

A Practical Guide to ~~Enslaving~~ Encouraging the Obedience of LED Light Fixtures, Via Computers

Danny Maland's informal and possibly irreverent manual for building a (**hopefully**) trouble-free computer-operated lighting system.

Maybe. Cross your fingers...



Table of Contents

Introduction.....	4
--------------------------	----------

The Hunt For The Right Gear.....	5
---	----------

Choosing DMX Control Software.....	5
---	----------

Choosing A DMX Interface.....	8
--------------------------------------	----------

Choosing Components For The Main Control Computer.....	10
---	-----------

<i>Cables.....</i>	<i>10</i>
--------------------	-----------

<i>Optical Drive.....</i>	<i>10</i>
---------------------------	-----------

<i>Card Reader.....</i>	<i>11</i>
-------------------------	-----------

<i>Case.....</i>	<i>11</i>
------------------	-----------

<i>Main Processor.....</i>	<i>11</i>
----------------------------	-----------

<i>Hard Drives.....</i>	<i>12</i>
-------------------------	-----------

<i>Keyboards and Mice.....</i>	<i>12</i>
--------------------------------	-----------

<i>Motherboard.....</i>	<i>13</i>
-------------------------	-----------

<i>RAM.....</i>	<i>13</i>
-----------------	-----------

<i>Monitor.....</i>	<i>14</i>
---------------------	-----------

<i>UPS.....</i>	<i>14</i>
-----------------	-----------

<i>Video Card.....</i>	<i>14</i>
------------------------	-----------

<i>Operating System.....</i>	<i>14</i>
------------------------------	-----------

Choosing A Remote Control Device.....	15
--	-----------

Choosing Network Hardware and Software.....	16
--	-----------

.....	17
--------------	-----------

Choosing DMX Lighting Fixtures (and Accessories Too!).....	18
---	-----------

Putting The Components Together.....	23
---	-----------

Building Your Main Control Computer.....	23
---	-----------

<i>Getting Ready For The Build.....</i>	<i>23</i>
---	-----------

<i>The Build Itself.....</i>	<i>25</i>
------------------------------	-----------

Configuration and Basic Testing of The Main Control Computer.....	42
--	-----------

<i>BIOS, RAID, and Operating System Tasks.....</i>	<i>42</i>
--	-----------

<i>Settings.....</i>	<i>42</i>
----------------------	-----------

<i>Updates, Drivers, and General Utilities.....</i>	<i>44</i>
---	-----------

<i>The DMX Interface.....</i>	<i>45</i>
<i>Lighting Control Software.....</i>	<i>45</i>
<i>Remote Control Software.....</i>	<i>45</i>
Setting Up The Network Hardware.....	47
Setting Up the Remote Control Device.....	48
Setting Up and Testing Your Lighting Instruments and DMX Lines.	50
The Full System Test.....	56
System Reference Diagrams.....	58
Main Control Computer Internals.....	58
Overall Control System Data/ Signal Paths.....	59
Troubleshooting.....	60

Introduction

If you're like me, you're probably getting into this because of necessity. You may have a dirt-simple lighting system that you use regularly, but isn't living up to your expectations. You may not have any lighting system at all, and you're trying to figure out what to implement. You may be looking for something bigger and better, so as to impress your clientele. Maybe you're just bored, and are desperately in need of something to do – hey, that's as valid a reason as any!

Have I got a deal for you.

We're going to go on a little journey together. This journey will take us to a mystical world where electronic signals flit through cable and ether, causing colored lights to bend to our whims. (Maybe – as long as the signals are actually intact upon arrival. I make no guarantees.) A world where our centers of control and power are safe from invaders. (As long as the invaders aren't really determined to break in, that is.) A world where dreams come true. (Assuming that your dreams are about simple lighting rigs controlled by computers. Anything else is beyond the scope of this manual.)

Maybe you're asking yourself, “Is there a serious side to any of this?”

The answer is an emphatic, “Yes!”

The bottom line is that this manual is meant to serve as a readable, informative guide to getting a flexible, sane, and problem-resistant lighting control system up and running. The end result should put you in a place where you have a system that is both usable in the now, and able to handle expanding needs in the future. There's no way for this manual to cover everything – indeed, success in these kinds of endeavors (actually, probably all endeavors) rests heavily on a person's own curiosity and ability to “figure it out.” However, this manual should at least give you enough guidance that you can get to that point of “figuring it out.”

There are some prerequisites for reading this manual to be useful to you. First, you should have some familiarity with lighting for the theater, or the world of music production. (If you know what lighting cues are, can recognize a PAR can, and have some notion of how DMX-controlled lighting works, you should be okay.) You should be comfortable with the idea of being “hands on” with PC hardware, and have at least a basic understanding of what server software does. You should also have at least a little experience in getting a basic computer network connected and operating. While I will certainly try to walk you through each part of the process in detail, having a bit of knowledge about the above concepts will help you to understand what the walk-throughs really mean **for you**.

Also, please be aware that I do **not** guarantee the safety, profitability, or anything else that you find in this document. Of course, I am going to show you what works for me, but this kind of enterprise is something that you try **at your own risk**. Just so you know.

Let's dive in.

(Splash)

The Hunt For The Right Gear

Choosing DMX Control Software

Whatever you do, don't rush this bit.

Now, why would I say that? Simple. The software you choose is at least 50% of the foundation of the entire system that you're going to construct. Just picking something at random is, therefore, squarely in the “bad ideas” filing cabinet. The software you choose will determine what operating system you'll need for the control computer, what DMX interface you can use, and a whole range of things that you can or cannot do easily (without picking different software).

Don't get me wrong – you can always pick different software, install it, and go in a different direction. There's always room to screw up, and I believe that some of our best learning comes from failure. However, taking the time to think through your software selection will have real benefits, even if the selection ultimately turns out to be a bad one. If you have a good idea of what your needs and wants are, then you are far more likely to identify a piece of software as falling short before too much time elapses. You will also have a much better foundation in place for choosing the right software on the second go-around.

So...before we crank up our web-browsers and go looking for slick screenshots and ten-screen-high feature tables, let's take a look at some considerations:

- Do you have a specific DMX interface in mind already? If you do, you will want to be **very** sure that your potential software choice can handle it. If you have any doubts at all, contact the software developer and make sure. If you can't get a positive, un-equivocated claim of the software supporting your hardware, **run away**.
 - Hand-in-hand with the above, is the software bundled with, and meant for only one specific interface? Such a thing isn't as evil as some people make it out to be, but be aware that you may be locking yourself into a narrow range of overall options.
 - Also, be aware that your software may be perfectly happy running on your OS of choice, but hardware drivers may be a different matter entirely.
- How many channels does the software support? Unless you are focusing on simplicity as a design consideration, I highly recommend that you accept nothing less than support for one full DMX universe (512 channels).
- Does the software have an intuitive way of handling cues or presets? It doesn't matter if the software can store and recall a huge number of cues if finding and recalling them is difficult. (Get a demo in your hands and try it out. If you can't get a demo, **run away**.)
- Does the software look and feel polished? A piece of software may claim to be able to do all sorts of wonderful things, but all of that may be for naught if the software has major bugs, or a clunky user interface. (Again, get a demo.)
- How much does the software cost? You know your budget, of course, but be careful of expensive software (especially when bundled with hardware) that is finicky, restrictive in what

it supports, and may represent a large monetary loss if it does not perform up to expectations. (Another reason to get a demo.)

- Does the software have a good user base? Do some searching on the web to see if you can find forums where the software is mentioned. A piece of software that nobody is talking about, and nobody seems to be using might not be the right choice for you. However, be aware that very cool things don't always have a huge following when they first appear. If something looks amazing, yet doesn't have a lot of “buzz” around it, it may just be new.
- What is the software's control metaphor? (That is, how is the interface presented?) Some simple DMX software implementations focus solely on simulating the layout of a small DMX hardware console – which isn't necessarily bad! However, simplicity often comes at the cost of flexibility. It's not that you can't control multi-channel fixtures with a simple console layout, it's just that the control metaphor isn't suited to doing it. This is **very** important to think about if you want to make extensive use of DMX devices that go beyond simple dimmers and relays.

Multi-channel fixturewhatsits? What does that mean?

A multi-channel fixture is a DMX luminaire (a complete lighting instrument) that allows for discrete control of several features in one unit. An example of this can be found in the increasingly-common LED color changer/ LED PAR. In order to mix colors, you need to control the intensity of red, green, and blue LED lamps, each color independent of the other. In order to have that independence, each color needs its own control channel. That means at **least** three channels are needed, because some LED color changers implement features beyond just intensity controls for each color of LED.

Bottom line? The number of channels you can control is directly proportional to the number of fixtures you can support. That's why having an excess of channels available is helpful. Multi-channel fixtures can eat up your available channels in a hurry.

For the purposes of this manual's walk-through, I have chosen the software called “Freestyler.” In terms of the above questions, Freestyler looks like this:

- Do you have a specific DMX interface in mind already?
 - Yes, the planned interface for the walk-through is the Enttec DMX USB Pro. Freestyler not only supports this interface, it practically endorses it via a direct website link.
- How many channels does the software support?
 - Freestyler supports up to 512 DMX channels.
- Does the software have an intuitive way of handling cues or presets?
 - This one was a little bit tricky, as I had to engage in a bit of trial-and-error (and digging through the help file) to figure out why my test cues weren't being recalled. I was mostly on the right track, and just needed a bit of a nudge in the right direction.

- Does the software look and feel polished?
 - Yes. The software looks like it belongs to the “Windows Vista Generation” (love it or hate it), has a help file that is accessible from within the program itself, and seems to be easy to read in general. (The tool-tips that appear when hovering the cursor over the various tool-bar buttons are very handy.)
- How much does the software cost?
 - Not a penny. (It's hard to compete with “free.”)
- Does the software have a good user base?
 - I was introduced to Freestyler via a lighting forum that I occasionally visit, and it seemed to get a number of positive mentions. (With the propensity of forum-goers to moan and groan about everything, positive results are something to take note of.)
- What is the software's control metaphor?
 - Freestyler employs what I would call an “object-oriented” point and click interface to manage fixture control, a “media player-esque” window for building scenes (probably what most people would call chases), a table for cues (each cue being a scene), and a set of buttons for override control (interrupting cue playback with specified scenes). As a special note, I will point out that choosing colors for an LED color changer is as easy as using a Windows-style, we've-all-seen-this-a-million-times color picker.

There are some possible substitutions for Freestyler, of course. They are:

- JS DMX, which appears to be a fairly simple piece of control software, built around a “small console” control metaphor – but with some extras.
- Q Light Controller, which appears to have some similarities to Freestyler, in terms of control capabilities and the way the interface works.
- Software from Chromakinetics, which is actually a number of different, seemingly lightweight utilities. One of these is a small web server that allows users to control lights via simple commands passed through an HTML form.
- MagicQ, a system that appears to hybridize a “console” interface metaphor with features similar to Freestyler or Q Light Controller. The manufacturer claims support for a large number of DMX universes (18), but it appears that DMX over Ethernet (or Ethernet controlled DMX interfaces) are required for a multi-universe setup.

All of these software packages appear to be free of charge for end users.

Choosing A DMX Interface

If you've already chosen your software, then you are probably in one of these two situations:

A) You need to pick a DMX interface that plays nicely with your software. (You may have had your eye on an interface already, or not.)

B) Your software is bundled with an interface.

If you're in situation “B,” then the manufacturer has done all the compatibility work for you. (Hopefully, anyway. You never know with some people...) However, you should check that the bundled system really is compatible with the operating system you plan to use for the control computer. If you have to rethink your system design, **now** is the time to do it. (Reworking a system design after buying actual parts is a serious headache. I know – I've done it.)

If you're in situation “A,” then there's a bit more homework to be done. You should consider the following:

- Is the interface **really** compatible with the chosen software package? (**Yes**, you should check again. Really. We'll wait. **Don't rush!**)
- Does the interface have drivers available for the operating system I want to use? This can be a huge issue, if you don't watch out. If you want to use, say, Windows 7 64-bit, you need to be **very** sure that the manufacturer has taken the trouble to get their drivers updated. (It's easy to assume that the manufacturers will keep abreast of operating system releases, but it's not a safe assumption.)
- Does the interface rely on the computer to generate DMX, or does it have its own microprocessor to do so? It is possible to get rather inexpensive DMX interfaces that rely on the host computer to do all the heavy lifting, but be aware that this leaves you with no safety net if the computer becomes loaded down with something else. (Yes – modern computers are really fast. However, they also have a lot more to do, and can sometimes get “hung up” on what seems like a simple task. If you end up relying on the computer to generate DMX, try not to use it for anything other than the light show.)
- Does the interface actually have the connectivity that I need? Some interfaces only have DMX output – which is fine, if that's all you want, On the other side, some interfaces can support multiple DMX ports and advanced features, like DMX over Ethernet (Art-Net), and RDM (Remote Device Management). Some interfaces use 3-pin connectors, and some use 5-pin layouts (which can be adapted to each other, but it **is** more equipment to buy). My advice is to buy something that addresses your needs into the predictable future, and nothing more. That is, don't spend money on bells and whistles that you don't have a clear intention of using in the future.
- How much does the interface cost? Better interfaces usually cost more, but beware of sinking your entire interface budget on a “gold plated” device. On the flip side, beware of being lured into buying a cheap interface, just because it is cheap. Reliability is a big deal for a piece of hardware that will be driving a light show “for the public,” so a few extra dollars for a well-supported and robust device are likely to be well-spent.

For the walk-through, I will be using the Enttec DMX USB Pro. The reasons for this choice are as follows:

- Is the interface **really** compatible with the chosen software package?
 - Indeed, Enttec even lists Freestyler as recommended software for the device.
- Does the interface have drivers available for the operating system I want to use?
 - We'll probably end up using Windows 7 64-bit for the walk-through, and Windows 7 is listed as supported on the DMX USB Pro's driver page. (I also downloaded the driver package to my Vista PC, and tried an install. The installer appears to work correctly, and the behavior of the "OS detection" routine implies that 64-bit systems are supported.)
- Does the interface rely on the computer to generate DMX, or does it have its own microprocessor to do so?
 - The USB Pro has its own microprocessor for generating DMX, which I see as a worthwhile expense for a bit more reliability.
- Does the interface actually have the connectivity that I need?
 - The connectivity for this walk-through will be fairly simple. All we really need is DMX output, as our use of Ethernet will be to transfer data between computers (as opposed to DMX devices). The USB Pro does make use of 5-pin connectors for DMX signals, so adapters will almost certainly be a necessity.
- How much does the interface cost?
 - The USB Pro is about \$151 before shipping. It's not as inexpensive as Enttec's "bare bones" offering, but again, it has its own microprocessor for DMX generation, and also appears to be more tolerant of the way the latest generations of Windows handle COM ports.

Possible substitutions for the DMX USB Pro are:

- The Enttec Open DMX USB, which is an inexpensive alternative to the USB Pro. It relies on the host computer's processor to generate DMX, and is a bit more finicky in terms of what COM port drivers it will agree with. It may or may not work with the latest generation of operating systems.
- The Velleman VM116, which appears to be somewhere in-between the Open DMX USB and the DMX USB Pro. It is apparently a bit more robust than the Open DMX, but without having the full feature set of the DMX USB Pro.

Choosing Components For The Main Control Computer

The plan for this walk-through is to build the main control computer from scratch. Why? Because we're going to want to install Windows on a RAID 1 drive system.

A **what?**

RAID means “Redundant Array of Independent Disks,” and the “1” means that the drives are copies (mirrors) of each other. The idea is that, if a drive fails midstream, the computer will be able to operate until a replacement can be installed. The reason for doing this is that it would **really** be a bad time if your lighting control system died in the middle of a show, with no way to bring it back. It's true that many different things could potentially cause a catastrophic failure, but using a RAID is relatively easy these days, and takes one potential problem out of the equation. (Or, at least, mitigates that problem.)

