11. Designing the PC

The core of a pin cab is a PC running Windows. You could theoretically build a pin cab around a Mac, an iPad, a Raspberry Pi, or just about any other sort of computer. But for our purposes in this guide, the only real option is a Windows PC, because that's where all of the software runs.

I've observed that most pin cab builders like to start their projects by building the PC and setting up the software, before they've even started thinking about what's needed to build the pinball machine body that'll house it. This is a natural first step for most of us, because most of us know our way around PCs at least a little bit - the way that most people discover the pin cab world in the first place is through the PC pinball simulation community. It's also an attractive place to start because you can see some immediate results, before getting into the more daunting parts of the project.

Off-the-shelf or custom build

The easiest and most obvious way to get a PC is to buy one from a retail PC maker, or even re-use one you already have. But most pin cab builders come from a PC gaming background, so you probably already know enough about PCs to know the benefits of building one yourself rather than buying off-the-shelf. If you've built your own PCs in the past, you know what's involved. If you haven't built one before, you might be surprised at how easy it is. Modern PCs snap together out of components practically like Lego blocks. The hardest part is often the shopping, since there are so many options out there.

The big benefit of building your own PC is that you get to pick exactly what you want for each component. The retail PC makers usually give you a few options for CPU speed, hard disk size, graphics card, and so on, but they're usually pretty limited choices from a small pre-set list. If you build your own, you can choose exactly what you want from the whole universe of available products in each category.

The rest of this chapter proceeds from the assumption that you're going to build a custom PC, because that's what most pin cab builders do. But that's not a must; if you're not comfortable building your own PC, you can definitely build a perfectly good pin cab around a retail PC. If you go that route, I'd suggest you focus on PCs that are specifically designed and marketed for gaming. PC pinball is fundamentally a video game, and it benefits from exactly the same hardware upgrades that mainstream video gamers need. Pay particular attention to the graphics card: that's the hardware element that makes the biggest different for PC pinball performance. You might find the material in this chapter helpful even for picking out a pre-built PC, just for the background knowledge of what to look for and which elements are the most important to pin cab performance.

Performance considerations

I can't give you any hard numbers for performance metrics, since things change too quickly in this business and any benchmarks I quote would be obsolete in a couple of months. I can offer some general advice, though.

The first bit of advice is that you should consider the virtual cab PC to be a gaming PC. That might seem obvious, but my point isn't merely that you're going to use it to play games, but rather that "gaming PC" is a special category of PC. The thing that makes a gaming PC different from a run-of-the-mill home or business PC is upgraded performance, particularly for graphics. Gamers use special disks, special memory, and most of all special graphics cards.

The second bit of advice is that you don't have to take this idea of upgraded performance to its logical extreme. You do need good performance, but you don't need the absolute best performance available. Pinball emulation is demanding, but it's not as complex as the latest "triple A" video games at any given time. My rule of thumb is that you should look for the "second best" in most of the product categories. Survey what's available, and don't buy the most expensive thing you can find; focus your attention on the second price tier. Products in that second tier are usually only slightly less capable than the top-tier products, but much cheaper - you often see crazy things like 90% of the performance for half the price. The gamers

who want the *very best* are willing to pay, pay, pay for it. So products in that second tier often offer a much better balance between price and performance.

To a first approximation, the CPU and GPU together determine your machine's overall performance. And of the two, the graphics card is generally the more important. These are the parts you should pay the most attention to when researching what to buy.

Other components - motherboard, memory (RAM), disks, USB controllers - also contribute to performance, but to a much lesser degree. How much should you worry about those? If you talk to serious video gamers, they'll tell you that every element is critical, down to the military-grade titanium screws holding their ballistic carbon-fiber cases together. That's true as far as it goes, but "extreme gaming RAM" and the like will only contribute a few percentage on most systems. Most people can't perceive that kind of difference in actual use; you'd only know it's there if you measured it with benchmarking tools. If it's important to you to build the fastest system possible, then by all means do so; that can be fun in its own way. If your main focus is pinball rather than PC benchmarks, I'd focus my research time and cash budget on the CPU and GPU, and I wouldn't go too far out of my way seeking the optimal choices for the other components. Just look for parts from reputable manufacturers that fit the specs you need.

Operating System

Recommended: Windows 10, 64-bit, Home edition.

