.

Note that for blink codes 5 and 7 there will be about a 10 second delay before the code is flashed. During this time the system is waiting for a DHCP server.

After this flash code, the red LED will flash 1-4 times, then either come back on again and stay on, or remain off. The final state of the red LED indicates whether or not the web server is running. Note: The web server doesn't operate continuously because the E681 is only capable of 4 simultaneous internet connections. So, although the first 3 connections can be dedicated to receiving DMX dimmer data full-time, the last connection has to be shared between receiving DMX and running the web server.

The firmware as shipped will default to an initial static IP address of 192.168.1.206. If an IP address other than this has been assigned at the factory it will be indicated by a sticker on the CPU chip. If this is incompatible with your network, see instructions below to force the use of DHCP at startup.

Forcing a Specific IP Mode at Startup

It is possible to over-ride the default network configuration, if necessary, using the on-board program pushbutton. At power-up, or after restarting by pressing the RESET button, **press and hold** the program button. After a few seconds, the red and green status LEDs will begin flashing on and off together at a rate of one flash per second. To over-ride, release the PROGRAM button:

After 1 flash to force the web server to start without affecting the network mode.

After 2 flashes to bring up the web server, and force the IP mode to be static.

After 3 flashes to bring up the web server, and force the IP mode to be DHCP.

These functions are in place to allow you to communicate with your E681 via a web browser, regardless of the state it was in. For example, you may have your E681 configured for a static IP address, but you've moved it to a new network where that address isn't available. Or perhaps it's configured for a static IP, but you don't know the saved address. Or perhaps in your configuration you have disabled the web server.

Using the over-ride option at startup, you can force your E681 to enable its web server, and you can force it to get an address via DHCP, or force it to static address 169.254.74.73. Then can you access its configuration page with your browser, change the configuration, and save it.

An alternative method to access the web page for the first time is to connect an ethernet cable directly between a PC (that is configured for DHCP) and the E681, and power both up. At startup, if you're not sure of the saved configuration, or you know it's configured for a static IP, force the E681 to DHCP mode using the pushbutton procedure detailed above. Since no DHCP server will be found by the E681 or by the PC, they will both have assigned addresses in the 169.254.x.x range. After bootup, type 169.254.74.73 in the address bar of your browser and press ENTER. This should bring up the E681 web page. From there, you can use the system configuration commands described later, to re-configure the module to use a static IP address that is available in your network.

It is recommended that you assign each E681 a permanent static IP address on your network. That way you won't have to access your router to find out what the IP address of a particular unit is. Although typically the IP address assigned by your router with DHCP will be the same every time you start up the board, it's possible that it could change from time to time. Also please be aware that at this time the E681 does not support "renewing" of DHCP addresses. Again, in the vast majority of cases this shouldn't cause any issues, however the recommended practice is to use DHCP for initial access to the E681, then to reconfigure it to use a static IP address.

The procedure to select an appropriate static IP address is beyond the scope of this document. Briefly, on a typical network that uses IP addresses beginning with "192", the first 3 number of every IP address will be the same, only the last varies, so we'll just refer to the addresses on the lan by the value of the last number. Typically your router will use "1". 255 and 0 aren't allowed. You also need to avoid the range of IP addresses that your router assigns automatically, see your router's setup pages to learn which addresses these are.

Configuring the E681 to operate with your pixels

Before you can program the E681 to operate your pixels, you'll need to plan the pixel layout; in other words which types and lengths of pixels will attach to which connector on the E681. In order to determine how to attach your pixels, it's important to understand how the E681's pixel outputs are organized. There are 16 pixel string connections. These are divided up in clusters, and there are 4 clusters of 4 connectors each for a total of 16 strings. The first digit of each connector is the cluster number (1-4) and the second digit is the number of the string within that cluster (also from 1 to 4).

It is important to know that the left-hand pixel power connector supplies power to the 8 left-hand string connectors (clusters 1 and 2), and the right-hand pixel power connector supplies power to the 8 right-hand string connectors (clusters 3 and 4).

If you are using 2 different voltages of pixels, you MUST make sure that each pixel string is plugged into the proper 'side' of the E681.

The significance of clusters, is that all string plugged into the same cluster **MUST** be identical in many ways, since many of the E681s programmable options are done "per cluster". For example, every string in a given cluster must use the same type of controller chip, they must be the same length, etc. Please keep this in mind when planning how your strings will be organized. The concept of a 'cluster' of strings will become clearer when we look at how the E681 is configured. Also note that starting DMX addresses are assigned by cluster. In other words you tell the board what the DMX address will be for the 1st pixel on a cluster, and all of the other pixels in that cluster will follow in order.