...okay, but why does that necessitate building a machine from scratch?

Mostly, the problem comes from many manufacturers not offering the option of having Windows installed on a RAID from the get go. When you couple this with not knowing whether or not you'll actually get real Windows installation media (like a CD or DVD), the complications involved with tearing into a pre-built PC and re-installing Windows are a bit painful. My view is, “Why not just sidestep all that?”

A quick note on shopping: If you shop with a site that has ratings, make sure that an overall rating is backed up by being averaged over a good number of individual ratings. In statistics, this is called “having a sufficient sample size.” A high (or low) rating doesn't mean much if only one or two people's opinions are involved. Just as a guess (how's **that** for statistical rigor for you?), I would say that 50 reviews are the minimum needed to **really** get a feel for how an item rates.

Maybe.

Your mileage may vary.

So – on with the show.

Cables

It's good to have some spares around, especially if you don't know exactly what your internal drives will ship with (if anything at all).

Since we're going to be using SATA (Serial ATA) drives for this machine, what you need to look for in cables pretty much comes down to whether they're SATA or not, and are long enough. Eighteen inches is probably enough for most applications.

You'll probably want some extra “internal” power cables as well.

Optical Drive

In one sense, the optical drive is a critical component, as the operating system will be loaded from it. In another sense, though, the optical drive is mostly going to be around as a convenience, or “just in case” item. Spending a lot of money on features probably isn't necessary, but you should buy an optical drive that can burn all the sub-formats of the media type that is currently common.

Card Reader

This is another non-critical item, but one that should probably be included. Why? Because sometime, for some reason, you're going to want to be able to read a memory card of some kind. Trust me. Look for one that reads as many formats as possible.

Case

The case that you put the computer components inside is important, but unless you have very specific form factor needs (like, say, rack-mounting), probably not something to be extravagant with. I will say that little extras and touches can be **very** helpful during the build of the main computer (removable everything, case panels that just open by lifting a latch, and the like), so they can be worth paying for.

Things to look for in particular:

- What is the primary material that the case is constructed from? Plastic and aluminum cases can be light and cheap, but steel is usually a bit stronger.
- Does the case include a power supply? This can save you the trouble of buying your own, but beware of cheap power supply units. They may not hold up for very long. If you do buy a case bundled with a power supply, you probably should accept nothing less than a 500 watt supply. (This should be enough breathing room for this machine, as it won't have an enormous number of components – but if you're planning on building something more extensive, you may want more power available.)
- Is the case large enough? Get yourself some breathing room. Unless the case absolutely must be small, you might as well get something that will fit a normal (as opposed to mini) motherboard.
- Does the case include fans? The nice thing about getting a case with included fans is that you don't have to fiddle with buying your own. Larger fans may be quieter than smaller fans, but not always. Also, if you want to build a really quiet system, you may still want to buy your own fans.
- Does the case have potentially breakable parts, like fancy drive doors and other cosmetics? This does come down to a matter of taste in some respects, but it can be frustrating when a flimsy part breaks (or when a piece of decor becomes nothing more than a nuisance). For instance, it is **really** annoying when an optical drive cover's button-extension becomes misaligned, and you can no longer get the drive's manual open/ close button to work.

Main Processor

Boy howdy – is this going to be fun.

The choice of a CPU is the ultimate outcome of weighing an entire busload of factors, from cost, to performance, to religious convictions about which manufacturer is less evil, to...you name it. It's tempting for me to just say, "buy as much processor as you can afford," and leave things there. However, there are some meaningful things to think about in terms of what really constitutes how much processor you're getting for your money.

- How much future proofing do you need? More expensive processors will tend to give you performance that feels adequate for a longer period of time. However, at some point, you begin to pay a large premium for that extra performance. Also, in an application like ours, brute force really isn't the factor that it might be in, say, 3D rendering.
- Will this computer be used for something other than lighting control? If there is a real chance that the computer will be pressed into doing something beyond just lighting control, then more horsepower is probably desirable. I should, however, take the time to say that doing things **other** than lighting control on the control computer during a show is to be filed in the "Bad Things" category. (There are legitimate needs, of course – like music playback in a pinch. Still, the fewer things you ask the lighting controller to do, the fewer things that can go wrong.)
- Are any of my processor choices "crippled?" Some processors may be made more cheaply by leaving out cache memory, or using less cache. A processor's cache allows for storage of instructions and data in a memory space that is much faster to access than system RAM. Lighting control is unlikely to become highly memory intensive, but it is certainly worth noting that CPU's with more available cache may show better overall performance.

Hard Drives

For our application, the only real watchword for the hard drives is "middle-of-the-road."

Let me explain why.

This machine is not intended to be a "hardcore, uber-A/V workstation of doom." It's possible to spend large amounts of cash on capacity and performance, and to have neither of those factors be worth the cost in the end. I am of the opinion that, these days, really high drive performance is only needed for the most demanding A/V applications anyway – and that isn't what we need to do.

For this reason, I don't see a need to have any bullet points about what to look for. Get a couple of drives with a favorable performance record, and about as much capacity as you would want for a "home" computer.

Notice that I said "a couple" of drives. We're going to use these for the RAID 1.

Keyboards and Mice

If there's anything in this whole walk-through that is entirely a matter of taste, I think this might be it. Buy a keyboard and a mouse that you like, and can use easily. A Microsoft Curve Value Pack might be all you need. (Because I'm really cheap, I'm just going to rummage around for a spare keyboard and mouse in my pile-o-spare-parts.) You might want to consider going with a trackball instead of a traditional mouse, to save space.

Motherboard

Now, in contrast to the keyboard and mouse selection, this piece of gear has to be chosen carefully. Here are the particulars:

- Does it explicitly support your processor choice? If it doesn't you may or may not get away with using the board. Look for one that really is meant to support your processor series (**not just the socket**).
- Does it support enough RAM? As Windows becomes more and more hungry for memory, you will want some breathing room. As of this writing, Windows 7 64-bit claims to need 2 GB of RAM. In my opinion, you should see the OS RAM requirement as being only **half** of what you should have installed.
- Does it support the latest RAM standard? I don't think that it's necessarily worth paying a large premium for the very fastest RAM available, but I do think that it's worth using the most up to date overall RAM standard. This should help you with both future-proofing the machine a bit, and also with getting good performance out of your CPU.
- Does it have the latest standard expansion slots, and enough of them? The plan for this machine is by no means meant to fill the motherboard's slots to capacity, but you might as well have some breathing room.
- Does it have RAID support built in? This is very important for us, as using a RAID is important for the reliability of the machine during a show. The support might as well be right on the motherboard, so we don't have to add another piece of hardware to make it work.
- Does it have on-board video? Depending on your needs and desires, you might want to skip the step of getting a discrete video card. My preference is to always go with discrete graphics, but having your video on-board can simplify your life a touch.

RAM

When choosing RAM for your system, I would look at the following features (beyond just buying twice the capacity needed for your operating system):

- Is the speed explicitly supported by your motherboard? Some motherboards may only support certain RAM speeds by using overclocking of some kind. In a system like this, I would argue that reliability is a more important goal than brute force. In my mind, native support (that is, being designed to use RAM within normal operating parameters) is key.
- Is the speed explicitly supported by your processor? Similarly to the above, you might as well find RAM that mates well with your processor. Buying RAM that is outside of your processor's specifications may result in having put money into something that you can't use to its full potential – if you can use it at all.
- Does the RAM use “normal” voltage? Some high-performance RAM kits require you to increase the voltage to the RAM modules. To me, this is an unnecessary complication for a machine of this nature. Use “plain vanilla” RAM, and spend less time configuring things in the BIOS (Basic Input Output System).

- Does the RAM come as two modules, so that you can implement dual-channel RAM? As much as we are not chasing “top performance at all costs,” having two identical RAM modules allows for the better performance offered by dual channel RAM – a potential performance enhancement that is essentially “free.”

Monitor

This is another highly subjective piece of the puzzle. You're going to want something that's a comfortable size, and supports a resolution that you find to be easy on your eyes. I don't recommend putting a large amount of money into the monitor, as we're not making critical color decisions based on its output. (Yes, we will be previewing color choices for the lights, but that's really not a high accuracy application. Well, unless you have a really picky lighting designer coming in.)

Also, depending on your needs (and what your video card will support), having two monitors available can be really cool.

UPS

The reason for having an un-interruptible power supply for the main control computer is one of reliability. Even though a power outage may shut down the actual lighting instruments, keeping the lighting controller operating allows you to perform a graceful shutdown (and hopefully save the state of the light show). If the power comes back on quickly, there is the added bonus of the control computer still being up and ready to go. (As an aside, with an LED lighting system, you might not find it too expensive to even keep part of the light show on UPS systems – which might be a good thing for safety, preparedness, and just plain being a hero.)

Video Card

This is another somewhat subjective element. For this walk-through, I really have no interest in 3D visualizations of the light show – but that doesn't mean that you're the same way. Additionally, with Windows driving the actual window manager via accelerated graphics, you're probably going to need something that's good enough for the basics. Practically speaking, any card that works with your motherboard will probably do enough to be usable.

Operating System

If you've been following this walk-through from the beginning, this choice has probably been made by now. If not, it all comes down to compatibility. Whatever OS you choose has to be compatible with all your software and hardware. Another consideration is whether or not it will install easily on a RAID. Windows XP required a floppy drive in order to load RAID drivers, but Windows Vista (and beyond) should let you use more modern methods. Using a “late model” OS will get you some future proofing, but pushing the envelope on a newly released OS may also get you compatibility problems.

Choosing A Remote Control Device

In terms of this walk-through, the whole point of the remote control is for taking the reigns of the lighting rig while you're actually working on the lighting instruments.

Why?

Because it is a **royal** pain to be working on a lighting setup, and to be required to run back and forth from, say, your ladder to the lighting controller. How much better is it to be able to check the operation of a DMX fixture while you actually have that fixture close at hand?

Answer: Lots.

When choosing a remote device, you should probably ask these three questions:

- Does it have a screen that's big enough for what I need to do, but not so big that it's unwieldy? On the one hand, the idea of running a lighting controller from something like a smartphone has a certain appeal. On the other hand, navigating the high (or high-ish) resolution interface from the main controller may be difficult on a really small device. On a different hand (or maybe a foot), an enormous, glossy screen looks great on a desk, but may be just another heavy, unwieldy part of a heavy, unwieldy device – a device that's almost as hard to lug around as the main controller.
- Can it easily run software for a remote connection to the main controller? If you can make your remote control solution work over something like a plain-old-vanilla web browser, this may not be a very difficult thing for the remote to do. Other solutions, however, may require the remote to have full capabilities for a “heavy” operating system (Windows, Mac OS, Linux, FreeBSD), and the associated software.
- Is wireless networking built-in, or easy to add? The whole point of the remote is to not require a cable run. If getting wireless networking to operate on the remote is difficult (or impossible), then the remote may be a lot less “remote” than you might like. (Then again, wired networking is simple and bomb-proof, so maybe you won't mind running a long cable out to your remote. It's up to you.)

Choosing Network Hardware and Software

In order to have the remote control actually be, er...”remote,” you will need to have some sort of data network in place. In order to have the flexibility called for in this walk-through, the data network will have to be wireless. (However, I should point out once more that you may be perfectly happy with running a long cable to where you want to be.)

Now, another question is what will be transmitted over the network. Some of you might opt for a DMX-over-Ethernet solution, but this walk-through will be using an interface-presented-via-network solution (which you might call remote desktop, VNC/ Virtual Network Computing, or...whatever fits, really).

On the hardware side, this is what to look for:

- Does the wireless access point have a built in firewall? In general, your show-control network should not be connected to any other network at all. However, you are going to make an exception at some point (don't tell me you're not going to, because I know you're going to. How do I know? Because I've done exactly that.) When that exception is made, you want something standing between you and everything “in the great beyond.” In fact, I recommend just buying a router with basic firewall features, so that you can have your own private network for the lighting system.
- Does **all** your wireless hardware (both the access point and the remote's wireless networking adapter) implement WPA2 security with AES (Advanced Encryption Standard) encryption (or better, as time marches on)? WEP is **not** strong enough for effective security on a network that is used for a public performance, and TKIP (Temporal Key Integrity Protocol) can also be attacked quite effectively.
- Does all your wireless hardware support the fastest wireless protocol available? You will want as much network bandwidth available to you as possible, even if only as an “insurance policy.” You want the remote to be as responsive as possible, even if the remote won't be used to run an actual show. (Then again, you never know...)
- Does all your wireless hardware implement Wireless Protected Setup? This can save you some headaches, although there are some security issues that can arise. Still, it's nice to have in a pinch.

The software side can actually be a bit more tricky. Sometimes, support for one “neato” feature may displace another feature that is less flashy, but more fundamental. (For instance, you might be able to use a Java-enabled browser for VNC sessions, but not have encryption available for the VNC connection. Really!)

When looking for VNC software, you probably want to take these items into account:

- Does it support encryption? If not, you will need to implement some sort of VPN (Virtual Private Network) to give yourself that extra layer of security. If an attacker actually does gain access to your network, you want to make the job of gaining access to your main control computer to be as difficult as possible.

- Does it allow for connections via a Java-enabled browser? This is that “neato” feature I described above. However, it does have a practical side. If you get into a jam, and need to connect from a remote device without a VNC client, this can be a lifesaver. (Then again, it's also a security risk, because it lowers the difficulty of connecting to the VNC server. Decisions, decisions...)
- Are clients available for a variety of operating systems? This is also handy if you get into a jam, and don't have your normal remote device available. Like the Java client support mentioned above, it also raises your “attack surface area.” (This is a fancy, techie buzzphrase for saying that there are more options for folks who want to get in without permission. It's like having a lot of gates around your pasture – it's convenient, but there are more chances for varmints to get in.)

Choosing DMX Lighting Fixtures (and Accessories Too!)

Strap yourself in – we're going to do a bit of science and math. (Don't worry, I'll let you use a calculator. I'm not like your 5th grade teacher.) Some figuring will have to be done in order to predict the performance of lighting fixtures in your application, and that figuring involves logarithms. Logarithms are a very handy mathematical tool for dealing with problems that involve proportionality, and large ranges of magnitude.

Before we dive into the logarithm-based math, let me do a little groundwork on some basics. (You know how you always used to whine in your math classes, “How are we gonna use this?” I'm going to actually answer that question for you.)

Consider the fact that, say, during a blackout, you can get enough light from a single candle to make out the objects in a room. (Not in any detail – I'm talking about the general shapes, here.) Now, also consider the fact that you can also function in very bright sunlight. That's quite a range of intensities. Also, consider that, if all you have is a single candle, adding another candle appears to make a noticeable difference in the available light. Now, what if you're in bright sunlight, and light a candle? The candle adds nothing significant to the light radiating from the sun. That's proportionality in action for ya.

Proportionality is also very useful for dealing with the intensity of light as a function of distance. Light intensity obeys the inverse-square-law, which means that each time you double your distance from a light source, the apparent strength of the source is quartered. (The intensity is inversely proportional to the square of the distance.)

Here's what the above looks like in Math-ese:

$$\text{Apparent Intensity} = \frac{\text{Intensity at an Arbitrary Distance}}{\text{Arbitrary Distance}^2}$$

So, if we put the number “1” in for all the numbers, we get:

$$\text{Apparent Intensity} = \frac{1 \text{ Intensity Unit}}{1 \text{ Distance Unit}^2}$$

Which easily works out to an apparent intensity of 1. (I'm leaving out the “real” units to make things easier to read.)

Now, if we double the distance, things look like this:

$$\text{Apparent Intensity} = \frac{1 \text{ Intensity Unit}}{2 \text{ Distance Units}^2}$$

Which is 1 divided by 4, or ¼, or 0.25. That's inverse-square in action.

