An Introduction to RGB Pixels.

Jan 18, 2011	Original Date
Jan 19, 2011	Added info re waterproofing and UV resistance
	Added basic information about common controller chip protocols
	Added caution concerning wiring color codes

Being presently involved in the design of a controller for RGB pixels, I often see emails from users that want to get started with pixels, but don't really underatand exactly how it all works and what's required to come up with a working RGB pixel system. So I've put together this document as a starting point for those needing some info on the basics of RGB pixels.

An RGB Pixel is a single light source that can be lit in any intensity and in any color, because it has all3 primary colors built in, red, green, and blue, hence the name RGB. Typically RGB pixels come in strings similar to normal Christmas lights. In strings, they are a bit larger than standard LEDs or incandescent mini-lights, because each individual light needs a small circuit board attached, with an integrated circuit that controls the red, green, and blue LEDs. These individual LEDs are spaced very close together so that to the eye they appear as one single light source. By varying the intensity of those 3 LEDs, we can make that light source appear to be any color.

Think of your flat screen TV, or computer monitor. The picture is made up of many RGB pixels, each a tiny dot that can be lit in any color.

Although large RGB flood and spot lights are in effect RGB pixels, here we're focusing on the smaller pixels that typically come in strings similar to incandescent mini and LED Christmas light strings.

Although strings of pixels are the most common, they are available in many forms such as flexible strips of pixels,and pixels that come pre-mounted in rigid metal or plastic channels. They come in many sizes as well.

The advantage of using RGB pixels for a lighting display is that you have control over the color and intensity of every single pixel individually. Instead of having to light up an entire string of LEDs at a time, you can light up only those pixels that you want. Instead of having to use multiple LED strings to allow a selection of colors, you can use a single string of RGB pixels, and light it in any color of the rainbow. The pre-automation style of Christmas displays, the 'static' display, was either all on, or all off. Then we started to get fancy, with the ability to turn individual strings of lights on and off, and to dim them as well. Well the step between today's string-based displays and displays based on RGB pixels, is even more dramatic than the step from static displays to control of individual strings. The potential capabilities o RGB-based displays are incredible. For example you could have a 'wall' of RGB pixels that you could use to display images, either still or animated, even full-motion video.

Although the technology isn't new, it's only recently that the cost of the pixels, and the controllers to operate them, have dropped to affordable levels. Affordable is a relative term, I suppose. Many of us who do exotic lighting displays for the holidays tend to spend more than most of the population would think reasonable. But hey, it's a hobby, right?

The least expensive pixels that I'm aware of, at least as of January 2011, run about $0.60 to $0.70 each, plus shipping, from China. Right now China is the least expensive source for RGB pixels. From there, prices can go up significantly, it all depends on the style you want, whether they are made from UV-resistant wire, ruggedness, degree of waterproofing, etc. For a full setup, for the do-it-yourselfer, figure at least $1 per pixel or so for everything, pixels, shipping, controller, power supply, cabling, etc.

In addition to the pixels you need a few more things. First you need a controller. Pixel strings aren't like standard Christmas light strings, in that you can't just plug them into a wall outlet. They operate off of DC, typically 5 volts is most common, some use 12 volts, some up to 24 volts. Besides power, you obviously have to have some sort of control device attached to the strings to tell them when and how to light up, that's the job of the pixel controller.

There are two basic types of pixel controllers. The first is the stand-alone controller. These typically have several preset patterns built in, and you select the pattern that you want with a pushbutton. An example of an all-in-one system with a built-in controller and power supply is the GE Color Effects pixels. These were available at Costco this past Christmas, priced at $60 for a 50-pixel string, or $1.20 per pixel. The controller and power supply are included in the package, so it's pretty much plug and play. You can find several videos on YouTube that show the capabilities of these pixels.

Most folks who are into exotic light displays want more control than the stand-alone controllers can offer. They want to be able to control pixels the same way that they control the rest of their 'show', in other words control the pixels from a lighting control software program that's running on a PC. For this, you need a pixel controller that receives lighting control signals from the PC, and converts it into a form that the pixels understand. This type of controller is sometimes referred to as a protocol bridge, because it converts or 'bridges' from one protocol (one of the various standard lighting control protocols) to the protocol used by the pixels.

There are quite a few controllers to choose from, both commercially-built units, and units offered by the do-it-yourself lighting community. Some support only one type of input protocol, some support more than one, some support only a single pixel protocol, some support several. Some will drive only a single string of pixels, some will drive many strings.