In most cases the capabilities of the board will be more than adequate, but bear in mind that not everything is possible. For example, if you want to use 5 volt 6803s, 12-volt 6803s, 5-volt 2801s, 5-volt 1804s, and 5-volt GE pixels all at the same time, you can't do it, because that's 5 different types, each would require its own cluster, and we only have 4 clusters available.

Some guidelines on planning the pixel configuration:

Plan on connecting pixel strings in order, beginning with the first available connection with the proper voltage. Strings that must light sequentially, in other words typically all of the strings in a given display element, such as a mega-tree, should be plugged into consecutive connectors, possibly spanning more than one cluster.

Typical pixel strings are 50 pixels in length, and 50 doesn't divide evenly into 681. If using 50-pixel strings, the suggested arrangement is to use 3 strings per cluster, and start each cluster's DMX address with the channel 1. This will use 450 DMX addresses of each universe.

Also please keep in mind that a single E681 can control a maximum of 680 individual pixels, or groups of pixels. Understanding pixel grouping is important because it can allow your total number of pixels to be larger than 680. For example, if some elements of your display don't require individual control of every single pixel, by grouping pixels you can reduce the number of DMX channels needed for that display element, freeing up channels for another display element that perhaps does require control of every pixel.

Using grouping, you can have a single set of 3 DMX channels control as many consecutive pixels as you like, a few, many, even an entire string or multiple strings. For example, if you're putting 300 pixels on your roofline, but you can get by with controlling them in groups of 4 pixels (say 1 linear foot of pixel string), then you have 'saved' enough DMX channels to control 225 more pixels or pixel groups. Or if you have a wreath, candy cane, etc that will always light a solid color, you can control all of those pixels with just 3 DMX channels.

Once you have a plan, you're ready to access the E681s built-in web server to enter your configuration information. Configuration is pretty easy, so if you get started and realize you need to re-think your arrangement, it's no big deal.

Accessing the Web Configuration Page:

Please be familiar with the information listed earlier regarding IP addressing, particularly "Forcing a Particular IP Mode at Startup". Using the techniques described there, you should be able to know the IP address of your E681, either from your router's DHCP client table, or if no DHCP it'll be 169.254.74.73.

Make sure the red status LED stays on after the startup sequence. If it doesn't you'll need to use one of the over-rides to force the web server to come on. **You can't access the board with your browser if the red status LED is off.**

Type the E681s IP address into your web browser's address bar and press ENTER. This should being up a web page similar to this one:

SanDevices E1.31 Pixel Controller Model E680

CPUs Used: Main E1.31 TIMER Array PixA1 PixB2 PixB3 PixA4

Current Network Mode is: Normal STATIC
IP Address: 192.168.001.206 Subnet Mask: 255.255.255.000 Gateway: 192.168.001.001 DNS Server: 192.168.001.001
Mac Adrs: 4A:49:4D:C9:4F:F1
System Up-Time: 0000:20:49 Firmware Version: 2.020 Web Timeout in: 300

E1.31 Packet Statistics:

	Socket s1	Socket s2	Socket s3	Socket s4
Universe Number	1	2	3	4
Packets Received	32,234	32,224	0	0
Sequence Errors	31	34	0	0
Invalid Packets	0	0	0	0

Global Configuration:
Static Network Information:
IP Address: 192.168.001.206 Subnet Mask: 255.255.255.000 Gateway: 001.000.000.000 DNS Server: 192.168.001.001

Dynamic Page Select Address: OFF Last Firmware Update Status: None Tried
Default IP Mode: STATIC Web Srvr Mode: Boot+Auto No Data Timeout: Disabled Test Pattern: 00

Pixel String Configuration:

Cluster#	Strings	Chip	Pixels	Grp	Str Len	RGB DMX Address Range	Reverse Zigs	Null Pixels	Refresh	Gamma
1	3	WS2801	050	001	0050	RGB s1-001 thru s1-450	NNNN	0000	0-0-0-0	0212	
2	3	TLS3001	050	008	0400	RGB s2-001 thru s2-450	NNNN	0000	0-0-0-0	0010	2.0
3	3	TLS3001	050	001	0050	RGB s3-001 thru s3-450	NNNN	0000	0-0-0-0	0081	2.0
4	3	WS2801	050	001	0050	RGB s4-001 thru s4-450	NNNN	0000	0-0-0-0	0212	

Current Page is: 3

Command: no 15

Note: The illustrated screen capture is for firmware version 2.20 for a modelE680 controller. The E681 will show 'E681' in the title line. Other versions of the firmware may have slightly different page layouts.