Windows is really the only viable operating system option for a pin cab PC, because all of the popular pinball software runs only on Windows.

Which version

I'd recommend the latest, currently Windows 10. The main reason is that Microsoft only offers full updates to their DirectX technologies (their gaming technlogy layer) on the current OS version at any given time. Windows 8 is already somewhat behind on DirectX support, and Windows 7 is very behind. This means that newer games will increasingly be unable to run on the older Windows versions; if you want to be able to run the latest games, you pretty much have to have the latest Windows.

Older versions: As of this writing, Windows 7 is at end-of-life, meaning Microsoft will no longer offer updates for it. As I'm sure you've heard from many other people, the big concern when Microsoft stops updating Windows 7 is that security bugs in the OS won't get fixed, so it will become increasingly vulnerable to malware. For a pin cab PC, I think the lack of DirectX updates is also a big issue. Windows 8 will continue to be updated (according to Microsoft) until the beginning of 2023, but it also is already lacking in some newer DirectX features.

Which edition?

The "Home" edition of Windows is fine for a pin cab. You can buy the more expensive "Pro" edition if you prefer, but the added features in the Pro editions are intended more for business users than individual users. I don't think the extra cost gets you anything you really need for a pin cab.

32-bit or 64-bit?

Easy: Use the 64-bit edition. The 32-bit version of Windows is only for old hardware with CPUs from about 2002 or earlier. Every Intel and AMD PC CPU you can buy today is 64-bit. The 64-bit version of the operating system takes full advantage of the CPU's capabilities, and is still fully compatible with 32-bit application software.

Emulation and virtualization options

Nope. Don't even consider it. Even though it's technically possible to run Windows as a guest operating system using VM software on Linux and MacOS, it's not a viable option for gaming software. 3D gaming performance on virtualized hardware is uniformly unacceptable. The same applies to Wine (a Windows API emulator on Linux).

Hardware components

Here are the PC components you need to assemble the computer that runs a virtual pin cab.

CPU

Most people start planning PC builds with the CPU, because other choices hinge on which CPU you choose.

The most common recommendation for CPUs is a four-core Intel i5 chip. Note that the i5 line contains two-core as well as four-core versions, so don't assume that it has four cores just because it has an i5 label. Check the specs on the specific product you're looking at.

AMD makes Intel-compatible four-core CPUs with performance levels similar to the Intel chips. My own experience is mostly with Intel CPUs, but other virtual pin cab builders have successfully based their systems on AMD chips.

Intel and AMD also make CPUs with more than four cores, such as the Intel i7 and i9 chips. As you'd expect, these are faster than the four-core chips on generic performance tests. However, this can be misleading. It might seem intuitive that eight cores would be twice as good as four, but things aren't nearly that simple. Some applications benefit from more cores, others don't. Most virtual pinball software will see diminishing returns above four cores. If you have the budget to upgrade beyond four cores, you're likely to see a much bigger performance gain by putting the money into a higher-end graphics card than into the CPU.

Even after narrowing things down to, say, the i5 line, you'll still have several chips to choose from. The main variation will be clock speed. Higher clock speeds generally yield faster performance, but as with the number of cores, the correlation isn't always straightforward. I always consider clock speed to be a good place to apply the "second best" rule that I mentioned under Performance Considerations above.

If you want to do more thorough research on the current CPUs available, there are numerous Web sites with detailed performance tests. You should give the most weight to tests for gaming performance, since pinball simulators are similar to other video games in the way they use the machine's hardware resources.

CPU fan

Most modern CPUs require a special fan mounted directly on top of the chip. If you buy your CPU in retail packaging, it usually includes a suitable fan. However, some vendors sell unpackaged "OEM" versions intended for use by business buyers building systems for resale, and these usually don't include anything but the bare CPU. In that case, you'll need to buy a CPU fan separately. These can be found on Newegg and other sites that sell components by using a search term like "i5 fan". Check the specs on the options you find to make sure your specific CPU type is listed, since these fans usually have to match the exact shape and size of the chip.

Motherboard

The motherboard is the main system board with all of the core electronics, and connectors for all of the add-in cards, disks, and input devices.

Choose a CPU before looking for motherboards. Any given motherboard only works with specific CPUs. Once you know the CPU you're going to use, you should be able to find suitable motherboards by searching the Web for "xxx motherboard", where "xxx" is your CPU type. Use the detailed CPU part number, like "i5-7600k".