Of the various input protocols, some are open standards, while others are manufacturer-specific. One example of a manufacturer-specific device would be the Light-O-Rama Cosmic Color Ribbon. It's a 50-pixel strip with an attached controller that understands the Light-O-Rama protocol. Other manufacturer-specific protocols include Vixen, Renard, and Lynx.

The most popular "standard" lighting control protocol is known as DMX. A single DMX circuit is known as a universe, and it can transmit up to 512 channels of lighting data. Since RGB pixels require 3 channels per pixel, 1 each for red, green, and blue; a single DMX universe can control up to 170 pixels, using 510 channels, with two channels left over. Although the DMX standard specifies a 5-wire "XLR" connector type for DMX, 3-wire XLR connectors are extremely common since with those it's possible to use standard microphone cables to connect DMX gear. Recently it's become popular to use RJ-45 connectors for DMX, to allow the use of inexpensive CAT5 type cables to connect DMX devices together.

One of the realities of working with RGB pixels is that the number of needed channels can get very large very quickly. Let's say you want to do a large RGB mega-tree with 24 strands of 100 pixels. Well 2400 pixels * 3 channels per pixel = 7200 channels of control information. With DMX, that's 15 universes! That's a lot of DMX cables to be stringing, not to mention a lot of DMX interface 'dongles' needed at the PC.

The commercial lighting community faces this same dilemma, and recently a new style of the DMX standard was developed that allows DMX lighting data to be sent over a LAN using standard Ethernet. The advantage is obvious. A single universe of DMX data requires about 250,000 bits per second. Putting it on Ethernet adds some additional overhead, so call it about 350,000 bits per second over Ethernet. But most Ethernet networks can run at up to 100 million bits/second. That's the equivalent of about 300 MDX universes of data on a single wire. Now, practically speaking, 300 is probably optimistic, but 100 universes should be easily do-able.

So instead of 15 DMX interface 'dongles' on your PC, and 15 DMX cables running out to your mega-tree, using DMX over Ethernet you have no need for any special hardware at the PC, assuming your PC already has a LAN card, and you run just a single Ethernet cable out to your mega-tree. As icing on the cake, that Ethernet cable just needs to plug into your LAN anywhere. As long as the controller is plugged into your LAN, and your PC is plugged into your LAN, you are good to go.

Because of these advantages, DMX over Ethernet is likely to be the most dominant 'protocol' for communicating with large numbers of RGB pixels. The simplest of the DMX over Ethernet protocols is known as E1.31, and sometimes called SACN, or Streaming ACN.

Granted, many installations can get by with the standard DMX, or other manufacturer-specific wiring. If you don't need to control every pixel individually, some controllers will allow you to group them, so for example, you might control the pixels in groups of 5. This reduces the required number of control channels by a factor of five. So that same mega-tree, if controlled in groups of 5 pixels, could be handled by 3 DMX universes, and that's not unreasonable to do over standard wired DMX.

The future, however, in my opinion, is E1.31, or some variant. I expect virtually all of the popular 'sequencing' programs to have E1.31 compatibility sometime this year. Some do already of course; Light Show Pro is a popular and very inexpensive package that already has good support for E1.31.

The popularity of RGB pixels is going to require the vendors of sequencing software to add RGB compatibility. Many of the old techniques for sequencing quickly become impractical when the channel counts increase to the levels that RGB requires. Think about it, a 100-light string that you control now as a single channel, or maybe 2-5 channels if you have multiple colors of strings run together, now will need as many as 300 channels using pixels. Another reason why pixel grouping is an attractive option to have!

If you're not interested in the 'how it works' aspect, feel free to skip this part. RGB pixels consist of 2 main parts, a tiny integrated circuit (pixel control IC), and a 3-channel LED. In addition, there are typically a few resistors. These parts mount on a very small circuit board that is a part of each pixel. The LED mounts on one end of the board, and the wires coming in and going out mount at the opposite end. The whole thing is typically potted in clear silicone for insulation, protection, and (to some extent) waterproofing. The addition of the electronic components and the circuit board is what makes RGB pixels 'bulkier' than their standard LED string counterparts.

The pixels are interconnected by either 3 or 4 wires. Two wires carry power, and then there are 1 or 2 wires to carry the lighting data to the pixels. The controller chips built into the pixels are pretty clever, and it's their smarts that allows a long string of pixels to be controlled by 1 or 2 signal wires. Lighting data is sent to the pixels as a series of individual on/off bits. Each bit is represented by a high or low signal on the data wire. The number of bits sent to each pixel varies according to the 'color depth' of the particular pixel control chip being used. Some support 5 bits of level data per color, giving only 32 separate light intensities. Others support 8 bits, giving 256 separate intensity values for each color.