Once you have the web page displayed, put your cursor on the command box and type this command: QUIT 999. This tells the web browser not to time out. If you don't do this, the web browser may (depending on how it's configured from the last setup) timeout after 5 minutes. The "Web Server Timeout in" entry on the web page tells you the seconds until the server will timeout.

Note: The E681s web page is ALWAYS static, in other words it NEVER updates by itself. It will stay as-is until you either hit refresh or type in a command. The times and statistics shown on the page are as of the last time the page was displayed.

What's Displayed on the Web Page:

First we'll talk about the information that you see on the web page, and what every entry means. Then we'll talk about the commands that you can enter to change that information.

The line beginning with "Current Network Mode is" displays information about the current network connection: the IP address, the subnet mask, the gateway, and the DNS server. If this is a STATIC IP, this will be from information that was previously saved in the E681s memory, if this is a DHCP address, it will be information that was assigned by the DHCP server on your network. Network mode will be "Normal" or "Forced", followed by "Static" or "DHCP", and optionally, "Failed". Normal means the mode, static or DHCP, was selected based on the saved configuration. Forced means the current network mode was forced using the pushbutton at startup, over-riding the stored setting. DHCP or Static shows the type of address currently in use, and failed means the E681 tried to use DHCP, but no server was found.

DHCP server, if not 0.0.0.0, is the IP address of the server that gave the E681 an IP address, usually it will be the IP address of your router. MAC address is this boards unique MAC address, it will always begin with 4A.49.4D. DHCP Lease Time Remaining shows the time before the IP address given to us by the DHCP server 'expires'. Please note: There are some limitations of the DHCP capabilities of the E681 at present, and one of them is that it will not attempt to renew a lease. Once it has obtained an IP address it will keep that address until the next reboot. Once again, the preferred method is to assign every E681 a unique STATIC IP address. Life will be simpler.

System up-time is hours, minutes, and seconds since the last restart. Also displayed on this line is the version number of the E681 firmware, and the remaining time in seconds, until the web server shuts down. If the latter shows as "---" it means the server won't time out.

The next few lines show statistics of the E1.31 data that has been received, and which DMX universes are assigned to the available four internet connections, or "sockets". The 1st line is the "socket", or connection number. There are 4 sockets on the E681, numbered s1 through s4. The next line displays which DMX universe is assigned to each socket. These numbers may range from 1 to 63999. Any valid universe number may be assigned to any of the 4 E1.31 sockets. Normally they would be consecutive, eg, 1,2,3,4, but any assignment can be made. Note that the four assigned universe numbers should be unique, do not use the same universe number twice.

The next line shows the total number of E1.31 packets received since the last restart, followed by the number of 'missed' E1.31 packets (sequence errors), and the number of improperly formatted E1.31 packets. Skipped packets may occur on occasion when accessing the E681 via the web browser. Large numbers of skipped packets would indicate a problem at the source, or a network problem. An occasional skipped packet isn't significant, as this is repeated at about 40 times per second.

The next section of the web page is for global configuration data, in other words all configuration options that aren't related to the pixel strings themselves. This is where the static IP information is entered: IP address, Subnet Mask, Gateway, and DNS Server. Other entries in this section include:

Dynamic Page Select Address: This allows the controller to switch between various configurations based on DMX data received. This is a specialized feature discussed in detail later.

Last Firmware Update Status: The E681 (with firmware versions 2.10 and above) can update its own firmware over your network. This entry shows the result of the last firmware update.

Default IP Mode: Determines whether the normal system network mode will be STATIC or DHCP.

Web Server Mode: This defines when the web server will be operational. One of the limitations of the Ethernet module used in the E681 is that it can have a maximum of 4 simultaneous connections. Normally these are all used for DMX data, hence the number of available DMX universes is four. In order to operate a web server, one of those four channels must be temporarily reassigned as a web server. During the time the web server is up, DMX data may only be received on the first three universes.

The specific Web Server Modes are:

> Button: Server does not ever start. If you select this mode and save it, the only way to restart the web server is to use the pushbutton over-ride at start-up.
>
> Boot: Server comes up for 5 minutes after any system restart.
>
> Auto: Server comes up after a loss of incoming DMX data on socket 4. Every few seconds socket 4 will alternate between looking for web connections and looking for DMX data.
>
> Boot+Auto: Both boot and auto modes.