I've had good results with motherboards from Gigabyte, but several other manufacturers make good motherboards as well.

For a pin cab, your needs from a motherboard aren't very complex. Here are the main features I'd look for:

- Must have: Compatibility with your chosen CPU
- Must have: At least one fast expansion slot for a graphics card, typically PCI Express x16 (as of this writing).
- Must have: At least two additional expansion slots, in case you want to add a sound card, Wi-Fi card, USB card, or any other add-ins.
- Must have: Memory slots for at least 8GB of RAM. (This is almost a given; it's hard to find a board *without* at least this much capacity these days.)

- Nice to have: on-board Ethernet port. This is standard on nearly all modern motherboards. Wi-Fi is less important, because you might not be able to use a built-in antenna effectively; the walls of a pin cab are thick enough to significantly block the signal. An external antenna is usually better if you want Wi-Fi on the cab, and for that you'll probably need an add-in card or an external USB Wi-Fi adapter.
- Nice to have: integrated audio. Nearly all modern motherboards include audio outputs. This isn't required, though, as you can add a sound card via an expansion slot if needed.
- Nice to have: USB 2 **and** USB 3 connectors. Some older USB devices don't work well with USB 3 ports, so it's helpful to have both types in case you need a USB 2 port for some devices. This isn't required, though, since an external USB 2 hub can serve the same function.

Performance considerations: Not really an issue, unless you're looking to build an extreme gaming system. A motherboard designed for a particular CPU is almost always based on the Intel or AMD chipset mated to that CPU, so you won't see a huge amount of variation among different boards for the same CPU. If you're concerned about finding the fastest motherboard for your CPU, you can do some research on benchmark sites on the Web.

What about on-board graphics? Unimportant, because you'll need a separate graphics card whether or not the motherboard has its own built-in graphics. There might be exceptions, but all of the built-in motherboard graphics chip sets I've ever seen are low-end, suitable for business graphics, not gaming.

If the motherboard doesn't have on-board graphics, great, that's one less thing to worry about when configuring the BIOS. If it does have on-board graphics, as most modern motherboards do, it's still not a problem because you should be able to disable it in the BIOS setup. In fact, many BIOSes will do this automatically when they detect the presence of a separate video card.

Graphics cards

The graphics card is the most important component for game performance. It's even more important than the CPU for games, because it's actually a whole separate computer in its own right that does most of the computing work for displaying 3D graphics. Fast graphics cards are capable of drawing more complex images more rapidly, making for smoother game action.

You should wait until after selecting a motherboard to choose a graphics card, because you need a graphics card that matches the "slot" type on your motherboard. Your motherboard specs should tell you what kind of graphics cards it accepts; look for "graphics cards" or "expansion slots" in the spec sheet. For quite a while now, motherboards have been standardized on "PCI Express" slots for the graphic interface. These are quoted with a speed like "x16", so you might see "PCI Express x16" in the expansion slot list. Once you find that information, that tells you what types of graphic cards are compatible.

Graphics cards are available from many manufacturers, but most (regardless of manufacturer) use chip sets made by either Nvidia or AMD. The spec sheet should tell you the underlying chip set, and in fact, most cards from most brands include this information right in the name. For example, a "Gigabyte Geforce GTX 1050" is based on the Nvidia 1050 chip set. You'll start to recognize the chip set names if you shop around enough, since you'll see the same numerical designations over and over on different brands of cards. The performance of a graphics card is almost entirely a function of the chip set, not the brand, so you should see reasonably similar performance from cards based on a given chip set even if they're from different brands.

Which chip set? The standard answer on the forums, for Visual Pinball use, is "Nvidia x60 or better". That applies if you're using a 1080p TV. For a 4K TV, the standard advice is that an x70 is required.

What these numbers mean: The "60" part refers to the speed class of the cards. NVidia names their chip sets by combining a generation number (roughly analogous to a model year) and a processor speed level. For example, the "960" chip is generation 9, speed class 60. So you might see 960, 1060, and eventually 1160 chip sets, representing successive model years of the "x60" speed class. Higher speed class numbers represent faster chips: "60" is faster than "50", and "70" is faster still. The generation numbers increase about once a year. The cards in a given speed class

get a little faster each generation, so a 1060 is probably faster than a 960, but the speed class differences are bigger and typically hold across generations. That is, an older "60" is probably faster than a newer "50". So don't assume that newer is automatically faster; the speed class is the number to look at first.