Eight bits of color depth is preferred because of the smoother dimming that can be accomplished when you have 255 available light levels (plus off). An 8-bit control chip then requires 24 bits of lighting data to be sent to it to light a pixel, 8-bits per color X 3 colors = 24 bits. Now here's where the 'smarts' of the controller chip comes in. Each controller chip has a data in wire and a data out wire. You control the string by sending data to the data in wire on the first chip. That chip's data out wire connects to data in on the 2nd pixel. That process continues to the end of the string.

Now here's how it all works. These chips are smart enough to grab the first 24 bits they see as their own data, then, after they have received "their" data bits, they start passing any subsequent bits to their output wire, to be passed to the next pixel. So, say we have 5 pixels in a string. We're going to send 8bits of color depth X3 colors X5 pixels = 120 bits of data to the string to control those 5 pixels. Call them bits 1 through 120. The first chip sees all 120 bits, it grabs bits 1-24 for itself, then starts passing the rest of the bits to the next pixel in line. Pixel # 2 only sees bits 25-120. It grabs the first 24 bits it sees, bits 25-48, for itself, then starts passing on data down the line. The process continues until pixel #5, it only sees pixels 97-120, it grabs them for itself, then we're done. The process repeats over and over.

All RGB pixels operate in some variation of this manner. The beauty is that we don't need to run a separate wire to each pixel (imagine that nightmare), but rather we can control a whole string with just 1 or 2 data wires. Why 1 or 2? Well every pixel needs a DATA wire, and some also need a CLOCK wire. The tradeoff is speed. Pixels that use both a CLOCK and DATA wire can typically be updated at a faster rate than the pixels that use a DATA wire only. In practice, either is fine.

All of these pixels update so quickly that it's faster than the eye can follow, typically an entire string can be updated 50, or 100, or more, times per second.

Here are some popular controller chips and their characteristics:

Chip	Voltage	Wires	Color Depth
6803	5V or 12V	4	5 bits
2801	5V	4	8 bits
1804	12V typical	3	8 bits (dimmer LEDs than 5V pixels, but require more power per pixel)
GECE	5V	3	4 bits (this is the GE ColorEffects chip, it's a bit of an oddball)
3005	5V	3	8-bits (the data protocol for this chip is considered proprietary by the manufacturer)

Data Protocol information for common controller chips added Jan 19 2011:

This is not intended to be all-inclusive, but a decent starting point. Data sheets for all of these chips can be obtained from Ray Wu. All of these operate at standard 5V logic levels but can be driven with 3.3V logic levels as well.

6803: Requires data and clock. 16 bits per pixel. The 1st bit sent is the start bit, always a 1, then 5 Red intensity bits MS bit 1st, followed by 5 Green bits, and 5 Blue bits. Data is clocked on the rising edge of the clock signal. Set data line, wait500 us, set clock high, wait 500 us, take clock and data low. Repeat for all 16 bits. Repeat for all pixels in the string. Then send a word of 16 0's, this marks the end of the data frame. About 1mhz is a good clock rate.

2801: Requires data and clock. 24 bits per pixel. No start bit, just 8 bits per color starting with the MS red bit, ending with the LS blue bit. Same bit format as the 6803. 1mhz is a good clock speed. Pause > 500us between frames.

1804: Data only, no clock. These chips have a jumper option to select between low speed and high speed. The ones that Ray is selling are set for high speed. For low speed one time increment is 1000ns, for high speed one time increment is 500ns. Send 24 bits per pixel, starting with MS red bit, ending with LS blue bit. To send a bit:

Take data line high, wait 1 time increment

Set data line = data bit (1-high, 0=low), wait 1 time increment (note, no change here if sending a 1)

Set data line low, wait 1 time increment (note, no change her if sending a 0)

Repeat for all bits of all pixels. Data line held low for > 10us indicates a break between frames. You must pause at least 10us before starting the next frame. You must NOT pause > 10us during a frame.

3005: I have searched high and low and have found no protocol information on this one. Supposedly it's a manufacturer trade secret. Smart, huh? Guess that's why they're so popular.

GECE: These are the GE color effects chips. It's a strange protocol. Data line only, no clock. I refer you to the massive thread here:

http://doityourselfchristmas.com/forums/showthread.php?13062-RGB-LED-s-Now-Consumer-Grade-Hackable

Basically you send 26 bits, bit format similar to the 1804 except the data polarity is inverted (go low for a 1 and high for a 0). 6-bit physical pixel address (0-63) with address 63 being a broadcast address to all pixels. Next is an 8-bit master intensity value, followed by 3 4-bit color intensities. Difficult to implement well with DMX due to the small color depth.