No Data Timeout: (only supported in firmware versions 2.20 and later)

This specifies how long to wait before turning off the pixels in the event that E1.31 data is no longer being received. A value of 0 means DISABLED, and no automatic shutoff occurs. A value other than 0 (1-999) means the controller will shut off all pixel output if no data is received for the specified number of seconds.

Test Pattern: (Only supported in firmware versions 2.20 and later)

Test patterns may be used to verify proper pixel operation without having to connect the controller to a source of E1.31 pixel data. They may also be used to verify that pixel colors are correct, and that pixels are responding in the proper order.

0 means no test pattern is displayed, this is the normal operating mode. If a value other than 0 is entered, the pixels will no longer respond to incoming data. Values 1 through 6 are presently defined as test patterns:

1 All pixels full on RED

2 All pixels full on GREEN

3 All Pixels full on BLUE

4 A bright RED pixel chases through a dim RED background

5 A bright GREEN pixel chases through a dim GREEN background.

6 A bright BLUE pixel chases through a dim BLUE background.

Note that all test patterns will activate every possible pixel.

Important: Make sure to turn test patterns off by entering in a value of 0. Any non-zero value will disable normal pixel operation. As of Sep 12, 2011 the default will be to ship eeproms and assembled boards with test pattern #4 enabled to allow testing of the E681 without a source of E1.31 data.

The next portion of the page entitled **"Pixel String Configuration"** is where the pixel strings are defined. As previously mentioned, strings are divided into four clusters of up to four strings each. All strings within a given cluster must have the same operating voltage, controller chip type, RGB mode, length, and grouping.

The first column is simply the cluster number, 1 through 4. The second column is the number of active strings on that cluster, this can range from 0 (no strings) through 4 (maximum). Next is the chip type. This defines the type of pixel controller chip used by these strings. Examples would be 2801m 6801, GE, 1804, etc. Next is the number of pixels. If controlling every pixel individually, pixels will be the same as the length of the string. If controlling pixels in groups, pixels multiplied by grouping will equal the length of the string. The number of DMX channels assigned to each string = pixels X 3.

Grp is the grouping factor. This will be 1 if we are controlling each individual pixel. If more than one, then each set of 3 DMX channels will control that many consecutive pixels on the string. String Length is calculated and displayed automatically depending on the values entered for pixels and grouping.

RGB defines the order in which the 3 color channels appear on the DMX channels. Ordinarily this will be RGB, meaning red first, followed by green, then blue. For whatever reason some pixel strings use BGR instead. Who can figure those Chinese? In any case this allows you to handle ANY possible combination that you may encounter. To the controlling software, every string will appear to be R first, followed by G, then B.

The next entry is the range of DMX address for this cluster. Starting DMX addresses are entered as U-CCC, where U is the universe number, 1-4, and CCC is the channel number within that universe 1-510. Why not 512 you say? Well, pixels aren't allowed to overlap universes, in other words all 3 DMX channels for a given pixel must be within the same universe. If a pixel started at address 511 or 512, it would overlap to the next universe. So just pretend that channels 511 and 512 don't exist. Also please remember that the universe number entered here is always 1-4, and corresponds to one of the four

actual physical universes that are assigned to the four DMX sockets. So think of the universe number here as meaning the 1st, 2nd, 3rd, or 4th universe used by the controller, rather than being an actual physical universe number. **Physical universe numbers are set in the E1.31 Packets section.**

The ending DMX address for this cluster is calculated automatically, you cannot change it. It may be in a higher universe, depending on the starting address and the number of DMX channels this cluster needs. The number of DMX channels needed by a cluster is (# of strings) * (# of pixels) * 3.

Reverse is four Y/N values, one for each string in this cluster. A Y means that the string is reversed, ie, it is driven from the 'far' end. Sometimes it's advantageous to be able to do this. For example, if you are lighting a roofline and want to use two 128-pixel strings. That would give you a total length of > 60 feet. The easiest configuration is to mount the controller at the midpoint of the 2 strings, that keeps each string connection short. But to do that, and for it to work properly as a single long string, the controller needs to know that one of those strings is "reversed". That way as you light up pixels 1-256 in order, they will light up from one end of your roofline to the other, not from the middle to one side, then from the middle to the other.

Or if you want to connect your mega-tree strings at the top, but your sequencing assumes pixel#1 is at the bottom, just define all strings as reversed and you are good to go.