For a real-world demonstration, find yourself a lamp, and hold the back of your hand fairly close to it. Notice how bright your hand is, with lots of highlights. Now, hold your hand about twice the distance away. You should see noticeably less brightness.

Okay – so what?

The issue is that you need to pick fixtures that have enough brightness to be usable at the distances involved in your productions. A wall wash/ accent/ uplight/ light-that's-close-to-what-it-hits doesn't need as much intensity as lighting that's 20 feet (or more) from human performers. (Or dog performers. Or turtles. Or giant robots...) If you're working in a small room, you **might** be able to get away with light fixtures that have less output than more costly devices. It all comes down to your expectations.

So – how do you know what your expectations are?

If you're after a stage that is fairly bright in a small-venue context, you will probably want to be able to achieve between 500 and 1000 lux at whatever you're trying to light. (Lux is the unit for lumens per square meter. Most of the LED fixture specifications that I run across are in lux units.) I would personally see 50 lux as the very lowest limit for any usable lighting – and I may be being a bit too permissive in that regard.

Now – a word of warning before we jump into doing math again. Manufacturers love to give you ratings that look good on paper, so you need to read between the lines. If you're buying LED color changers, it is my opinion that the **only** safe assumption is that the light's output was measured with all the LED's emitting at maximum intensity. You should expect a **dramatic** drop in output if you are planning on using lots of saturated colors, especially red, green, or blue (because those are the “native” LED colors, so no mixing is involved). Also, you need to be sure that the manufacturer provides a distance where the measurement was taken. A fixture might have a big number published, but where was that measurement taken? At 1 foot? At 1 inch? At 1 Planck-length? (Most manufacturers seem to be content to measure at 1 meter, but if the distance isn't mentioned, **don't assume anything.**)

Okay. Back to the math.

You thought that I forgot about those logarithms didn't you? Nope. We're almost there.

One way to start is by knowing the theoretical output of a fixture you're interested in, and how far that fixture is going to be from what you're lighting. First, take a look at the units of measure that the manufacturer used for the distance in the fixture rating. If you're using feet, and they're using meters, you should convert your measurement to meters. (You can also go the other way – it's just important that the units match up.)

One foot is 0.3048 meters. Start your calculators!

Once all your units are converted, here's the operative question: Compared to the manufacturer's measurement distance, how many doublings does my target distance represent? Each doubling can be expressed as 2 raised to some power.

Here's what this looks like so far:

$$\text{Target Distance} = \text{Measurement Distance} * 2^x$$

Now for some algebra. We have to move things around to isolate the unknown power being

applied to 2. We divide both sides by the Measurement Distance, and get this:

$$\text{Ratio of Target versus Measurement Distance} = 2^x$$

See how we now have a proportionality between the target and measurement distances? Here's where logarithms become handy. They're a very convenient way to figure out what that x actually is in a situation like this. The way you do it is by grabbing that 2, and using it as the base for taking the logarithm of this proportion that we created. Here's how it looks:

$$x = \log_2 \text{Ratio of Target versus Measurement Distance}$$

If you're lucky, and have a calculator that can do logarithms with arbitrary bases, you've pretty much got your answer. If not, you'll have to engage in a bit of logarithm trickery, where you take the base 10 logarithm of the distance ratio, and divide that amount by the base 10 logarithm of the base you actually want. Like this:

$$\frac{\log_{10} \text{Ratio of Target versus Measurement Distance}}{\log_{10} 2}$$

(The base 10 logarithm of 2 is 0.30103, by the way.)

So – at this point you have a number which tells you how many doubles of the measurement distance your target distance is. You're still not quite done, though, because this number of doublings (which we'll call “the answer you just got”) has to be applied to the inverse square law, like this:

$$\text{Manufacturers Measured Intensity} * (1/4)^{(\text{The Answer You Just Got})}$$

That's 1/4 (or 0.25) being raised to the power of “The Answer You Just Got,” in case that expression looks confusing.

At this point, what you have is an intensity at your target distance, which you can check against your expectations for acceptability. This method is good if you have a little bit of “swing” in what you will accept from a fixture, and want to see just how much a proposed light deviates from what you need.

Here's an example, using a very inexpensive LED fixture. The manufacturer claims about 1200 lux at 1 meter. The distance from any particular light to a performer is probably about 15 feet, or 4.572 meters. Here are the steps:

$$4.572 \text{ meters} = 1 \text{ meter} * 2^x$$

$$4.572 = 2^x$$

$$x = \log_2 4.572$$

$$x = \frac{\log_{10} 4.572}{\log_{10} 2}$$

$$x = 2.19282540355$$

$$1200 \text{ lux} * (1/4)^{2.19282540355}$$

Which gets us an answer of 57.4 lux at the target distance. This light is just barely over the dimmest acceptability threshold that I would personally allow – and that's with all the LED's going at full tilt. (Now you can see why the fixture is as inexpensive as it is, being about \$45 US.)

If you don't mind some generality (or you just don't want to do any math), here's a table I've prepared which gives you an idea of how different fixtures can perform at different target distances. Please note that I'm rounding up to the nearest whole number of lux. (What you'll notice pretty quickly is that a bright stage at long distances probably means balancing between high-dollar single fixtures, and a larger number of less expensive fixtures.) Also, please realize that this table is one of those “frictionless vacuum” calculations – meaning that the numbers do not take into account any optics that might be applied to a fixture, reflections from light-colored objects on your stage, and other such considerations. Your mileage may vary.

Lux at 1 meter	Lux at 10 feet (3.048 meters)	Lux at 20 feet (6.096 meters)	Lux at 30 feet (9.144 meters)	Lux at 40 feet (12.192 meters)	Lux at 50 feet (15.24 meters)
1000	108	27	12	7	4
1500	161	40	18	10	6
2000	215	54	24	13	9
2500	269	67	30	17	11
3000	323	81	36	20	13
3500	377	94	42	24	15
4000	431	108	48	27	17
4500	484	121	54	30	19
5000	538	135	60	34	22
5500	592	148	66	37	24
6000	646	161	72	40	26
6500	700	175	78	44	28
7000	753	188	84	47	30
7500	807	202	90	50	32
8000	861	215	96	54	34
8500	915	229	102	57	37
9000	969	242	108	61	39
9500	1023	256	114	64	41
10000	1076	269	120	67	43

Now then. Other than this very long discussion of “does it give you enough light,” here are some other factors to consider:

- How is DMX addressing accomplished by the fixture? There's nothing wrong with DIP (Dual In-line Package) switches, but they're not as easy to operate as other methods.
- How many channels does the fixture use? Be aware that some fixtures may chew up DMX channels with features that you don't necessarily want.
- What connectors are used by the fixture? Three-pin DMX is a bit easier to get cable for, but be sure that you're ready to adapt any discrepancies between the fixtures and your DMX interface.
- How much power does the fixture draw? LED fixtures pull a lot less power than conventional lights, but you still have to be aware of how much power you need to run them. (This is especially true if you're going to put together a system with a lot of lights).
- How heavy are the fixtures? Like the power issue, this can “bite” you if you're not careful. LED fixtures tend to be more lightweight (in general), which may lead you to overload a mounting system unintentionally. Safety first!
- Do the fixtures have any special features? Special features, like master/ slave operation, built in programs, and other such additions can be both a help and a hindrance. If you want the extras, then you should definitely look for them, but also be aware that a mis-configured switch may cause the light to do something unexpected. You may need to take a little extra care when setting up the light.
- What accessories are needed? Depending on how you arrange your lights, you may need longer or shorter DMX cables. You should probably buy a DMX termination adapter, just so that you have it handy. This is also a good time to double check that you have the adapters necessary to make both the lights and the interface compatible with the cables you're going to get. You may also need to buy the mounting hardware (things like “C” and “O” clamps) separately from the light. If you're going to hang lights where they can fall on people or property, you will want to get a safety cable for each light.

Putting The Components Together

Building Your Main Control Computer

Getting Ready For The Build

Before you start the actual build of the control computer, there are some things you should scrounge up, if you can:

1. A multi-tip screwdriver, where the shaft of the screwdriver is significantly thinner than the handle. The shaft should at least be roughly $\frac{3}{4}$ as long (or longer) than the handle. What you want to avoid is one of those drivers with the big, fat handle and very short shaft, because they are often too big to be effectively maneuvered between computer components. An example of what you do want is below.



2. A pair of small-ish needle-nose pliers. These will come in handy as “fingers” when you have to deal with smaller components, or apply pressure on a “tab” or “latch.” (Seriously, doing it with your real fingers can be painful. Who comes up with these designs?) You may also need to break off metal parts from the case, which is another case where the pliers are helpful. Here's an example:



3. Some extra-long tweezers, which are very helpful for handling little things like jumpers, and jumper-style connectors. Here's what I'm talking about:



4. One of those magnetic “picker-upper” thingamabobs, with a telescoping shaft. These are life-saving when you drop something small inside a computer case (and you **are** going to drop something), or if you need to maneuver a screw into a tight corner.



5. It's also nice to have some of those “twisty-ties” close by for doing cable-management. You may even be able to find some that were used for packaging the computer components you bought – love the Earth...Recycle!

6. A fine-tipped permanent marker can make it easier to identify things like mounting holes, if you don't feel like playing a game of “memorize the positions of things” all day long.

That's pretty much it.

Oh – wait!

An anti-static wrist strap. Yes, I should recommend one of those to you, because zapping a component and killing it would really make your day go downhill. Except...you're not going to use one. How do I know? Because I know that you (like me) have tried to work on something before, while wearing an anti-static strap. I also know that you (like me) discovered very quickly that the blasted thing's cord was constantly getting hung up on what you were working on, impeding your movements and generally making it impossible to get anything done.

The moral? Be careful. Touch the bare metal of the computer case often, and don't wear clothing that generates a lot of static. Handle things by the edges. Don't touch the circuit paths. Did I already say to be careful? Be careful.

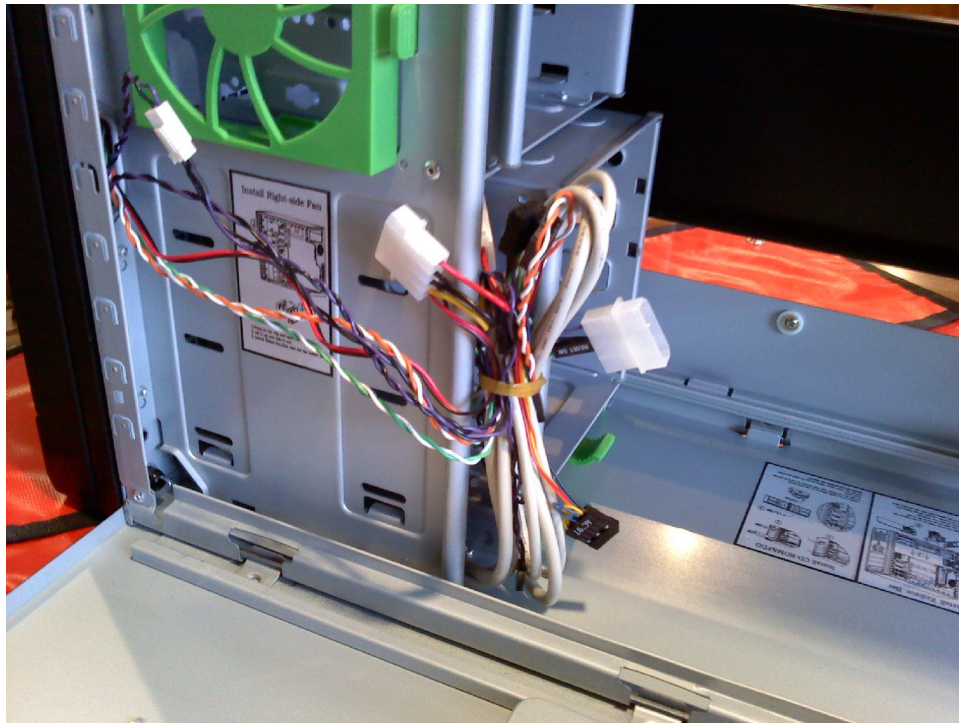
Now, if you can get ahold of an anti-static mat of some kind, that's at least something.

At this point, I'd like to share some general principles of “building a computer without getting to the point of desiring someone's death.”

- Take your time. This isn't a race, and good overall craftsmanship will leave you with a finished control machine that is more reliable, and more easily maintained than one built in slipshod fashion.
- Try to work from the ground up. That is, try to put things together in such a way that parts which will block access to other parts go in last. (You may find it easier to think of this in terms of working progressively nearer to yourself.) This sometimes means removing things pre-installed in the case, like fans. This is another reason not to rush – take some time and **think** about how all the bits and pieces are going to be layered in there.
 - If the power supply isn't pre-installed in the case, put it in first. It's really irritating to put a bunch of components in the case, and then go to add the power supply – and realize that you're going to have to undo a large amount of work to get that sucker in there.
 - Get the motherboard in the case, and then immediately work on things like jumpers, and pin-header connections. It is orders-of-magnitude harder to get those dangly things in the right places when you have to work around something else. After that, install the processor and connect the power supply to the appropriate point(s) on the motherboard. **Now** you're ready to start adding other components.
- Test as you go. If adding a component causes the machine to do a “no-POST, no-error, no-beeping, no-nothin',” then you want to know immediately after you add that component. Backtracking to find a problem can be downright **painful**. (POST is “Power On Self Test,” and it's very disconcerting when you get no POST at all – you just sit there with a blank screen, and no clue as to what's gone wrong.)
- Take breaks when necessary. (Seriously!)

The Build Itself

First, get the case unpacked. You should remove most of the packing materials (especially things that will get in your way during the build), but there are some things you should leave intact. Cables, for instance, **will** get in your way during the build if you don't tuck them away somewhere. If the manufacturer has been nice enough to tie the cables down for shipping, you should leave them as they are for now.



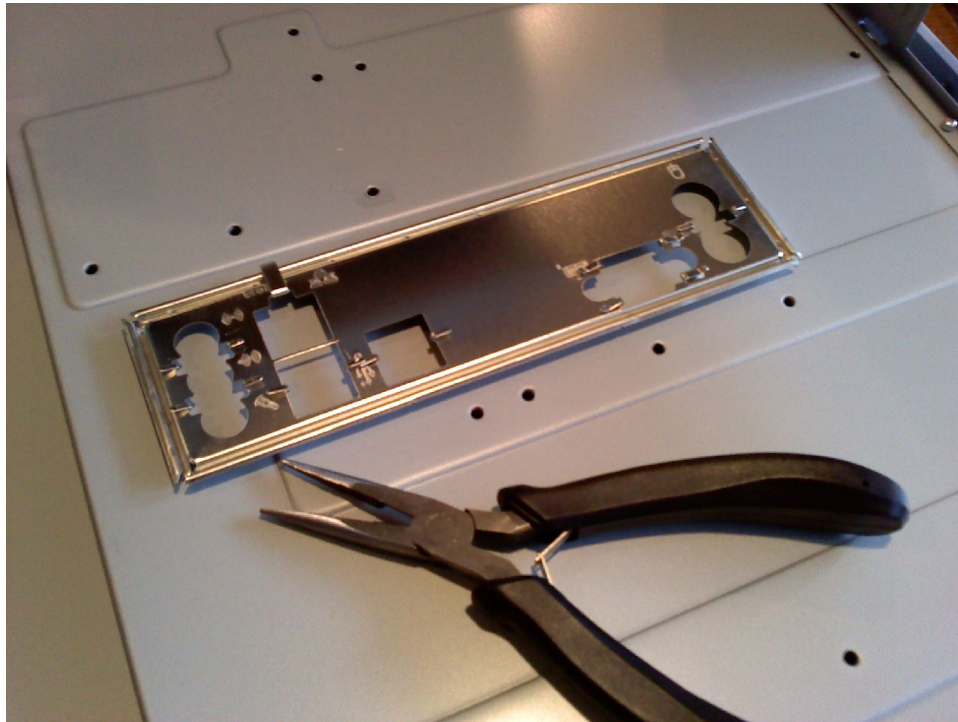
If necessary, you should now install the power supply, and then do any needed “path clearing” so that the motherboard can be installed with as little fuss as possible. In the case of the...er...case that I chose for this walk-through, there's a bit of extra luck. The motherboard rests on a tray which easily folds out of the main case body. (This is one of those “neato” extras that I talked about earlier.)