So, at the starting end of a pixel string you have either 3 or 4 wires to deal with. One wire will be power, a positive DC voltage. 5 volts is very common, 12 volts is also common. A 2nd wire will be ground. Then there will be either 1 or 2 more wires, depending on the type of control chip the pixels use. These wires need to be connected to a controller and a power supply. Some controllers provide the data and clock signals only, so power wiring has to be done separately to each string. In that case, you would connect ground, data, and clock wires to the controller, and ground and +V would go to the power supply. Other controllers provide power and data/clock signals to the strings, so you just have to plug the string(s) into the controller, and then connect the power supply to the controller.

Color Code Caution, added Jan 19, 2011:

One note of caution: There appear to be as many wiring color-codes used on these strings as there are manufacturers. Please be sure you know which wire is which before you hook these things up! If in doubt contact the vendor. You also have to know which end of the pixel string is the start, and which is the end. If the strings come with connectors attached (not all do) the end with the male connector is the start of the string. Don't assume that RED is +V and black is ground, it may not be! I have encountered at least 3 different color code schemes so far, and that's just on the 4-wire pixels.

Speaking of power supplies, that's the other component that you'll need to build a pixel display. Actual power requirements will depend both on the number of pixels, and the specifics of the particular pixels being used. For 5V pixels, figure about 2.5 amps per string of 50 pixels as a minimum. There are other considerations when supplying power to pixel strings. Typically, for 5 volt pixels, 50 pixels is the maximum length of a single string. This is because a small amount of the power to the string is consumed by every pixel, and because of the resistance in the wire, the voltage gradually drops off from the first pixel to the last. To go beyond 50 pixels in a 5-volt string requires that power be connected to the string at more than one point. That is probably beyond the scope of this document, just a heads up.

One method of accommodating longer strings is to use 12V pixels. These are commonly available up to 128 pixels. There are trade-offs though. Total power consumption is greater, and the pixels aren't quite as bright as the 5 volt pixels are. I f you want to build a big mega-tree, the 12 volt pixels might be a good choice, since you could use 128 pixel strings that would be more than 30 feet long. With 5v pixels you would be limited to about 13 feet in length before you have to come up with an arrangement to add additional power to the strings.

Environmental Considerations, added Jan 19, 2011:

There are additional considerations for outdoor displays. The first is waterproofing. Bear in mind these comments refer to the popular Chinese-made pixel strings. First, it's important to understand the original purpose of these pixels. They were designed to be installed in digital signs, pushed in from the back of the faceplate of the sign, in a pretty closely-spaced grid. So, they are relatively waterproof, as in designed for outdoor use, but there originally intended application doesn't subject them to direct rain or snow because they are pretty-much shielded by the sign enclosure.

The weak spot is that there are two sets of wires going into each pixel, two 3 or 4-consuctor flat cables. These two flat cables are just pressed against each other and they enter the pixel through a single hole in the silicone. The problem is that there is no silicone between the two sets of wires, and this not only creates a potential entry point for moisture, with a 'hanging' string, the weight of the string will tend to pull the silicone apart slightly, allowing a larger path for water entry. Some pixels are made from a pretty stiff silicone, others are made from silicone that's a lot more flexible. Those made from the stiffer silicone are probably more inherently waterproof, but those made of the softer stuff do lend themselves to some additional waterproofing by a quick shot of hot melt glue between the two sets of wires. The silicone is soft enough that you can just jam the tip of the glue gun in, forcing the wires apart a bit, then you shoot in a dab of glue, and when you remove the gun the silicone snaps back and seals everything up tightly. I personally ran 40 strings of pixels this year, about 20 of each type. I did do the hot-melt glue waterproofing of the softer pixels. All strings survived the season.

The second consideration for outdoor use is UV resistance. Some strings claim to be UV resistant, other don't. To be honest I'm not sure how much faith you can place in those claims, or waterproofing claims for that matter. The Chinese manufacturers seem to be willing to claim anything if it'll improve their shot at a sale. As far as UV, I honestly don't know how critical this is. For Christmas displays, I'd say UV is more of an issue for those in the Southern hemisphere with Northern exposure, and probably a lot less significant for those in the Northern hemisphere. I'd say the jury's out on this until we have more real-world experience. I will say that mrpackethead of response-box.com is a major believer in UV, and pays a pretty hefty premium to have his strings made of UV-rated wire. RJ, of diylightanimation.com made the opposite choice, going with non-UV-rated wire due to the cost differential.

So, to sum it up, here's what you need to build an RGB pixel display:

Some pixel strings, length, type, and quantity according to your application.

One or more pixel controllers.