Zigs. Well this one can be a bit confusing. Let's say you're doing a mega-tree, and you want it to be 50 pixels high, but your strings are 100 pixels long. If you light pixels 1-100 in order, you want to light the first strand from bottom to top, then the 2nd strand from bottom to top. But if you go up and down with your 100-pixel strings that's not what will happen. You'll light the first strand from bottom to top, then the 2nd strand will light from top to bottom. To fix this you tell the controller that your strings 'zig' after 50 pixels. The Zigs value applies to every string in this cluster. Set Zig to 0 if not used, Zig=1 is the same as Zig=0. Normally Zigs will divide evenly into the length of the string. For example, you may have a string length of 100, and a zig of 25 or 50. If you wanted a zig of 33, define the string as 99 pixels long instead of 100, and the last pixel won't be used.

The final configuration entry for strings is Null Pixels. The purpose for this option is due to the fact that there is a limitation on how far a string can be from the controller. This in turn is due to the fact that the signals that drive the pixels simply aren't intended to be sent over a long piece of wire. The exact limitation depends on the type of controller chip, and other variables, but can range from a few feet to maybe 20 feet. So, what if you need to have some strings that are farther away from the controller than is normally allowed? Well, the neat thing about these pixels is that each pixel 'regenerates' the data signals, allowing them to travel up to another 15-20 feet or so. So, what we can do, is make an "extension" cable that has a single pixel wired in every 15 or 20 feet. Say we need to drive a string that is 100 feet away. Our 100 foot extension cable would have perhaps 5 to 7 pixels wired in it, spaced every 15 feet or so. But, if the controller doesn't know we are doing this, it will just light up those pixels as if they were the 1st pixels of the string. By telling the controller that these strings have 'null pixels' the controller knows to leave those pixels dark, and start lighting up pixels where the real string starts.

The final two display items in the pixel string configuration area are Refresh and Gamma.

Refresh is a calculated value, and indicates approximately how many times per second the pixels in that cluster will be refreshed. Gamma represents the gamma correction value, at present it is only used for the TLS3001 pixel type.

Immediately above the command box is a line that displays the number of the current memory page (0 thru 7) and a warning if changes have been made to that page but not yet saved.

The command box is where we type in commands to modify the setup. After modifications are made, the current configuration data can be saved to any one of 8 memories to be recalled later.

That sums up the information displayed on the web page. Now we'll look at the commands that you can use to configure the system to your needs.

Configuration Commands

Every command consists of a command word followed by 1 or more numeric values. Only the first two letters of the command word need to be entered, more won't hurt, but only the first two are checked, so "GROUP" is the same as "GROPE". Upper or lower case may be used.

Numeric commands must be separated from the command word, and from each other, by one or more characters that aren't digits. Values are always positive integers. The following commands would all be interpreted the same:

Strings cluster 4 quantity 3

ST 4 3

ST 4.3

ST 0004-003

We'll divide the commands into groups:

First we'll look at the commands that allow you to load and save the configuration data.

There are 8 memory pages in the E681, numbered 0 through 7. Each page stores the entire board configuration, from networking information, to pixel information. By having multiple pages, you can have more than one saved configuration that you can recall easily.

When the system starts up, it <u>always</u> loads memory page 0. So, whatever configuration is saved here is what will run the lights. When you make configuration changes using the web commands, those changes will be shown immediately on the web page, but they are not automatically saved to system memory, so they won't be immediately shown on the pixels.

Only the SAVE command does that, it stores the configuration data shown on the web page, in one of the 8 memory pages. So please remember, after making changes, SAVE them.

You will see a warning message reminding you if there are unsaved changes.

The multiple memory pages are handy if you want to experiment with different commands, but don't want to disturb your basic configuration. First save the current configuration into another memory page, say page 1. Then play around as much as you like with the configuration. To see the effect of your changes on the pixels, type SAVE 0. When finished, if you want to go back to your original setup, just LOAD 1, then SAVE 0, and you are back where you started. **Remember: the pixels will ALWAYS operate on whatever configuration was last saved into page 0 unless you have dynamic page selection enabled.**

SAVE n	where n is a memory page number from 0 to 7
	Writes the currently displayed configuration to the specified memory page
LOAD n	where n is a memory page number, 0-7
	Loads the specified memory page and displays the information on the web page.

The BOOT command forces a restart of the E681 (if BOOT 999).