You may notice in the picture above that a number of “stand-offs” have been pre-installed into the case. In a situation like this, where the mounting holes are all at a uniform height from the motherboard tray, we will require standoffs for as many of the mounting holes as possible. (Some

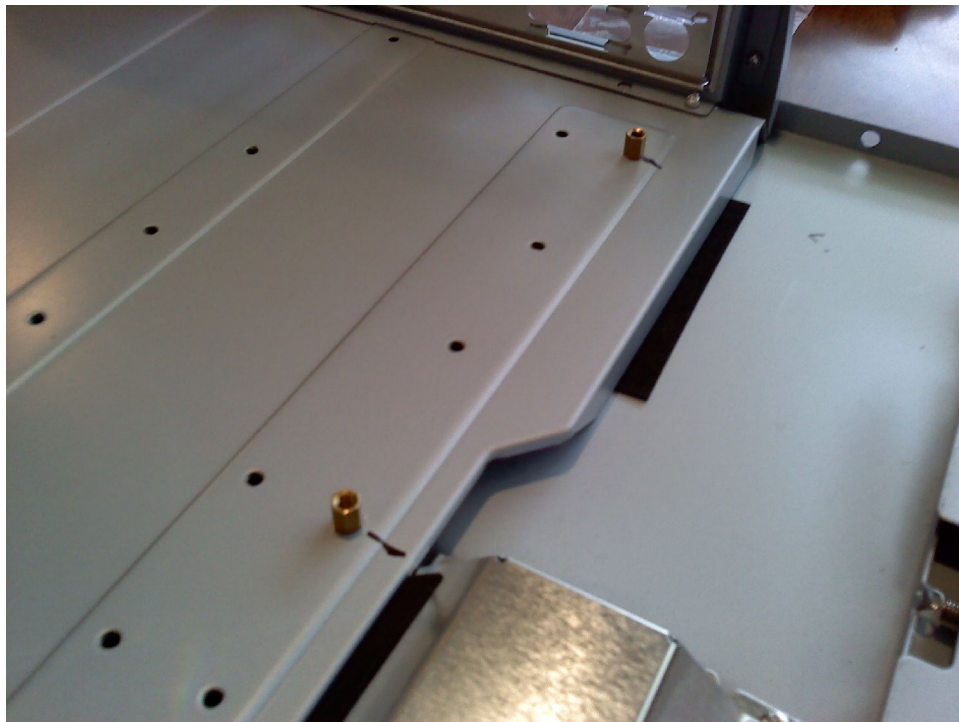
cases have raised mounting holes that remove the need for standoffs, and some may even require you to use standoffs in some holes and not others. Madness, I tell you.)

Now, get into your motherboard box, and find the I/O plate that the manufacturer bundles with the board. You're going to need this, because even though case manufacturers include a generic I/O plate for the back of the case, the chances of it matching the motherboard's configuration is about 0.00000001%. Get out your pliers, and bend away all those annoying little tabs over the individual cutouts. There's no need to break them off outright, but get them out of the way, and as flat as possible. When you're done with that, yank the case's I/O plate out, and snap the new one into place.





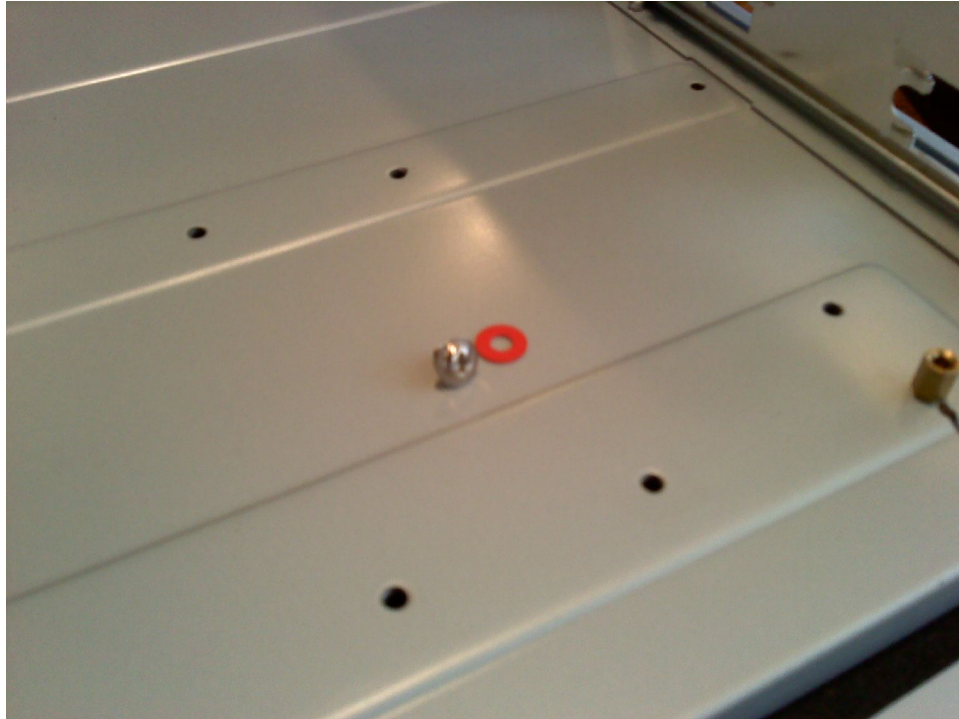
Now then, get the motherboard unpacked, and carefully place it on the motherboard tray. Get it all lined up with the I/O plate, and make sure that the motherboard's screw holes are as-centered-as-possible over their corresponding case mounting holes. (Don't be too worried about this, because you may not be able to get an exact fit without actually having your standoffs screwed in anyway.) At this point, you can get out your marker and make some hash marks near the needed mounting holes, so that you can figure out which ones need standoffs. (You can see what I mean below.)

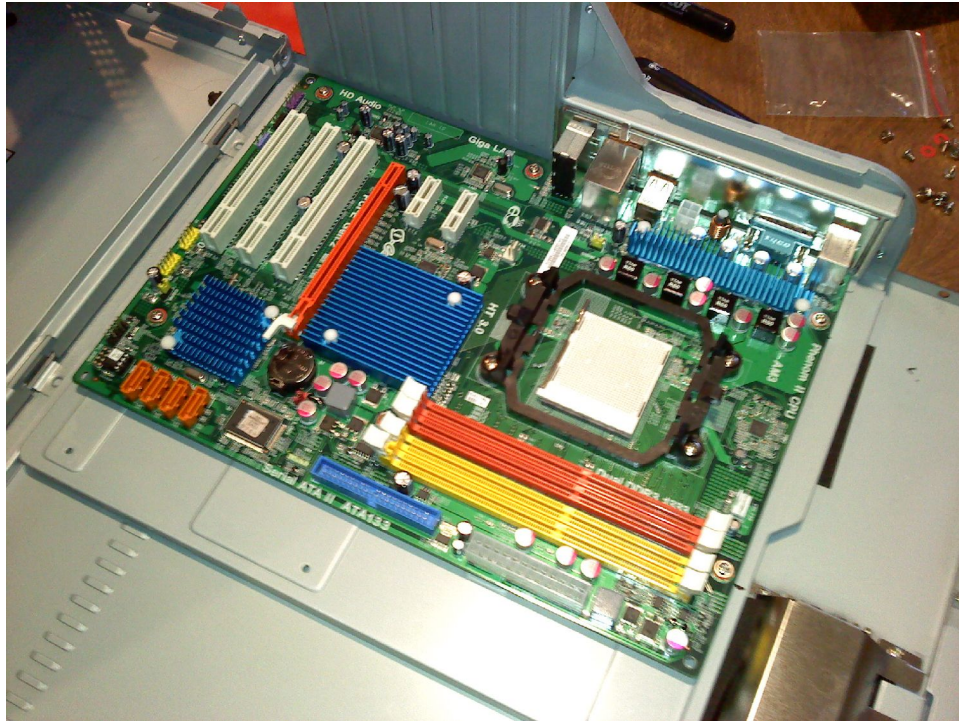


At this point, you'll need to fish around inside the hardware that came with the case, and find

yourself as many pan-head screws and washers as there are stand-offs being used. Get the motherboard maneuvered mostly into place, and then put a washer over each mounting hole. You can now start fastening the motherboard to the case with the screws. I recommend working outwards from the center mounting holes.

In general, you should leave a bit of play in the screws as you go, so that you can make small changes to the position of the motherboard. If you **don't** leave a bit of force off the screws, then you'll get to the last mounting hole, and be really frustrated when you can't get the motherboard positioned correctly above it. Even when you do get all the screws in, you don't have to tighten them down with superhuman force. "Just tight enough to stop" is plenty.

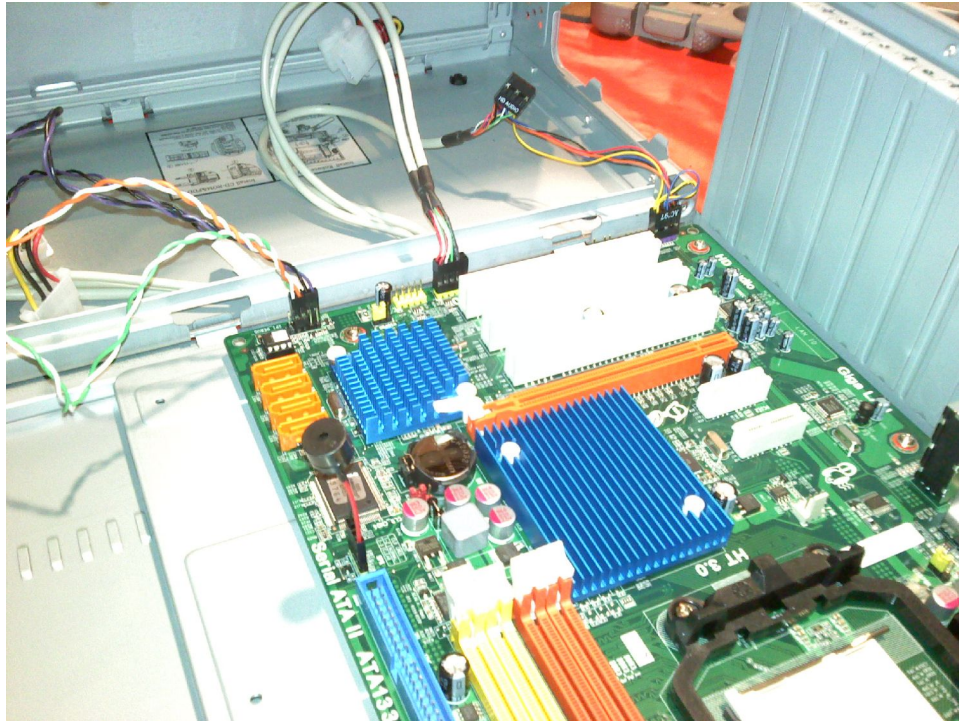




At this point, I know you're aching to get that processor slapped into that socket.

Don't do that yet.

Instead, get into your motherboard's manual (yes, you really should read the manual), and figure out what jumper positions to change (if any). After that, you should finally unwrap all those cables for your case LED's, fans, and front panel connectors, and get them connected to the right spots on the motherboard. In this case, I'm going to change the jumpers for the USB power to the 5VSB setting, with the idea that this is good craftsmanship – that is, that having standby power available at the USB ports is a “nice extra,” and good craftsmanship is all about enabling the “nice extras.” (Good craftsmanship also why you really should plug in that hard drive activity LED.)



Okay – **now** you can uncork the CPU.

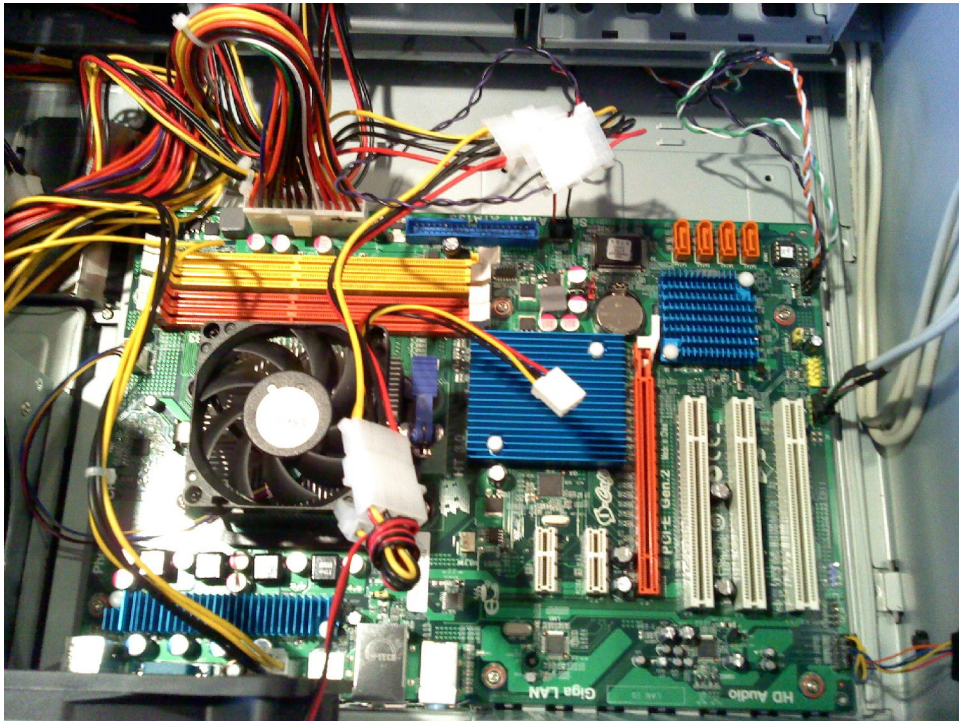
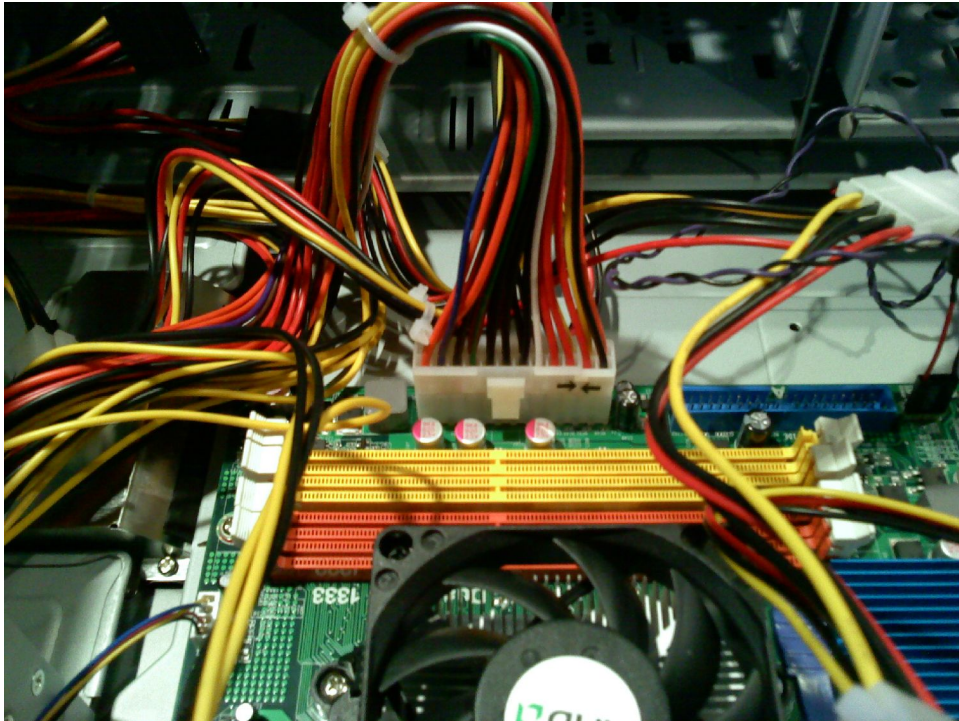
In this case, the CPU socket lever has to be all the way up, and the processor oriented so that the gold arrow mark on the CPU is on the same corner as the socket's inset arrow mark (kiddy corner to the socket lever's pivot). Be gentle when dropping the CPU into the socket – it should simply settle in. (There's a reason why they call these sockets ZIF, or Zero Insertion Force.) At this point, you can put the heatsink and fan onto the processor as well. **Also, remember to actually connect the CPU fan to the motherboard!**