Gaming cards with AMD chip sets will also work with Visual Pinball and other gaming software. But I don't think there's a standard answer for which AMD chip set to use, the way there is with NVidia, because pin cab builders have mostly gravitated to NVidia cards. I think the main reason they gravitate to NVidia is that the pin cab builders before them gravitated to NVidia, so those are the cards that people know will work. If you want to cast a wider net, there are lots of video-gamer Web sites that do seriously in-depth performance testing on video cards (as well as other PC components), so a little research should give you some idea of which AMD-based cards are comparable to the recommended NVidia cards.

Video memory: Video cards have their own on-board memory, usually 1GB or more on a modern card. The fastest type of memory has a type like "GDDR3" or "GDDR5". A higher number suffix indicates faster memory. Visual Pinball and other gaming software benefits from large memory sizes with fast memory. I'd recommend at least 2GB of GDDR3 or faster.

Display size and refresh rate: If you're using a standard HDTV for your playfield TV, it probably uses the 1080p format, which is 1920x1080 pixels and refresh at 60 Hz. Your backglass TV might also be 1080p, but some smaller TVs run at 720p, which is 1280x720 pixels. Any modern graphics card will be able to drive these image formats, but check the specs to be sure.

If your playfield TV is a "4K" model (Ultra HD or UHD), the image size is 3840x2160 pixels. Support for this format isn't a given in modern graphics cards, so you might have to look a little harder to find a suitable card. In addition, the higher resolution places much more computing load on the graphics card, so the earlier advice about Nvidia x60 cards probably doesn't apply. You'll probably need a very high-end card to get good performance; I'd recommend asking for advice on the forums.

Outputs/connectors: Be sure you have enough outputs for all of the monitors you plan to connect, taking into account the playfield TV, the backbox TV, and the score display (DMD) TV, if you're using that.

Most higher-end graphics cards offer several output ports with different types of connectors. You can almost always use all of the outputs simultaneously to drive multiple monitors. This lets you use a single graphics card to drive all of the TVs in your system.

I'd recommend finding a card with at least two of the following connectors, in any combination: HDMI, DVI-D, Display Port (DP). All of these types can be connected (using passive adapters) to HDMI inputs, which is what you'll need on almost any modern TV. As long as you have two ports of these types, you should have no problem connecting two TVs to the card.

If you're planning to also use a third display for the DMD area, you'll need a third output for that. A VGA or DVI-D connector will usually work for this third output, since DMD monitors are usually implemented with laptop displays or small desktop monitors. Most video cards have a VGA output in addition to one or more of the more modern connectors listed above, so this is fairly easy to find.

If you're going to use a real pinball DMD instead of a small video display, you **won't** need to connect that the graphics card. Real DMDs aren't video devices, so they don't connect to your graphics card; they connect instead to a special external controller via USB.

You should check the specs to confirm that the card you're considering can handle the two or three simultaneous outputs you plan to use. Nearly all modern graphics cards allow this, but it's worth checking to be sure.

Port compatibility: You don't necessarily need an exact match between the output port types on your video card and the input ports on your TVs and monitors. Many of the port types are electrically compatible with each other, meaning you can connect them with a simple cable that has the right plug on each end.

The following combinations of port types are compatible. The only requirement is a cable with the corresponding connector type at each end. These are relatively inexpensive and can be easily found online.

TV IN	Video Card OUT	Compatible?
HDMI	HDMI	Yes
	DVI-D	Yes
	DisplayPort	Yes
DVI-D	HDMI	Yes
	DVI-D	Yes
	DisplayPort	Yes
DisplayPort	DisplayPort	Yes
VGA	VGA	Yes
	DVI-I	Yes

Two cards for two monitors: Not advised. It *seems* like two cards would be better than one - more hardware is always faster, right? But in practice, two cards are actually *slower* than one. Everyone on the forums who's tried this has had the same results: you get lower frame rates, more stutter, and more lag with multiple video cards.

The technical reasons for this are unclear (my wild guess is that it's due to increased PCIe bus contention). Without understanding the cause, I can't rule out the possibility that *some* systems exist where two cards would go faster than one. But if there are, they seem to be the exception; many people have tried it and had poor results.