BOOT 999	will restart the system, make sure you SAVE first if you have made any changes!

The QUIT command is used if you have finished making (and saving!) changes, and want to shut down the web server to allow socket 4 to resume receiving DMX data. One gotcha: If your last command was QUIT, and you later bring the web server back up and press F5 or 'refresh' on your browser, it will send the same command (quit) again. Be aware of this.

Normally you would use QUIT when the web browser isn't running continuously. If socket 4, for example, is hunting between looking for DMX data and looking for web 'hits', your first access to the page will bring the web server up with a 5 minute timeout. This automatic timeout is there in case you forget to quit the browser manually. So, first use the QUIT command as QUIT 999 to 'lock' the browser on. Make you changes, save them, then type QUIT. The browser will shut down 10 seconds later and you're back receiving DMX on all 4 sockets.

The value used with the quit command can't be less than 10. This is to give you time to re-enter a different quit command if you enter a quit command in error.

QUIT n	Quit tells the web server to stop running. If no
	Number is entered, it will stop in 10 seconds, if
	10-998 is entered it will stop in that many seconds.

QUIT 999 will keep the web server running forever.

The **UNIVERSE** command is used to map physical DMX universe numbers to the E681's four Ethernet connections, or sockets. This information is displayed in the E1.31 Packet Statistics section of the web page.

In a simple configuration, the first E681 might be assigned to universes 1-4, the 2nd to 5-8, etc. But these assignments can be made using any four universe numbers, and universe numbers can be as high as 63,999. **Important: please make sure that there are no duplications,** in other words don't assign the same universe number to more than one socket. It won't work. Command Example:

UNIVERSE 1 1000 this command would assign DMX universe #1000 to socket #1.

The following group of commands set the configuration data displayed in the Global Configuration portion of the web page.

Every configuration page has an area to store a static IP address. Even if you set the network mode to DHCP, you can still save static information. This eliminates the need to re-enter the static information every time you switch from DHCP to static. There are 4 entries here, Static IP, Subnet Mask, Gateway, and DNS Server. Usually, DNS and Gateway will be the same, the address of your router. In most small LANs this will be either 192.168.0.1 or 192.168.1.1. Subnet mask in a '192' LAN will be 255.255.255.0, or just 24. The commands are:

IP a.b.c.d where a.b.c.d is any valid IP address

SUBNET a.b.c.d or SUBNET n where a.b.c.d is any valid subnet mask value, or n is the size of the subnet in bits.

GATEWAY a.b.c.d where a.b.c.d is any valid internet address. Usually the address of your router.

DNS a.b.c.d where a.b.c.d is any valid internet address. Usually the address of your router.

If you are only using DHCP (not recommended), you can leave the static IP areas unused.

Network addressing (static or DHCP IP) is only needed to access the E681's configuration page with a web browser. Once configured, no IP address is needed for normal operation.

The Default IP Mode, Web Server Mode, No Data Timeout, and Test Pattern commands affect what happens when the system starts up.

DEFAULT n where n is 0 or 1. DE 0 sets a default network mode of STATIC, DE 1 sets a default network mode of DHCP. (corrected 06/21/2011)

WEB n where n is 0 to 3. 1=start server for 5 minutes on every Restart, 2=start server if no DMX data received on slot 4. 3 is both of the above. 0 means never start the web server.

NO n where n is 0-255. If non-zero, after n seconds of no DMX Data being received, pixels will go dark. (vers 2.20 and higher)

TEST n Where n is a test pattern number, or 0 for OFF. (versions 2.20 and higher)

The final group of commands is where you enter the configuration information for your clusters of pixels. We will discuss these commands in the order the information appears on the web page.

All of these commands will have more than one numeric value. The first numeric value is always the cluster number, from 1-4. In the command descriptions that follow, the designation 'n' is always used to represent a cluster number 1-4.

The **strings** command defines how many pixel strings are attached to a particular cluster, this may range from 0 (none) up to 4.

STRINGS n a Define number of strings attached to cluster n, 0-4
Example: STRINGS 1,4 says there are 4 strings on cluster 1.

The **CHIP** command tells the E681 what type of pixel controller IC, or chip, is used on the pixels that will be attached to this cluster. Enter 2 numeric values, first the cluster number, then the chip number. The following chip types are defined:

0=6803, 1=2801, 2=GECE (GE ColorEffects), 3=1804 (fast), 4=1804 (slow), 5=Native DMX, 6 and 7 not presently defined, and 8=TLS3001 12-bit pixels.