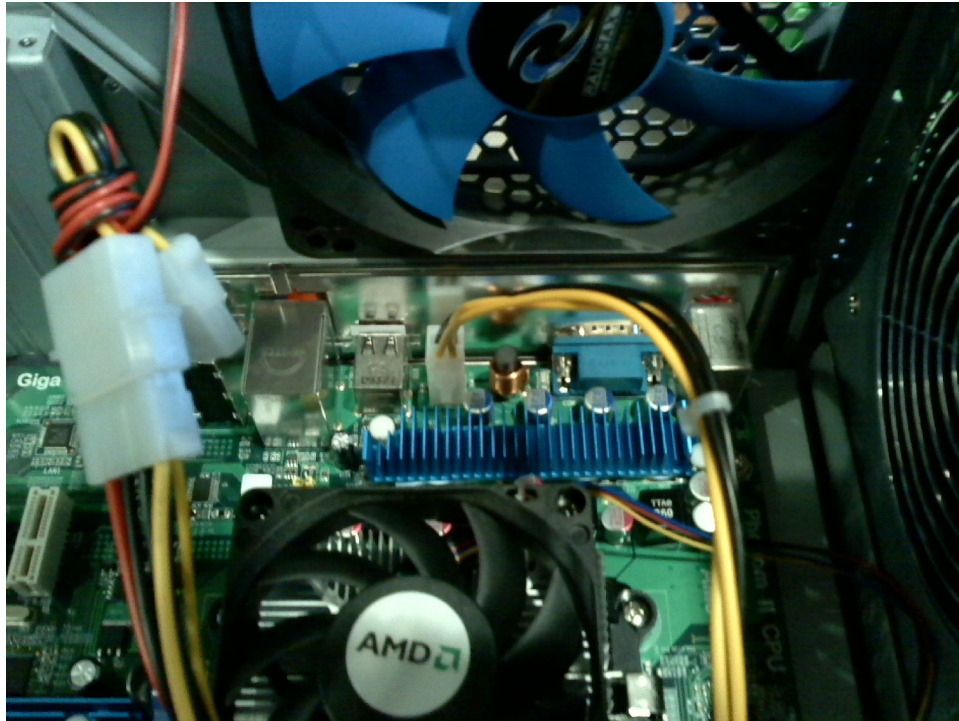
As a matter of opinion, I will also submit that you might want to double check that the CPU heatsink and fan are **solidly** anchored to the tabs around the socket. You may even want to gently push the “hooks” used by the cooling assembly a little further onto the tabs – but not so much that the heatsink loses contact with the CPU! The important thing is just to take that little bit of extra care, so that there is as little chance of the cooling assembly coming loose as possible.



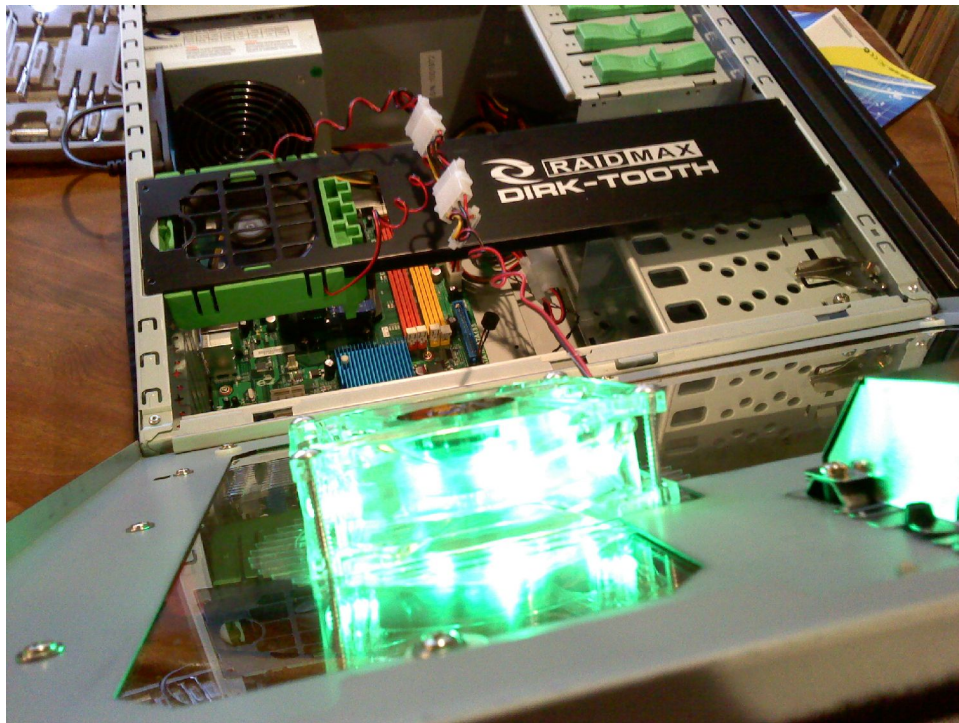
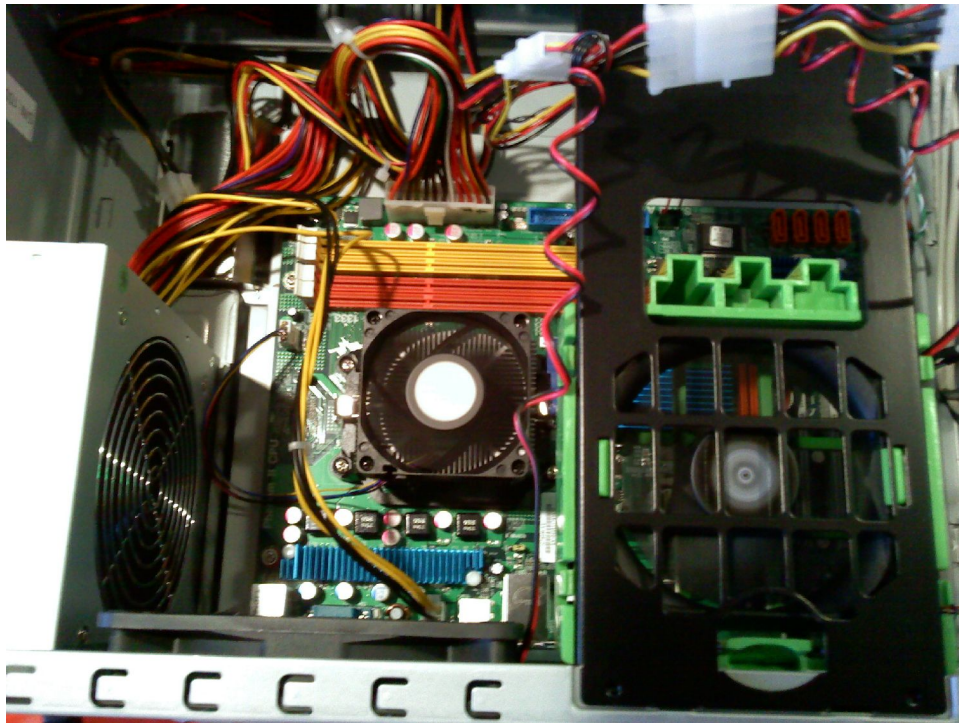
At this point, you will need to get the motherboard tray back inside the case. The reason for this is so that you can get the power supply connected. Take care to keep any loose cables from getting pinched or trapped in the process.

When plugging the power supply into the motherboard, take care that you connect **all** the relevant power feeds. It can be easy to forget to plug in an ATX12V connector, for instance, especially if the associated connection point on the motherboard is located “far” from the main power connector. It's also worth mentioning that you should do your very best to reduce the amount of connectors used from the power supply. That is, if there are components in the case with power splitters/ y-cables/ whatever-you-want-to-call-'em, you should try to use those splitters to pass power on to as many other components as possible. The reason for this is to keep as many other connectors available as possible for components that will be added later on. Don't worry about messy cabling at this point – we will be fixing that later on. (However, you shouldn't be overly sloppy either, as that will come back to haunt you.)





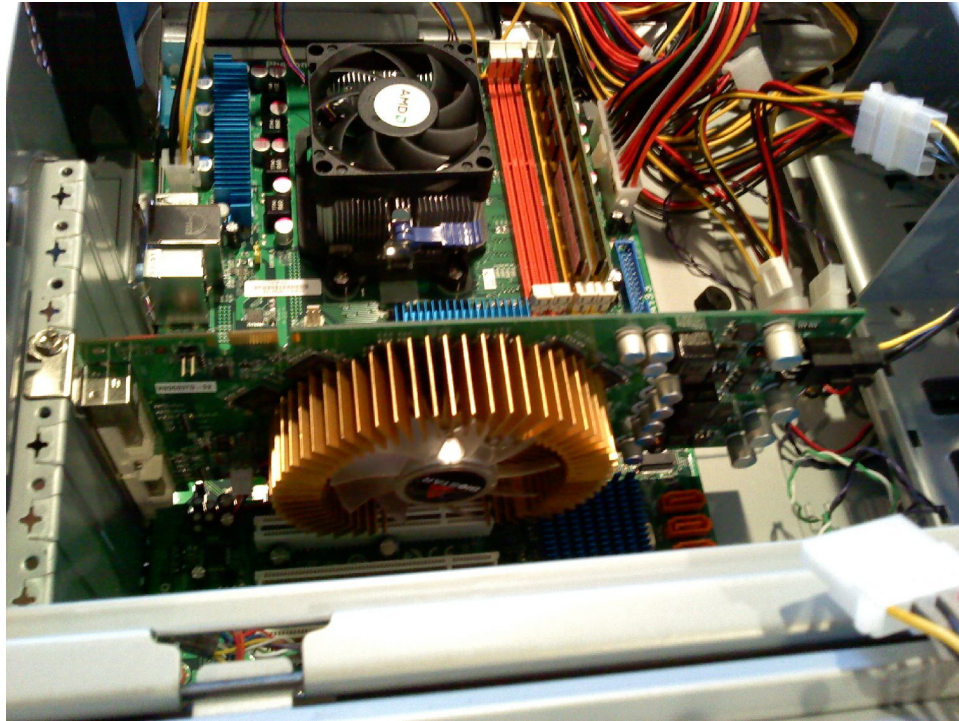
At this point, you have an opportunity to perform the first “test-as-we-go” check of the system's status. The purpose of the first check is to ensure that the power supply works, as well as to make sure that all the fans and LED's are operational. Make sure that all the fans in the system are unobstructed, and then fire up the system. Don't worry about any beeping that the motherboard makes when you energize the system – just make sure that all the parts currently connected are working. Make sure to try the reset button, and then shut the system down using the main power button. If everything seems in order, you can move on.





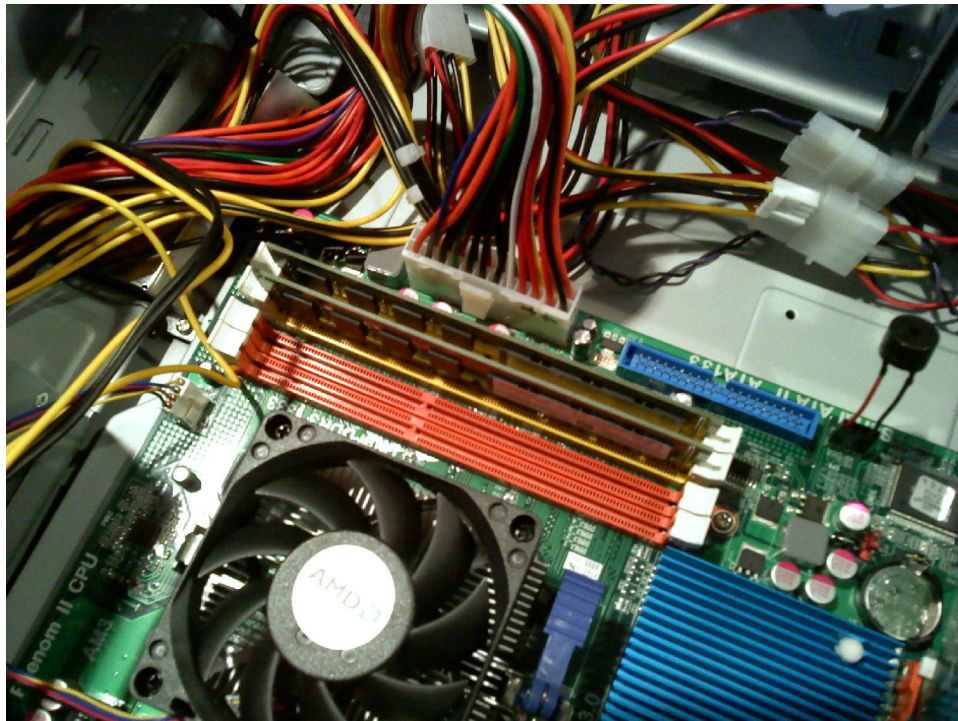
At this stage, you may need to backtrack a bit with your power connections, so that you can work on other things. However, keep as much connected as possible.

At this point, you should add the video card. As you continue with progressively building and testing the system, having some sort of visual feedback will be very handy. “Test fit” the card to figure out which slot cover to open/ break off (don't cut yourself!), and then make sure that any locking lever for the appropriate slot is set to allow the card to mate with the slot. It is probably best to connect any extra power that the card requires **before** you seat the card, because mating the power connector afterwards may put a lot of undue stress on either the card or the motherboard connector. Take care that the card is seated firmly.



Go find your monitor and keyboard, and connect them. After doing so, try starting the computer again. **Important:** Some motherboards will not allow the video card to operate unless RAM has been installed. You may have noticed that RAM modules are present in the above picture, and that is because the motherboard used for this walk-through requires them to be installed before operating correctly. At the very least, the video card's fan (if applicable) should operate when you apply power to the machine.

Installing RAM is quite simple, as long as you make sure that the modules are firmly in their sockets, and that the socket latches have snapped into place. A bit of pressure is sometimes necessary, but be sure that you are not trying to force a misaligned module into place. Make sure that you consult your motherboard documentation to find out which RAM slots to use – it is not always intuitive or obvious as to which slots will work for a given RAM configuration.



At this point, the computer should be able to boot to the point of attempting to start the operating system (which, of course, can't be done just yet.) In general, if the machine does **not** successfully start, you may have a problem with the processor, RAM, or video card.

- If the computer is not beeping, but you have no display, you should look at the video card as the most likely cause of failure. However, there may be other factors involved!
- If the computer is beeping as it was when you tested the fans and LED's, first try using a single memory module in the appropriate slot. If that solves the problem, then you have a bad stick of RAM. (Try swapping the modules to see if one definitely causes the failure consistently. You may have a stick that simply failed to seat properly the first time.)
- If you have RAM installed, but you are getting a different pattern of beeps than before you installed the memory, you might have a bad processor.