By far the best way that anyone has found to improve performance of the pinball simulators is to use a faster video card.

Using the motherboard GPU as a second video card: Not advised, for exactly the same reasons that you shouldn't add a second video card (above). Enabling the motherboard GPU is exactly the same as adding a second video card in terms of its effect on your overall system performance: you'll see lower frame rates, more stutter, and more lag if you enable the on-board GPU.

Memory (RAM)

I'd recommend at least 8GB of motherboard RAM. This is enough memory for Windows plus the pinball simulator to run comfortably without "swapping" to disk. More RAM is generally better - particularly for future-proofing, considering that Windows and other software tends to need more memory on every update. If you have the budget, you can install as much memory as your motherboard can accept.

The type of RAM chip you use must match the requirements for your motherboard. You can find the RAM type requirements in your motherboard's spec sheet, but it's usually easier to find the right chips by typing your motherboard's model number into a Web store's RAM search. Most online stores that sell RAM let you search for compatible chips by motherboard, narrowing the results to show only compatible products once you enter the motherboard information.

You'll probably be able to find many compatible RAM chips for your motherboard. These will be listed with a speed class like "DDR3-2000" or "DDR4-2133". "DDR3" and "DDR4" are essentially versions of the electrical interfaces, so your motherboard will probably accept exactly one of these types. The suffixes like "-2000" are clock speeds, so higher numbers are faster in terms of the bus clock. These numbers don't translate directly or linearly to overall system throughput, since there are many other factors besides the raw clock speed that affect the actual performance, but using higher-speed RAM will generally increase overall system speed. I'd recommend buying the fastest speed class that your motherboard supports, since the price differences between RAM types aren't usually dramatic, but you can let your budget decide, since the performance differences probably won't be dramatic either.

Note that you might see the "DDR" speed class combined with another class with a "PC" prefix, such as "PC3-16000". These are just different ways of stating the same information. Don't compare "DDR" speeds with "PC" speeds, since they're different

systems - only compare "DDR" speeds with other "DDR" speeds, and "PC" speeds with other "PC" speeds.

In addition to the "DDR" speed class, you might see a series of other specs, such as "Timing 15-17-17-35" or "CAS Latency 15". These numbers are further details about the memory speed. Hardcore gamers try to optimize these, but I don't recommend worrying about them, because they represent very slight differences in speed that might not even be noticeable in actual use. The "DDR" speed class and the total amount of RAM are much more important.

Hard Disk

The best type of hard disk for a virtual pin cab PC is an SSD, which isn't actually a "disk" at all, but serves the same storage function using flash memory instead of magnetic media. SSDs are much faster than conventional hard disks, especially for booting Windows and loading software. Booting Windows from an SSD typically takes ten or twenty seconds, compared with a minute or more with a conventional hard disk.

SSDs also smaller than convention disks, and they're essentially immune to damage from vibration or shock. The shock resistance is good for a pin cab since you'll want to be able to nudge the machine without worrying about damaging the disk.

The main drawback of SSDs is that they're more expensive than conventional hard disk per gigabyte. Fortunately, a pin cab doesn't need a very large disk, so most pin cab builders will find that a suitably sized SSD is well within a reasonable budget for the PC components.

How much storage do you need? Let's look at what you'll typically need to install on a pin cab PC:

- Windows operating system: about 20GB
- Visual Pinball: about 20MB
- Future Pinball: about 100MB
- VP and FP tables: varies, hundreds of MB
- PinballX (menu system): about 40MB
- PinballX media (table images, videos, etc): varies, hundreds of MB
- Web browser: 1GB
- Other software and utilities: varies, hundreds of MB

We obviously can't come up with an exact number here because the total will depend on how many tables you install. But we can still come up with a pretty good upper bound, since there are only so many tables out there (perhaps 1000 in circulation), and they're not all that big individually (perhaps 1MB to 20MB apiece). Even if you install all of the tables you can find, and even if you never delete the less interesting ones, you're probably talking about less than 20GB of total disk space required for them.