One or more DC power supplies.

A data cable or cables (Ethernet, DMX, or manufacturer-specific, depending on controller type)

One or more interfaces at your PC. This could be as trivial as nothing (if using E1.31), up to one or more 'dongles' (a short wire with a small module that converts from USB to the lighting data protocol).

Sequencing software that has RGB pixel capabilities.

Sources of RGB Pixels and accessories:

As far as the pixel strings themselves, the least expensive pixels are from China. The most popular supplier seems to be a gentleman by the name of Ray Wu, and he has a site on Alibaba. Ray Wu's site is like a candy store for lighting people.

Search google for "Alibaba Ray Wu" or follow this link: www.aliexpress.com/fm-store/701799

I have dealt with Ray many times with good results, his English is fluent, he's pretty prompt at replying to emails, and paying through Alibaba gives you some protection. As you can imagine, shipping from China can be expensive, but it's actually pretty quick. I received one shipment in less than 48 hours. Shipping becomes more economical as the size of the order increases so try to buy as much as possible at one time. I do recommend however that you get a small quantity of whatever you are buying first, to make 100% sure it's what you want. Shipping costs make returns difficult, you're probably better off selling on Ebay that trying to return something to China.

One issue I have had with Ray when buying pixel strings is getting varying spacing between pixels. These pixels weren't originally designed to be hung in strings, ie mega-tree, they were designed to be inserted in holes in electronic signs. In that application, pixel spacing isn't critical, but if you're going to be hanging strings to make a mega-tree you obviously want the spacing on all strings to be the same. So make sure that you make Ray aware that this is important to you.

Ray also has extremely good deals on power supplies, and he can supply 'standard' controllers (1 universe of wired DMX in, and 1 pixel string out) at a pretty good price.

As far as controllers, here's a list of the ones that I'm familiar with. As a disclaimer, the second one from the end of the list is my own project. These are listed, to the best of my knowledge, in the order of development. This is just basic overview information, please follow the links for details.

http://response-box.com/rgb/ This is the web site operated by "mrpackethead". He frequents the popular lighting forums and was the first, afaik, to offer a pixel controller that had roots in the DIY marketplace. His products appear to be first-rate, his pixels use UV-resistant wire and have good waterproofing. He can provide a complete plug-and-play system of pixel strings, cables, power supply, and controller. High quality gear, but fairly pricey. Available now.

http://forums.auschristmaslighting.com/index.php/board,34.0.html This is a project from the folks at the Australian forum, auschristmaslighting.com. It's a pixel controller only, you supply pixels, cabling, and power supply. It used 4 standard DMX universes in, but when used in conjunction with an E1.31->DMX bridge it can be used with E1.31. A single-board E1.31-based controller is in the works, I believe. Aussiephil is the hardware guy, and Tabor handles software. Available now.

http://doityourselfchristmas.com/forums/showthread.php?12334-NEW-PropController-Project This one is a bit different. It is not specifically for RGB pixels, rather it is a multi-purpose motherboard to which you can attach daughterboards for specific functions. One of the planned daughterboards is for pixel string control. Available soon.

http://doityourselfchristmas.com/forums/showthread.php?12444-A-DMX-controller-for-multiple-RGB-pixel-strings
This one is my own pixel control project. It's a controller only, you supply pixels and power supply. Please see the forum thread for details. Available February 2011.

http://diylightanimation.com/index.php?topic=3554.0 This is a project by "RJ", the guru at that forum. His is a complete system that consists of a hub board where power is injected, and individual small controller boards that mount at the head of each string. Cat 5 cables connect a modified Lynx PC dongle to the hub, and the hub to the strings. It uses a standard PC-style power supply. AFAIK, it is only compatible with the 1804-style strings. Co-op buy going on now.

As far as commercial hardware from major vendors, Light-O-Rama is promising some new RGB pixel offerings in 2011, no details available yet AFAIK. Not sure about other vendors.

Lighting Software:
http://www.lightshowpro.com/ LightShow Pro is one of the first commercial software vendors to get heavily into RGB pixels. E1.31 supported now. Not free, but a steal at $80.

http://www.madrix.com/ The opposite end of the spectrum. Not sequencing software in the traditional sense, but a killer product for controlling a matrix of closely-spaced pixels. BIG $, but if you need it, the capabilities will stun you.

Pixels and other goodies:
www.aliexpress.com/fm-store/701799 *Ray Wu*

That's all for now. I intend to post a link to an expanded version of this document with some pictures included, but due to size limitations of attachments on most forums, I've omitted the pictures in this verison.

Questions, comments, corrections welcome: jim@sandevices.com

Jim St. John