Once the machine is doing what it should, reconnect any power that you disconnected while installing the video card and RAM. This is so that you know how many connectors you actually have available for the remaining components.

You may want to stand the case on end at this point. The next device to go in will be the optical drive. Open the appropriate bay of your choice on the front of the case. (This can be a serious pain, especially if you cut yourself on something in the process. I recommend using the end of your screwdriver's handle to knock the bay's "blanking" panel out, if possible.) Different cases have different mechanisms for securing the drive, such as screws or brackets. In whatever case, make the drive as secure as the case manufacturer will allow.

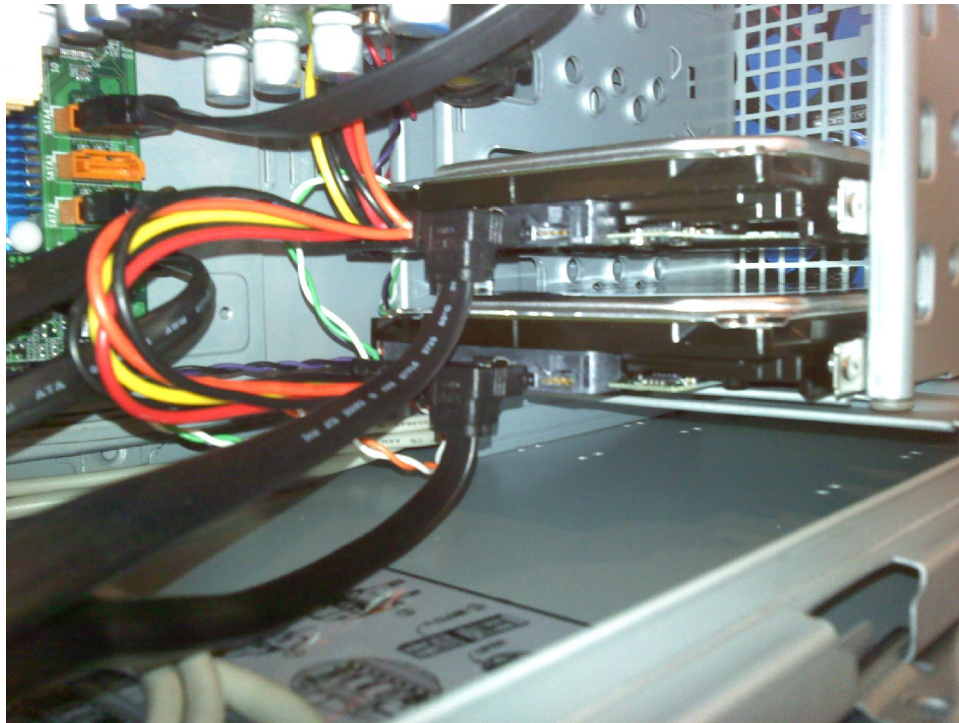
When connecting data and power, remember to **first** connect the cable that will leave the remaining connector(s) unobstructed. I also recommend putting the optical drive on a "boundary"

numbered data bus when possible. (That is, on the lowest or highest numbered bus.) This will help to ensure that the optical drive has its own bus, and is as “out of the way” of the hard drives as possible – both physically and logically. Make sure those connectors are well seated!



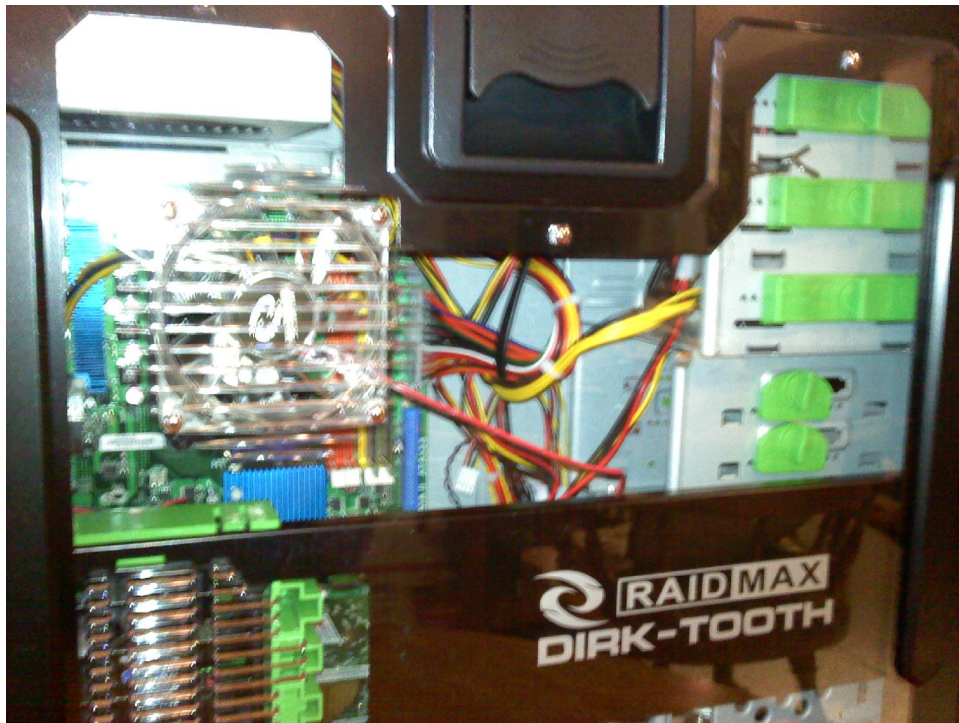
Don't get lazy here! Apply power to the computer, and make sure that the system successfully goes through POST with the optical drive installed. If it doesn't, you may have a bad (or unseated) cable somewhere. Make sure that the drive itself powers up, and can open and close properly. You may also want to get into your motherboard's BIOS setup, and confirm that the BIOS can actually see the drive. (Lots of motherboards go into the BIOS screens after the “Delete” key is pressed at the right moment – but not all do. Read the manual.)

Installing the hard drives is much the same as installing the optical drive. However, you may need to temporarily disconnect the power cable from the video card to install them successfully – or even completely remove the video card. Be sure that you are within reach of the power connectors necessary for the drives. Use the data buses that are numbered as far from the optical drive bus as possible. As always, remember to perform connections so that you don't block yourself from making other connections afterward.



Again, fire up the machine, and make sure that the BIOS can see both of the hard drives. If the computer fails to boot at all, you may have a bad cable connected to one of the drives.

At this point, you can do your cable management and general “buttoning up.” Try to tuck cables away from other components, and take special care that cables are restrained from wandering into fan blades. If you've made any temporary disconnects to clear the way for installing parts, now is the time to get everything put back together.



Just to be sure that everything has stayed in one piece, give the system one more try. If everything behaves, then the “heavy lifting” of installing the main computer's hardware is done. Huzzah!

Configuration and Basic Testing of The Main Control Computer

BIOS, RAID, and Operating System Tasks

Now that you're finally in the position of having a machine with all of its innards in place and working properly, you need to do a few things before installing the OS. First, you need to set up the “basics” in the BIOS. This entails getting the date and time set up, as well as making sure that the computer will scan the optical drive for bootable media. For this walk-through, setting up the basics also includes enabling SATA-RAID functionality.

Because every BIOS is different, it's beyond the scope of this walk-through to get into the little details of how everything is set. In general, though, I do encourage you to visit each BIOS menu. It can be very helpful to at least have some idea of what is available in each page, and to check to see if any of those options impact what you're trying to do. I **very** much encourage you to use the most “conservative” settings available when given a choice, and to allow the BIOS to use “automatic” configuration whenever possible. This isn't about squeezing every drop of performance out of your hardware, but instead about getting a working machine that is reliable.

After you've gotten the basics of the BIOS in order, you need to setup your RAID. Getting into the RAID setup menus usually requires a command different from what you use to get into the BIOS. It seems that most manufacturers prefer to use a command that includes the “Ctrl” key, but what other keys are involved can vary. (Have I said that you should read your documentation yet?) Also, you may notice that the key conventions for the RAID configuration utilities are **completely** different from the BIOS – so assume nothing.

In the RAID configuration menus, make sure that you are creating a RAID 1. RAID 0 is striping only, which has no fault-tolerance at all (indeed, if one disk fails, the whole file system is kaput). Assign both hard drives to the RAID 1 that you've designated. At this point, you don't need to worry about erasing anything, so if you get a warning message about data loss, you can confirm that you want to go ahead. Also, if you get a message asking about whether or not to use all the space on the drives, you should indeed use the entire capacity of the drives. Some RAID configuration systems require a special key-press in order to save your settings. Make sure that all your changes are being kept, and then allow the machine to restart.

Now is the moment you've been waiting for – the OS installation! Beyond your bootable installation media, you will also want to have your motherboard's software installation media handy. The reason for this is that (if you're using Windows), you may need to load RAID drivers before Windows is actually installed. In the case of the hardware and software used for this walk-through, Windows 7 appears to see the RAID without any fuss over drivers at all.

Once your OS has been convinced to install itself on your RAID, you can move on with the install. Most of this will probably be automatic, so you can go have a sandwich. Or two.

Settings

The generalized version of this is simply to go through every setting available in the operating system, and customize it to your needs. Pay particular attention to settings that impact security – try to enable as many security features possible. For instance, if your OS allows you to prevent people from

seeing the name of the last user who logged in, you should enable this. (More advanced versions of Windows may allow you to use “Administrative Tools” to change the local security policy, which lets you do things like enforce unique passwords, ctrl-alt-del before login, and other things.) Also, you should require user authentication whenever the computer is activated after sleep/ suspension/ whatever-you-want-to-call-it. In terms of usability, you should **prevent** the machine from launching a screensaver, and also prevent the machine from shutting down any hardware during idle time. (This is because you don't want to be doing a show where you sit for a while, and then need to make a quick lighting change...only to find that the machine needs to “spin back up.”)

The changes I make to Windows machines are as follows:

1. Change the desktop background to something that looks cool, and isn't the default background image. (Yeah, yeah, it's cosmetic – but leaving the default background unchanged is sort of an “amateur” thing to do.)
2. Change the refresh rate of the monitor to the highest available setting. (This can sometimes help with display issues on some flat-panel displays. **Sometimes.**)
3. Go into “Action Center,” and select “User Account Control Settings.” Change the notification settings to “Always Notify.” (This adds a very small layer of extra security, because you are forced to take notice of “goings on” with the computer a bit more.)
4. In “Internet Properties,” go to the “Security” tab and change all zones to have the highest security setting, and also to run in “Protected Mode.” (This may seem a little overboard, especially since I am **not** going to use Internet Explorer for web access on this machine. However, I want to make every possible avenue of attack as, shall we say, “narrow and bumpy” as possible.) Next, go to the “Privacy” tab, and set the privacy level to “High.” (The highest setting is actually too restrictive, especially since so many sites use cookies to manage logon sessions.) Also, you should check the box that disallows data gathering from “InPrivate” browsing.
5. Go into “Mouse,” and set the pointer speed as high as it can go. (If you need to make a quick change during a show, you want to use as few mouse movements as possible to get around. Of course, you shouldn't set the speed any higher than you are comfortable with. Enabling “Enhance Pointer Precision” may be helpful.)
6. In “Power Options,” locate the option to require a password on wake up, and enable the password requirement. (This is important, and I can't believe that it isn't on by default. You don't want to walk away from the machine, having put it to sleep, and then have just anyone be able to wake up and use the computer with your access privileges. I mean, DUH!) Next, go to “Create a power plan,” and select the option to create a high-performance plan. Change the plan's settings so that the computer never shuts down the monitor, and never goes to sleep automatically. Also, in the sub-window for advanced settings, you should set the hard-disk to never be shut down. (Why this isn't in the main window, I have no idea.)
7. In “Programs and Features,” go into the sub-window for turning on and off Windows features. I'm going to disable the features that govern games and tablet PC functions, but leave everything else. (I'm not going to disable Internet Explorer, as you never know when it will be handy to have around.)
8. In “System,” go into the “Remote settings” sub-window and disallow Remote Assistance. (This is another potential security hole that could be exploited.)

9. In “User Account Control,” you should set a password for yourself that is more than 8 characters, and contains letters and numbers (it's even better if you mix the cases of the letters). I should technically recommend that you not give yourself Administrator privileges, but that makes things a little too much of a pain sometimes. (As much as I do like to have good security, I also need to have good usability. Having User Account Control pop-up a “speed-bump” message when something is being changed should be enough prevention for most problems.)

Updates, Drivers, and General Utilities

At this point, you should connect your machine to a network with Internet access, and get your OS fully updated.

WHOA! WHOA! You're going to use a trusted network, right? One that's actually behind a firewalled router, with other security set up? (Another way to read this is, refrain from using the router you bought for this project for now.)

Anyway.

You should also make sure that all your device drivers are “the latest and greatest” available from the manufacturer – with one exception. I recommend getting your video card drivers from the manufacturer of the **chipset** (say, ATI or nVidia), rather than the card assembler. (Of course, the utilities provided by the card assembler may still be helpful – installing those is up to you.) The point of all this is to make sure that as many bugs are patched as possible, and that as many features are enabled as possible before going into the “shakedown.”

Note that you should do this **before** attaching the last pieces of hardware, like the card reader and the DMX interface. However, once those drivers are installed, you can go ahead and get everything attached. This would also be a good time to load up your light control software, VNC software, and any other software that you might want, like alternative web browsers.

For the machine used in this walk-through, the first step is to find the “Windows Update” control-panel, and explicitly tell Windows Update to download and install any outstanding updates. Restart the machine when prompted, and then install the latest version of Mozilla Firefox. (You may discover that your security settings have disallowed downloads. In order to fix this, you can define a “Custom level” of security in your “Internet Options.” Scroll down the list of options until you find “File download,” and select the “Enable” option.) When installing Firefox, allow it to become your default browser. You should also get a copy of Acrobat Reader, so that you can actually open PDF files.

Oh...yeah, I forgot that you should download the latest version of Java, as well. Don't let it install any annoying toolbars or anything. (“Rider” software is one of the most annoying things that legitimate software companies do – but I guess that enough people install the junk for them to keep asking you if you want it...)

At this point, you can get the latest drivers and utilities from your motherboard manufacturer, and graphics chipset manufacturer. You don't have to get absolutely everything, but you may find that getting “everything but the kitchen sink” keeps you from missing one file or another. As you

progressively work through installing the drivers and utilities, let the machine restart each time it asks. (I view this as getting each piece working at 100% before moving on to something else.) In some cases, you may find that the drivers you download do not have an installer. If you end up in that situation, make sure that the driver folder is sitting on your desktop, and is decompressed. If you go into the “System” control panel, you can open the device manager, and find the appropriate device. Right-clicking on the device should get you a context menu with “Update driver software” as an option. A series of windows should be brought up that allow you to navigate to the driver folder on your desktop, and if the driver is correct, Windows will install it when you confirm your choice.

Important: You should generally refrain from installing management utilities that allow for access across a network. They are a very neat kinda thing, I won't deny, but we want to keep a very tight lid on what can access this computer remotely. **There is an exception to this, and that is to install whatever management system is required for a system-critical component, like the RAID – even if that management system is network enabled.**

If you need a driver for the card reader, install it now. After you install the driver, connect the card reader and make sure that it actually works. (If you don't have an actual memory card handy, just make sure that your machine's hardware manager isn't reporting any problems with the device.)

The DMX Interface

This is the point where you can finally install the software for your DMX interface, get the interface connected, and check its functionality. I highly recommend making use of any and all utilities supplied by the manufacturer for checking that the device actually works.