Adding all of this up, we come up with about 45GB. Realistically, you'll want to increase that figure to account for the inherent overhead in the way Windows uses disk space, and to leave some room for temporary files, downloads, etc. So I'd recommend an absolute minimum disk size of about 65GB, and preferably at least 120GB. But given current prices, I'd step up to about 250GB - that size is available for about $100 US as of this writing, which makes it the best value in terms of price per gigabyte. 250GB is plenty of space for all current pin cab needs and leaves lots of space for future expansion. Depending on when you read this, the best value size might be even larger, so shop around to see what's available.

Power supply

Virtually all motherboards and disk drives are compatible with the standard "ATX" type of power supply. The only exceptions are motherboards designed for very small form factor machines, so as long as you're using a standard full-sized motherboard, you should be able to use any ATX power supply from any manufacturer.

ATX power supplies are so standardized that you don't have to worry about the types of plugs it has or the types of voltages it produces. These are all uniform across all ATX power supplies. The only thing that varies is the total power capacity, expressed

as a number of Watts. You'll have to pick a power supply that produces at least the wattage required by your motherboard and other components.

You can determine your wattage requirement by adding up the power figures in the specs for your motherboard and video card. Those are the two components that draw the most power in your system. Be sure to pay attention to the video card, because it might require even more power than your motherboard does.

For example, if your motherboard spec sheet says that it requires 200W, and your video card requires 300W, you'll need a power supply that provides at least 500W. I'd add about 100W to the total you come up with from the motherboard and video specs, to account for the little bit of extra power that will be drawn by the disk and USB devices, so in this example I'd look for a power supply rated for at least 600W. The number you come up with here is just a minimum: you can buy any power supply rated at this number or higher.

Sound cards

Most modern motherboards have integrated audio. For a basic setup, this is all you'll need.

If you want, you can add a separate sound card that plugs into an expansion slot on your motherboard. This will let you set up a second, independent set of speakers on the added card, in addition to the main set of speakers attached to the motherboard audio outputs.

Why would you want two sets of outputs? Visual Pinball has a special feature that lets you take advantage of two audio systems to separate the "music" tracks from the playfield sound effects. Many pin cab builders set things up so that the music plays from the backbox speakers, the same arrangement as in the real machines from the 1990s, and the playfield sound effects play through a separate set of speakers located inside the cabinet under the TV. The playfield effects include the sound of the ball rolling and bumping into things, so it improves the simulation to have these sounds come from the direction of the playfield.

If you do want to install a separate sound card for the playfield effects, you don't need anything fancy. Any inexpensive sound card will do. Just make sure that it uses the same type of expansion slot that you have on your motherboard. For most modern motherboards, these will be standard PCI slots.

Case or caseless

Most pin cab builders house the whole PC inside the cabinet. This makes everything self-contained and adds to the illusion of a real machine. That's not a requirement, though: some cab builders simply put a regular PC on the floor next to the cabinet. This works, but it's not as nicely integrated, and you'll have to run several cables (video and USB) between the external PC and the cab. It's fairly obvious how to set that up, so we'll ignore that option and focus on how to set up a PC inside the cabinet.

There are two main options for mounting the PC components inside the cabinet. The first is to build the PC using a conventional tower case, and then put the case inside the cabinet. The second is to skip the case and mount all of the PC components directly inside the cabinet, attaching them to the floor of the cabinet or one of the inside walls. The cabinet itself serves as the case. Each approach has some tradeoffs.

The main advantage of using a conventional PC case is that it provides structural support to hold the video card and other add-in cards in place, and provides places to mount the disks. A case also provides physical protection from falling objects, and provides shielding for the radio frequency energy that a PC produces.

The big downside of using a case is that it takes up a lot of space. A typical mid-tower case is about 14x16x7 inches. A standard pin cab is 20.5" wide on the inside, and you'll have about 9 or 10 inches of vertical clearance between the floor and the playfield TV. That leaves enough room for a tower case lying on its side, but just barely.

Note that there's such a thing as a "small form factor" case. These take up less space, as the name suggests, but they're not really a good option for pin cabs. The big problem is that they don't accommodate full-size PCIe video cards. The video card is so critical to performance that you won't want to be limited to the few available small form factor options.