CHIP, n, x where x is a valid chip number 0-8. Defines the type of chip used on cluster n's pixel strings. Example:
CHIP, 1, 0 (sets chip type to 6803 for cluster 1)

The **pixels** command sets the number of pixels assigned to each string of this cluster. A pixel here refers to a set of 3 DMX channels. A pixel may light more than one light on the string if the group size is more than 1. If the grouping is 1, then # of pixels = string length. If the grouping is more than 1, then Pixels X Group Size = String Length. For example, if you have a 50-count string, and will be lighting up LEDs in groups of 5, you would set pixels to 10, group to 5, and the string length will be automatically calculated as 50.

PIXELS, n, aaa Defines the number of pixels on each string in cluster n
Example: PIXELS 1, 50

The **Group** command tells the E681 how many consecutive lights on the string to light up for each set of 3 DMX channels. Normally this will be 1, meaning you are controlling each LED individually. This gives the maximum possible flexibility, but at the expense of number of DMX channels used. Using pixel grouping, in situations where control of every pixel isn't needed, can reduce the DMX channel count significantly, allowing a

single E681 to control more pixels. For example, if you are lighting a single display element, such as a wreath or mini-tree, you may want to use pixels to be able to select any color, but you may not need control of every pixel on that item. You might group the entire string as 1one group, thus being able to light that display element with only 3 DMX channels.

GROUP, n, aaa — Defines the pixel grouping for strings in cluster n to be aaa pixels. Example: GROUP 1, 1 says that all pixels in cluster #1 will be controlled individually.

String Length is calculated automatically by multiplying pixels X grouping.

The **RGB** command defines the order in which the 3 color channels are assigned to the pixels in a given cluster. Most pixels use the same sequence that the name RGB would imply: RED first, followed by GREEN, then BLUE. Some pixels however assign their color channels in a different order. BGR is fairly common. To cover all possibilities, the RGB command lets you define the RGB pixel order of a cluster's strings to be any of the 6 possible combinations, using a number from 0 to 5. The numeric codes, and the sequence they assign, is as follows: 0=RGB, 1=RBG, 2=GRB, 3=GBR, 4=BRG, and 5=BGR.

RGB, n, a — Defines the pixel color order for strings in group n RGB 1,5 would define cluster #1 pixels to be BGR.

The **DMX** command sets the starting DMX address for the pixels attached to this cluster. Only the start address is specified. All pixels on the cluster (every pixel in every string) will then be assigned sequential DMX addresses, beginning with the specified starting address. In some cases the ending DMX address will be in a different universe, this is allowed. The number of DMX addresses used by a cluster will be equal to:

(number of strings) X (number of pixels per string) X 3

Starting DMX addresses are entered in the form of s-ccc, that's a slot number (1-4) followed by a channel number (1-510). In other words, if we enter our DMX command as: "DMX 1, 2-101" what we are saying is that the DMX addresses for cluster #1 will begin with the 101st channel of the DMX universe assigned to our 2nd slot. Now, if you happen to have assigned slots 1-4 to DMX universes 1-4, then slot# and DMX universe number will in fact be one and the same. Just to be clear, you first assign DMX universes to SLOTS, then you assign SLOTS to CLUSTERS. Ending DMX addresses for a cluster are displayed, but you can't change that value. It's possible for DMX address ranges to overlap, this may or may not be intended.

Now, the question that's gnawing at some of you, if there are 512 channels per DMX universe, why can a starting address not be more than 510? Because each pixel takes 3 channels (3 dmx addresses), and all 3 of those channels must be in the same universe. In other words, a single pixel's DMX addresses can't span a universe boundary. The reason for this is that different DMX universes are updated at different times. If a pixel spanned a universe boundary, then there would be a short period of time when part of the pixels channels had updated, but not all. This would produce a brief color change in that pixel. So just pretend that channels 511 and 512 don't exist. The E681 will never use those 2 channels.

DMX, n, s-ccc Defines starting DMX address for 1st pixel of cluster n as the universe assigned to slot x, channel CCC.

The DMX ending address is automatically calculated and cannot be changed with a command. It is a function of the starting address, the number of strings, and the number of pixels per string.

The **reverse** command defines which, if any, of the strings in this cluster are reversed. A reversed string is one where the pixel nearest the controller is considered to be the last pixel in the string, and the farthest-away pixel is the first. It is used when it is more convenient to attach a pixel string to the controller at what would normally be the 'far' end. Although reverse is specified on a per-string basis, it is entered a s a single numeric value obtained by adding the reverse values of all strings that are reversed. String #1 is 1, String #2 is 2, String #3 is 4, and String #4 is 8. For example, if you want strings 1 and 4 to be reversed, you would enter a reverse value of 9 (1+8).