Lighting Control Software

Once the DMX interface appears to be in working order, you can move on to installing your actual lighting software. (Finally.) You should definitely check to see if the software can communicate with your DMX interface, but you should also resist the urge to start connecting lights right away. Everything in good time – get finished with all the tasks for the main control computer's setup first, so that you can perform real testing on the system in a “completed” state.

For this walk-through, this next step entails the install and configuration of Freestyler. Just install the software with the default settings, and try launching the program when everything finishes. If Freestyler causes a Windows Security Alert, simply take whatever action is necessary to **dis-allow** Freestyler from communicating over networks. (Yes, the default installation includes DMX to Art-Net functionality, but all of that is beyond the scope of this walk-through.)

At this point, you can begin following whatever “first start” procedure the software uses, or perform whatever setup is required manually.

Remote Control Software

This is the last major piece of the software puzzle. The real pitfalls are just to make sure that you understand what you're installing, and how it's supposed to work. Some remote access systems need to be manually started when you log in, and some can run automatically. Some may have some “neato” features that you can enable – but be careful of allowing your software to do unnecessary things that make the system insecure. For this software especially, you need to be aware of whether or

not your OS will block connections to it by default, and if so, you will need to allow those connections.

Setting Up The Network Hardware

Before getting into the process of setup, there is one foundational issue to discuss – to DHCP, or not to DHCP? DHCP (Dynamic Host Control Protocol) allows for “painless networks,” where a host that connects to the network can be automatically configured. This is a good thing in general, even though it slightly decreases security, but it has one glaring drawback. The IP addresses of your computers can change, which may complicate getting the remote device connected to the main computer's VNC server. Because there are only two user-level network hosts in this walk-through, I'm just going to save myself the trouble of doing manual addressing on the computers. However, this may not be the right choice for you – although, from a security and reliability standpoint, you really shouldn't be allowing a whole bunch of non-lighting-control computers on your control network.

Now then...

The first thing to do is to get the wireless router physically connected to the main control computer, and powered up. The very first thing you should do after a successful boot is to launch a browser on the control computer, and connect to the router's default address. The router used for this walk-through uses 192.168.2.1 for its default IP. (Some routers use 192.168.0.1, but the bottom line is to consult your documentation before you have a fit of rage over not being able to connect to the router.) Log in with the default Admin user name and password, and then change the default password (and user name, if possible) to something long and hard to guess. (You'd better believe that there are lists of default user names and passwords that are easily accessible via the web. All your other security settings will be for naught if somebody can just log in to the router and change everything.) In the case of the router chosen for this walk-through, you can make these changes in the “System Settings” page.

The next thing to look at is the router's wireless SSID (Service Set Identifier). This is the name of your wireless network, and it too has a default setting that you should change. I recommend changing the SSID to something fairly long, minimally descriptive, and somewhat unwieldy (containing letters and numbers). I also recommend disallowing the router from broadcasting the SSID. In concert, these two actions yield a wireless network that is a smaller target for mischief. It isn't shouting its name to everybody within range, and if people are looking for a “default configured” network to mess with, yours won't be that network. Having the name be minimally descriptive and a little tough to decipher is helpful, because it doesn't provide folks who do discover the network name from figuring out too much about what it's doing. (For instance, using an SSID like “West2187” may still help you identify your network, but is far less tempting a target than, say, “LightingControl.”) For the walk-through router, this can all be done in the “Channel and SSID” page.

At this point, you should configure your wireless encryption. As far as I am concerned, you should use the strongest and most up-to-date security options available. (For instance, WPA2 with AES encryption was the best option available when this walk-through was written.) You will need to pick a pass-phrase, and because this is a single pass-phrase that the security of the whole system rides on, you should be even more strict with it than other passwords. I recommend nothing less than 16 characters of indecipherable gobbledygook. The walk-through router handles this configuration step in the “Security” page.

A quick word about Wi-Fi protected setup: **Refrain** from using the push-button method, which lets you initiate a setup “handshake” between the router and a wireless device by simply starting the process on both devices within a certain time-frame. Although unlikely, somebody within range could

join your network, just by pushing a button before the time expires. Use the PIN method instead, because it's very unlikely that two devices will get the same PIN number at the same time.

Once all of the above is accomplished, you can get just a bit more security by restricting the wireless network to only known MAC (Media Access Control) addresses. (You do, of course, have to find out the MAC address of the wireless adapter being used by your remote device. The MAC address is sometimes printed on a sticker affixed to the wireless device itself, but if not, you can find the address using the network configuration tools for your remote control device after the adapter is installed properly.) Taking this step makes life a bit tougher for an attacker, because even if they manage to break through your wireless security pass-phrase, the system will disallow them from connecting. (It is entirely possible to fake a MAC address, but figuring out which MAC address to fake is difficult if you can't actually read packets from the network.) For this walk-through, you can set your allowed MAC addresses in the “MAC Address Control” page.

There is also yet another layer of security that you can add, and that is network-wide MAC address filtering. This is really the same functionality as the above, except that it applies to both the wired and wireless network. This helps if an attacker gets really ambitious, and tries to connect via a physical cable while you're not looking. Although unlikely that a random attacker would actually try this, setting up this extra bit of security is so easy that you might as well do it. The router in this walk-through puts this functionality in the “MAC Address Filtering” page.

The last step to take is to make sure that the router's firewall is actually enabled. I very much recommend that your wireless router not actually be connected to a WAN (Wide Area Network) except when absolutely necessary. Still, you may find it necessary to get connected at times, and you want as much protection available when that connection is made. For this walk-through, the router's firewall is automatically enabled, but checking the status (and you should check it, just to be sure) is as simple as going to the main “Firewall” page.

As a final word, I will say that you can be even more hardcore about security than what is presented in this walk-through. Some wireless routers allow you to reduce their signal strength, which shortens the range of their traffic visibility. You may be able to disallow all wireless protocols other than the one you will be using for the remote control device. You might be able to setup the router's DHCP server to only hand out two addresses (one for the main computer, and one for the remote). Personally, I feel like these extra measures don't really have enough of an impact to be worth chasing after – and there's also the issue of the walk-through router not allowing for reducing its signal strength. The bottom line is that I think the above points have covered enough bases for making the router an inconvenient target for a random “idiot” looking to get on the network uninvited, and I think that's what really matters.

Setting Up the Remote Control Device

Don't worry. We are getting **very** close to playing with some actual lighting – we just have to get around this bend. The first step in getting around the bend is to get the remote control device talking to your wireless network. If the wireless adapter isn't already installed, load its drivers and get it connected.

Unless you have a specific reason to do otherwise, I recommend letting the remote control device's OS manage your wireless connectivity.

If you've enabled all the MAC address related security that was talked about earlier, now is a good time to get the MAC address for the remote device's wireless adapter put into the appropriate places in the router's configuration pages. In order to get the MAC address, I did have to open a command line and invoke "ipconfig /all" - but that was because I didn't want to disconnect the wireless adapter to look at its MAC address sticker. (Being lazy usually requires you to do more work, in one way or another.)

Once you've gotten to this point, you'll need to do some manual configuration of your wireless adapter. You'll need to supply the SSID manually, as well as the wireless security pass-phrase. You may also need to explicitly specify that connections are allowed if the network is not broadcasting, and also what kind of encryption to use.

Once you get a connection established, restart the remote control device to ensure that it will reconnect after being rebooted.

Now that you've got a reliable connection, you can go ahead and get your VNC/ remote access software set up on the remote device. **Important:** If your remote control device has a different processor architecture than your main control computer (say, a 32-bit processor versus a 64-bit processor), you **can not** simply reuse the install files from the server. You **must** obtain the correct installer for the remote device's processor architecture. Also, a key thing to remember is that you only need the client software – not a full server install.

Here's the point where things get really interesting. Fire up the main control computer, and ensure that its VNC/ remote access server is started. Make note of the main control computer's IP address, which is usually easy to find via whatever you use for network configuration.

On the remote control device, start up the VNC viewer, input the IP address of the VNC server, and **try to make a connection without encryption enabled**. The idea is that you should **not** be able to make a connection properly without the encryption turned on – if you do make a successful connection, something is very wrong. Restart your VNC viewer, and enable the correct settings for encryption. Try making the connection again. After a few seconds of negotiation, you should get an authentication dialog. Enter the correct password (or, in this walk-through's case, enter your login information for the main control computer). You should be allowed to take control of the main computer's screen.

At this point, you should try opening Freestyler. If the display is slow, or overly blocky, you may need to experiment with your connection settings in order to avoid having to explicitly refresh the screen all the time.

A word of warning: Encryption may or may not be something that works smoothly with a Java-based viewer.

The very last thing to do is to perform a quick "range" test of the remote device. Get a VNC connection to the main control machine going, and then take the remote for a little walk. Try to find all the places where having the remote device would be handy, and make sure that you have a solid, usable level of control in all those locations. Try connecting and disconnecting from the VNC server multiple times. If you lose too much signal when you're in critical areas, you may need to expand the coverage of your wireless network. If you've reduced the output power of the wireless router's signal, then increasing the power may help. If the power is already at maximum, you may be able to buy an

improved antenna for the wireless router (although not all routers have interchangeable antennae). In the worst-case scenario, you may need to install additional wireless access points.

A last word on security: Whenever you are leaving the main control computer unattended, you should ensure that the machine is locked. (In Windows, you can do this by holding the “Windows” key, and pressing “L”.) If you do this all the time, you will start doing it automatically – and it is important to do it as a matter of habit if you are every called away from the main computer during a show. Similarly, you should always log out of a VNC session running on the remote device (and also lock the remote) if you are going to leave the remote unattended. Some security, like encryption, can effectively be done for you – but some things are entirely determined by your actions. Be safe!

Setting Up and Testing Your Lighting Instruments and DMX Lines

Finally! It's time to actually get our hands on the “business end” of this whole shebang. Get your lights, DMX cables, and accessories unpacked, assembled, and ready for action. As a general point, use all the parts that the manufacturer has sent along. If there are washers meant to go between screw-heads and another surface, **use them**. Remember, we're going for good craftsmanship here. The lights that I picked for this walk-through have a “yoke” that attaches via a couple of screw-mounted knobs. (Unfortunately, in my case it was very easy to get the screw-knobs misaligned, so it took a couple of tries per light to get the yoke to cooperate.)

If at all possible, you want to run a test on your lighting fixtures **before** you make any DMX connections. Why? Because you want to know if anything is not working correctly before you introduce any more complexity into the equation. If you just dive in to getting everything hooked up, and then you get strange behavior, you don't know if the light fixture is the problem, or if there's something wrong with the DMX configuration, or... (You get the idea.) The more comprehensive a test you can do, the better. Consult the documentation for your light fixtures. (In my case, no documentation came with the light fixtures at all, and I had to go to the manufacturer's website to figure out how the lights operated.) The lights for this walk-through allow for manual illumination of each LED color, so it's pretty easy to check that everything is working. As a side note, **refrain** from looking directly at the light when you plug it in. Some lights may flash all the LED's when they first receive power, and it can be a rather...bright experience.

Here is a series of pictures depicting a light with all its LED's working, and the DIP switch settings necessary to test the LED's. Please note that different manufacturers will implement different DIP switch configurations – this is just to show you what can happen with a particular lighting instrument.





Once you've determined that all the fixtures are in working order, you can configure them to have the correct DMX addresses. This is another process that may vary from manufacturer to manufacturer. Some lights have an extra DIP switch that prevents DMX from being received unless the switch is "on," so take the time to figure out if this is the case.



Now...here's where things can get sticky. The manufacturer lists these fixtures as having 5 channels. This means that the address for each light must work out to being 5 numbers higher than the previous fixture. That is, the addressing should go 1, 6, 11, 16.

Why is that?

DMX is a multiplexed protocol, which means that everything is sent down the same signal conductors (or bus, if you'd rather call it that). The fixtures have to be set up to receive **only** the information that is on separate channels from the other lights. If all the lights have the same address, they will all “listen” for commands on the same channels, and all the lights will do the same thing. This can actually be useful in some cases – but that's not what we're going for here. (Another way to look at this is if you have multiple TV sets in a house, all capable of receiving the same cable or antenna signal. Each TV can access all the available channels, but one TV might be tuned to channel 4, while another is tuned to channel 7.)











The tricky part of all this is figuring out which switch to flip to what setting, especially if the manufacturer doesn't provide a reference chart. This all comes down to counting in binary.

Okay...so how do you do that?


The key thing to remember for counting in binary is that each “digit” has only two values, those values being 0 and 1, or “on” and “off.” Each digit that you add doubles the range of numbers you can express. For instance, having one digit means that you can express two values (the aforementioned 0 and 1), whereas two digits lets you have four values (00, 10, 01, 11). The decimal equivalents of those binary numbers are 0, 1, 2, and 3, respectively. (That's another thing that can trip you up – you start counting from 0, instead of 1. Even more confusing is that DMX addressing starts from 1, not 0. Madness!) A handy thing to remember is that each digit/ bit can be assigned a value that is a power of 2, and when that digit or bit is “on,” that value is added to the values of the preceding bits.

Here's a chart to make this a bit easier to comprehend.











Fixture 1 (Starting Address 1):

Switch Number	1	2	3	4	5	6	7	8	9	10
On										
Off										
Max. Value	1	2	4	8	16	32	64	128	256	DMX Enabled





Fixture 2 (Starting Address 6, or 0+2+4):

Switch Number	1	2	3	4	5	6	7	8	9	10
On										
Off										
Max. Value	1	2	4	8	16	32	64	128	256	DMX Enabled

Fixture 3 (Starting Address 11, or 1+2+0+8):

Switch Number	1	2	3	4	5	6	7	8	9	10
On										
Off										
Max. Value	1	2	4	8	16	32	64	128	256	DMX Enabled

Fixture 4 (Starting Address 16, or 0+0+0+0+16):

Switch Number	1	2	3	4	5	6	7	8	9	10
On										
Off										
Max. Value	1	2	4	8	16	32	64	128	256	DMX Enabled

Now – I know what you're thinking. Yes! It is **not quite time** to connect lights to the main control computer yet. First, your control software has to have fixture profiles available to properly control your lights. Some programs have libraries of fixture profiles that are ready to go “out of the box.” However, that only helps you if your fixtures are established units from major manufacturers. Fixtures like those used for this walk-through require you to set them up manually. Some software will have you set up fixture profiles in the main program, but other packages have a separate utility for this purpose.

Once you have your fixtures defined, you can actually set up a control configuration that includes them.

Okay.

Now.

Now is the moment when we **finally** get to plug in a light. (Only one, though!)

Wait...why only one?

The idea here is that you want to do an “iterative” test. Just like testing other parts of the system, this helps you find a problem while things are still as simple as possible. If a problem pops up with only one light attached, then you have a reasonably restricted space for that problem to reside in. If all the lights are attached at once, and one starts doing something odd (or doesn't respond, or attempts to prevent you from opening the pod bay doors – yes that's a movie reference...) then you have a much larger range of possible problems that might be occurring. That is to say, if everything works beautifully with one light, and then quits working properly when one more is added, you can focus your effort on figuring out what's wrong with the second light (or its connection). If all the lights are connected, and a couple are acting up randomly, you have to start taking the setup apart to isolate the problem anyway. You might as well start with the simplest case, and then work up.

Oh, and when you connect the lights, use the longest cables first. If there are any problems related to cable length that are lurking in the shadows, this will help them to show up as early as possible.

In general, what I recommend doing is going through each fixture channel individually, making sure that the channel the software is showing you corresponds to what the manufacturer says it should do. If you make changes to a channel, and the fixture does something that seems to correspond to another channel, you probably have an addressing problem.