REVERSE, n, a Defines reversed status (Y/N) for each string in cluster n, value (a) is the sum of the reverse values for all reversed strings. REVERSE 1,15 would indicate that every string in cluster #1 was reversed. REVERSE 2,0 would mean that none of the strings in cluster 2 are reversed.

The **zigzag** command is typically used when you are building a matrix of pixels, and the height of your matrix is less than the string size. Say, for example, you want a 25 high x 48 wide matrix of pixels, and your pixel strings are 100 lights long. The easiest way to construct this matrix is to start the 1st srring from bottom to top, reverse after 25 pixels and come back down, then reverse again every 25th pixel. So one pixel string is actually 4 columns of your matrix. The problem is, when the controlling software in the PC starts lighting up pixels, we want them to light the 1st row from top to bottom, then the 2nd row from top to bottom, etc. But the way we have it strung, what will actually happen is that the 1st column will light from bottom to top as expected, but the second will light from top to bottom, and this pattern will repeat.

The zigzag command allows you to tell the E681 that you have hung these strings in a zigzag pattern. The controller then automatically re-arranges the addresses assigned to the pixels so that they light in the proper order. In this example we would use a zigzag of 25, meaning the pixels reverse direction every 25th pixel.

The zigzag command is per cluster, in other words this zigzag factor applies to every string in the cluster. Zigzag of 0 or 1 has no effect.

ZIGZAG n, a Defines all strings in cluster n to zigzag every a pixels.

(Please note that SOME sequencing software also has the ability to define string as being in a snake or zigzag pattern. If you are using software with that capability, then you can define the zigzag either in the sequencing software, OR in the E681, but not both at the same time.)

The Null Pixels command defines, on a per-string basis, if there are any extra pixels in line before the start of the 'real' string. Null pixels are a method of controlling

strings that aren't close to the controller. Normally, there is only a fairly short distance allowed between the controller and the pixel string. If you need to mount some strings at greater distances, you can make up an extension wire that has a single pixel built in every 15 feet or so, starting near the controller. These pixels act to 'regenerate' the data signals so that they are good for another 15-20 feet. For example, by using about a half-dozen pixels in this fashion, you could control a pixel string that was perhaps as far as 100 feet from the controller. The **Null Pixels** command lets the controller know that these pixels are there. Otherwise they would light up as the first few pixels of the string, not what we want. This way the controller knows to skip those pixels, and start lighting the string at the proper point.

Although null pixels is entered for each individual string, the values for all 4 strings of the cluster must be entered in a single command: NULLS 1 0-0-3-3 would define the 3rd and 4th strings of cluster 1 as having 3 null pixels each.

NULLS, n, a,b,c,d Defines the number of null pixels on each string of Cluster n. Example: NULLS 1, 0,0,0,0

Null Pixels, Zigzag, Reverse, and Grouping may be used in any combination. You could, for example, have a string that has 4 null pixels, is lit in groups of 5, zigzags every 25, and is driven from the far end.

GF This command is used to load a new firmware file over the Network. Refer to separate firmware update documentation.

GV, a.b This command sets the gamma correction value.

Gamma correction is only applicable to 12-bit pixels such as the TLS3001, and is used to make the dimming response of the pixels more like incandescent bulbs. A typical value would be 2.0, acceptable values are 1.0 to 3.0 in steps of 0.2.

DP, n Enables dynamic page switching. N from 0-512, 0=off.

Dynamic page switching is an advanced feature. This allows the controller to switch between different configurations "on the fly" based on a value that's sent over a specified DMX channel. When this feature is made active, by entering a value between 1 and 512, then the DMX value that is received on that channel of the first universe is used to select one of the 8 configuration pages. The page selected is the dmx value received divided by 32:

DMX Value	Page Selected
0-31	0
32-63	1
64-95	2
96-127	3
128-159	4
160-191	5

192-223	6
224-255	7

Before enabling dynamic page selection it is important that you make sure that you have the desired configurations loaded into any memory pages that you will be using. **This is an advanced feature that should only be used once you are comfortable with the basic controller operation.**

This concludes the command definitions. Remember, changes aren't saved until you do a SAVE command. Changes won't affect the operation of the strings until you do a SAVE 0 command.