Now, find the next longest DMX cable, and the number two light fixture. Connect the first fixture's DMX out to the second fixture's DMX in. **Refrain from testing the number two light!** First, re-try all your tests with the first light. If the second light starts responding to your inputs, you probably have a DMX addressing problem. (For instance, if you're manipulating the individual color channels, and the second light starts scrolling through its strobing settings, you almost certainly have a mistake in the second light's DMX address.) If the second light starts doing odd things randomly, you may want to try connecting a DMX terminator to its DMX output. (You may also have a bad cable, so try swapping the cable if the terminator doesn't work.) If the light still won't cooperate, try a different fixture (remembering, of course, to set its address appropriately). After you've re-verified the correct operation of the first fixture, repeat all your tests using the second light. If the second light is behaving normally, add the third. Again, re-start all your tests with the first light, and go from there.

Keep going until you run out of lights. Or cables.

It's true that this testing style can take some time, but if everything makes it through, you are much more assured of a reliable, predictably responding setup.

The Full System Test

While the last bit was potentially the most exciting piece of the puzzle, this last step is what really puts the final layer of finish on the whole thing. This is where you're going to get everything going in a configuration that's as close to real life as possible...and then try different methods of breaking the setup.

Fun!

The first thing to do (if you haven't done so already) is to set up the lights in the positions that they will be in during productions. Also, make sure that the main control computer and all network hardware is running off the UPS.

Now, take a few minutes and write some chases for your lights. Set them up so they can be looped. The chases don't have to be elaborate, but they will be a handy way for you to see if control is interrupted when we simulate a couple of failures.

Now that you have a sequence (or two...or 50) that you can run, launch your VNC/ remote access server on the main control computer. After it has started, connect with your remote control computer and try playing with your lighting fixtures. Now is the time to note any strange behaviors, or things you dislike about using the remote. (Now is also the time to fix those problems.)

Once the behavior of the remote is pleasing to you, it's time to simulate some problems.

The first thing to do is to have the remote connected to the main control computer, and actively able to control your lighting software. Start a chase or sequence, and let it run as normal. While the case or sequence is running, find your UPS...and disconnect it from mains power. The UPS, if working properly, should provide you plenty of time to:

- Calmly use the remote to stop the sequence.
- Disconnect the remote from the main control computer.
- Save anything in your lighting control software that needs to be saved.
- Shut down your control software.
- Shut down the main computer.

If the UPS does not allow you to do this, you may need a higher-rated UPS (or your UPS might be faulty). The way to tell the difference is to try pulling the UPS power again, but with only the networking hardware powered up. If your UPS can't supply emergency power to the networking hardware (which shouldn't draw very much wattage), then the UPS is very likely to be faulty.

You should note what happens to the lights when the main computer is shut off, just so that you're prepared for what happens. In the case of the setup for this walk-through, the lights simply stayed lit in the way they were when the scene was stopped. Obviously, a building-wide loss of power would shut down your lights (unless you have a UPS or generator to run them from), but it's just generally a good thing to be familiar with how all the parts of your lighting system behave when

something unexpected happens.

At this point, you should restart your main control computer, as well as your remote device. Get the remote device re-connected, and start your sequence running again. (An important thing to note is that, if your lighting control software is like Freestyler, the lights will be reset when your software is re-started.) Once you get your sequence running again, open up the main control computer's case. Find the most-conveniently-glommed-onto hard drive cable, and yank it out of the drive. (This simulates a drive failure.) The point of this test is to see if the RAID allows you to keep working with a drive down. Ideally, the main computer (and remote device) will stay responsive after this happens, and allow you to create and save a new sequence (you just want to test that you can write to the RAID).

If, on the other hand, the main computer locks up when you try this, then just reboot it. If the reboot is successful, then you should simply note that the reboot was required in order to regain control of your lighting system. It's true that this isn't ideal, but it's a lot better to be able to do an immediate reboot and “go on with the show,” than to not be able to continue at all. If you can't even reboot successfully, then something is probably wrong with your RAID system. If you're lucky, it's just mis-configured. If you're unlucky, you may need to look into getting a different RAID controller, and reinstalling Windows. (For the system used in this walk-through, the result of pulling a SATA cable from one of the drives was not even a hiccup.)

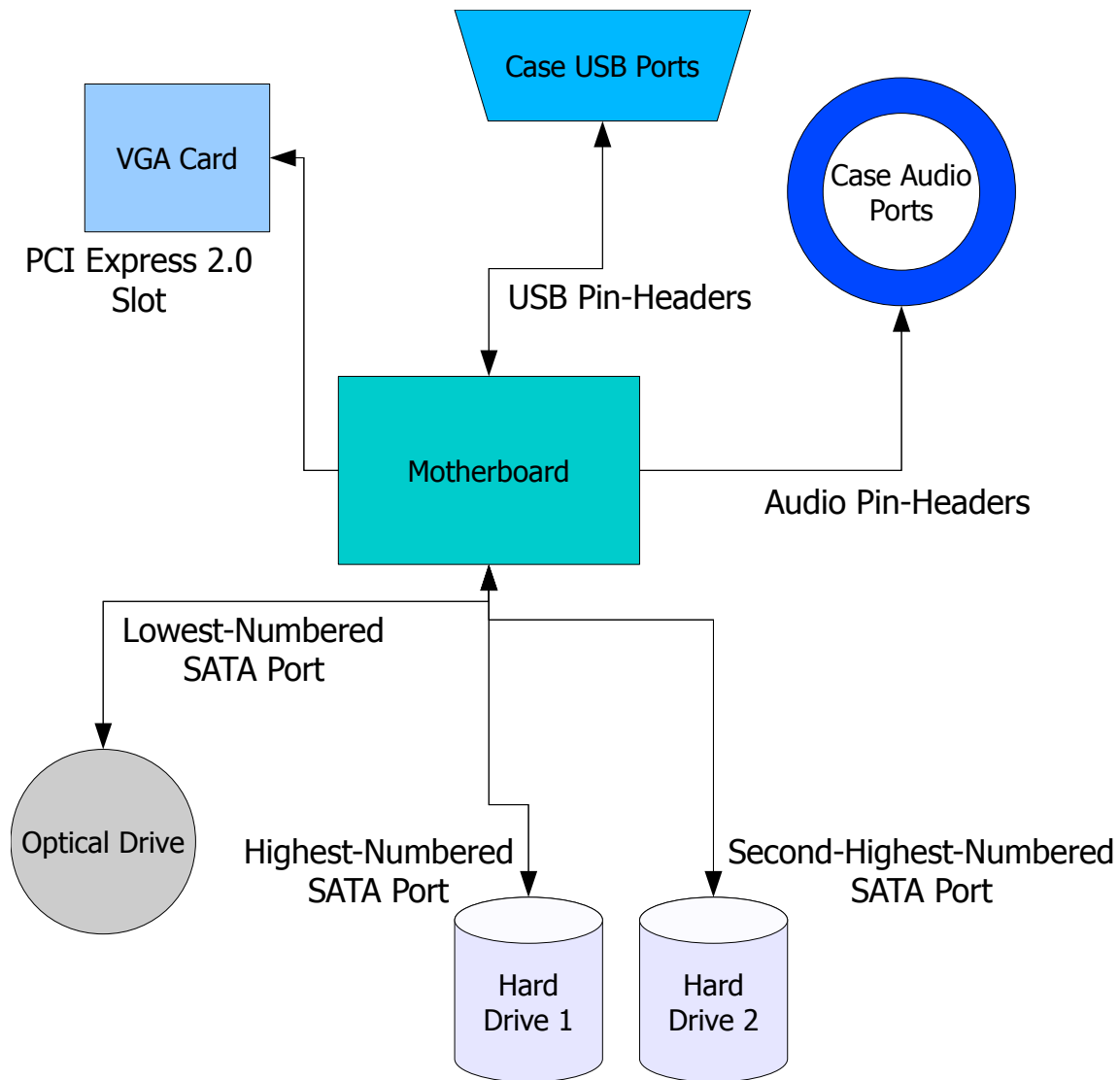
Important! Remember to rebuild your RAID after trying this particular stunt, or you won't actually have a RAID at all (when it really counts).

So – there you are! Your system is working and ready to go.

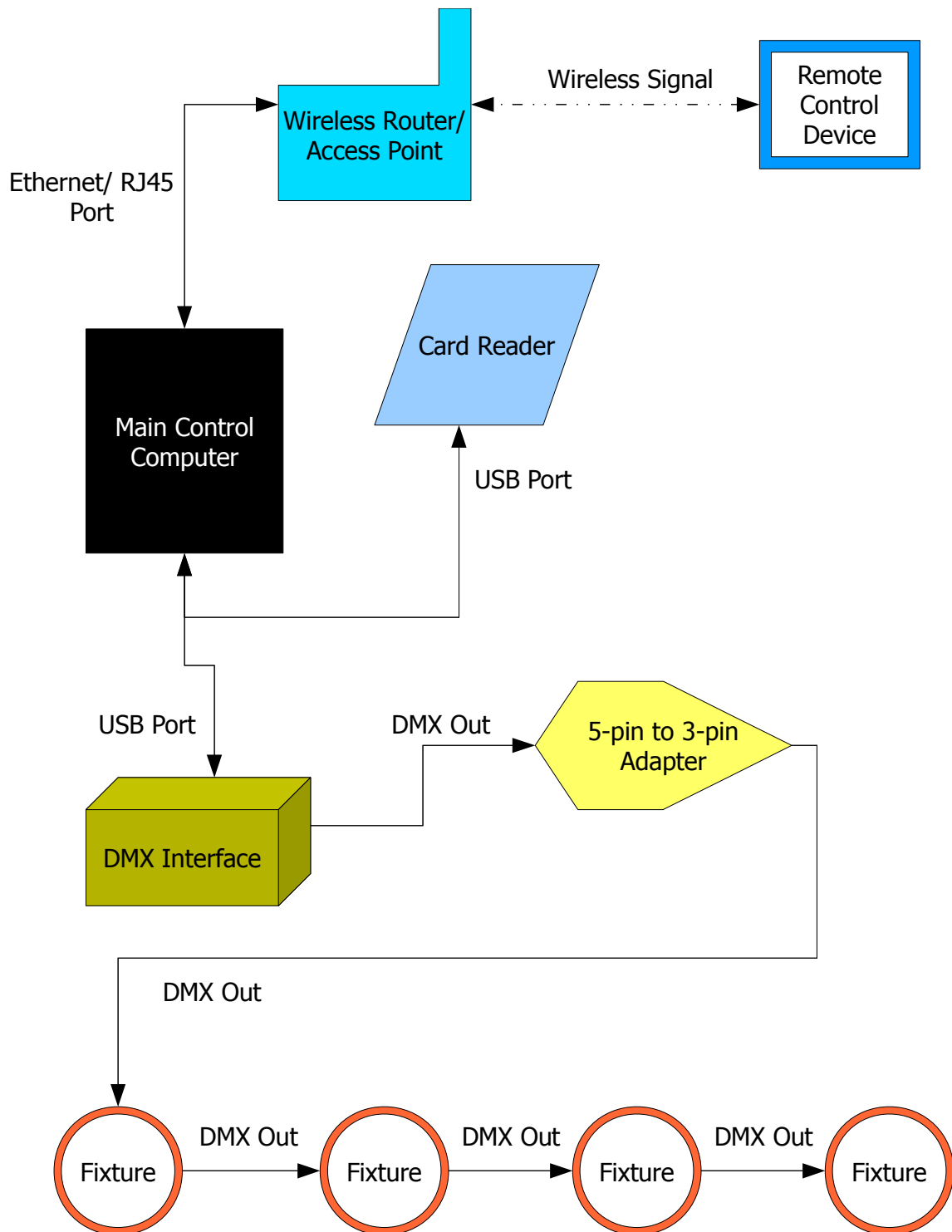
Here's to lots of great, creatively lit shows!

System Reference Diagrams

Main Control Computer Internals



Overall Control System Data/ Signal Paths



Troubleshooting

Problem	Possible Solutions
During the build of the main control computer, the machine will not boot, and the VGA card does not display anything.	<ul style="list-style-type: none">• Check that the VGA card is firmly seated in its slot.• Try installing RAM in the motherboard, if none is already installed.• If RAM is already installed, try removing RAM modules one at a time, in case a faulty module is the problem.• Try disconnecting ATA and SATA cables one at a time, to see if one is faulty.• Try replacing the VGA card with a card that is already known to work properly.• If all else fails, your processor may be faulty.
The main control computer will not allow for the setup of a RAID.	<ul style="list-style-type: none">• You may need to explicitly set the SATA functionality to “RAID” using the BIOS.• The RAID BIOS may require an odd key-combination to be pressed during the boot process.
The main control computer will not boot from the Operating System install media.	<ul style="list-style-type: none">• Your BIOS may not be set up to scan all possible boot media by default. Check the boot settings to ensure that your chosen install media will be scanned.
The Operating System will not install to the RAID.	<ul style="list-style-type: none">• You may need to load RAID drivers during the install process. Consult your operating system's documentation for directions. Some operating systems may require you to load drivers via a 3.5” floppy drive.
Software appears to install correctly, but then fails to run at all when an attempt is made to launch it.	<ul style="list-style-type: none">• Some installers will not check to make sure that the program to be installed can actually run on your CPU architecture. For instance, a program specifically meant for use on a 64-bit CPU may have its files installed to the right locations on a filesystem, but will not actually be executable on a 32-bit system.

Problem	Possible Solutions
The configuration page for the wireless router/ access point is inaccessible.	<ul style="list-style-type: none"> • You may need to configure the main control computer's network settings manually, if the wireless router/ access point does not have a DHCP server available. • You may need to perform a “hard” reset of your wireless router/ access point. Consult your documentation for directions.
The remote control device cannot connect to the wireless network.	<ul style="list-style-type: none"> • The remote control device's wireless configuration may not have security settings that match those of the wireless router/ access point. • The wireless security pass-phrase may be too long for the operating system to handle. (Some wireless routers have a bug which allows for the entry of a pass-phrase which is too long without causing the router to fail.) Try setting a shorter pass-phrase. • The remote control device's wireless adapter may not have had its MAC address added to the router's access control lists.
The remote control device cannot connect to the VNC/ remote access server on the main control computer.	<ul style="list-style-type: none"> • On the main control computer, ensure that any firewall software is set up to allow connections to the VNC/ remote access server. • Ensure that both the server and the client software are set up to use matching security/ encryption protocols. • Try restarting the server software.
The remote control device can connect to the VNC/ remote access server, but cannot actually authenticate.	<ul style="list-style-type: none"> • Ensure that the password for the VNC server is not too long. (In some cases, a password that is too long may not cause an exception in the server configuration software.) Try reducing the password length to 8 characters. • Some VNC/ remote access servers that use the operating system's user accounts for authentication may require permissions to be set for each user. Consult your documentation for instructions. • Try restarting the server software.

Problem	Possible Solutions
<p>The remote control device does not work properly with the DMX interface.</p>	<ul style="list-style-type: none"> • Ensure that you have the latest drivers for the DMX interface. (Some software shipped with interface may not have all the latest fixes applied.) • Ensure that you install the drivers before you connect the interface. If necessary, use the operating system's built-in tools to de-install the interface, and then try again.
<p>The lighting control software does not work properly with the DMX interface.</p>	<ul style="list-style-type: none"> • Some systems may allow you to specify a certain communications port to use with the DMX interface. You may need to use your operating system's tools to discover which port the interface is connected to.
<p>A lighting instrument attached to the DMX interface is not working properly, or at all.</p>	<ul style="list-style-type: none"> • Ensure that the configuration for the lighting instrument in the control software is correct. • Ensure that the light's DMX address has been set correctly. (Be especially careful that it has been set to be “DMX Enabled” if necessary.) • Check that the DMX cable is firmly attached at all connection points. • Try changing to a different DMX cable, to see if the original cable is faulty. • Try changing to a shorter DMX cable. • Try adding a DMX terminator to the end of the DMX chain.