If you skip the case, it's straightforward to mount the motherboard, disk, and power supply to the floor of the cabinet. The only real complication is that the video card and other add-in cards will need some kind of structural support to keep them from working loose from the motherboard slots. One option is a partial case, known as an "I/O panel" or "I/O tray". You can search for these on the Web by looking for terms like "ATX I/O panel" or "mATX motherboard tray" (use "ATX" or "mATAX" according to your motherboard's "form factor" spec). Another option is to fashion your own ad hoc support from wood or sheet metal. We'll look at specifics in Chapter 27, Installing the PC.

Fans

PC components and TVs generate heat when running, so you'll want to make sure the cabinet interior is well ventilated. Most cab builders do this by installing a couple of PC case fans in specially cut openings in the floor or back wall of the cabinet. See Chapter 28, Cooling Fans for more.

Network

You'll definitely want network connectivity in your cab PC, so that you can download software and pinball tables from the Internet.

Nearly all motherboards have built-in Ethernet ports for wired connections. If you'll have access to a wired router port in your pin cab's ultimate location, the built-in Ethernet port is all you'll need. If not, you'll want to consider another option, such as WiFi or powerline networking.

If you're already using WiFi for your other devices, you can get your pin cab on the network by adding a WiFi card. Some motherboards have built-in WiFi, so you might not even need to add anything; check your motherboard specs.

If you do add WiFi to your pin cab PC, I'd recommend doing so with a PCI add-in card that has an external antenna with at least a few feet of wire, to allow locating the antenna away from the motherboard. The reason is that the wood walls of a pin cab can substantially block the WiFi radio signal, so you'll get a much better signal if you can move the antenna outside of the cabinet. A built-in antenna or an antenna that's attached directly to the PCI card might not get a strong enough signal.

Another option is a "powerline" network. These send signals over your house's AC electrical wiring rather than by radio, so they're not susceptible to the interference and blockage problems that make WiFi problematic in some setups. They also don't require any extra wiring, since they use the existing power wiring in your house. To make this work, you'll need one powerline adapter connected to the pin cab PC via the motherboard's Ethernet port, and a second powerline adapter connected to a port on your Ethernet router. Netgear, Linksys, and others make starter kits that come with the necessary equipment to set this up.

I always prefer a wired Ethernet connection when possible, since it's extremely reliable and almost effortless to set up. Powerline is my second choice when a wired Ethernet connection isn't possible, since it tends to be more reliable than WiFi and easier to set up. WiFi is great for mobile devices, but a pin cab has to be plugged into the power outlet anyway, so I think powerline is the way to go if you can't arrange a regular wired Ethernet connection.

Port connections

Assuming you're placing your PC inside the cabinet, you'll need a way to connect the keyboard, mouse, and (if you're using one) the Ethernet cable.

One easy way to deal with the keyboard and mouse is to buy wireless devices. Modern wireless keyboards and mice come with USB transceivers; just plug the transceivers into USB ports on the motherboard and you're set. Similarly, using WiFi or powerline Ethernet (see the Network section above) eliminates the need for external network cabling. Making everything wireless is the most convenient approach, but it's more expensive, and I've never been fully satisfied with the performance of any wireless keyboard or mouse I've used.

If you're using wired devices, a simple solution is to drill a hole in the cabinet big enough for the cables (preferably somewhere inconspicuous, like the floor or back wall), pull the cables through the opening, and plug them into the appropriate motherboard ports. The downsides of this approach are that it uses up a couple of feet of cable inside the cab (which might put your keyboard and mouse on too short

a leash), and that it's inconvenient to disconnect and reconnect the devices (you have to open up the machine to get to the plugs).

If you want something a little more elegant and flexible, you can install the appropriate port connectors on the exterior of your cabinet and wire them to the motherboard internally. You can set this up pretty easily with parts made for installing data jacks in wall plates that resemble regular electrical outlet plates. These are commonly used for home theater and office installations. Here are the parts I'd recommend:

- 1 Keystone wall plate insert with 2 openings, for the keyboard and mouse
- 1 Keystone wall plate insert with 1 opening, for the Ethernet port
- 2 Keystone snap-in USB 3.0 female-to-female couplers
- 1 Keystone PS2 (6-pin mini DIN) female-to-female coupler (optional, if you're using an older keyboard with a PS2 connector instead of USB)
- 1 Keystone snap-in RJ45 Cat6 female-to-female coupler
- 4-foot standard male-to-male cables for each of the above connections

See Chapter 27, Installing the PC for installation instructions.

Optical disks

Most pin can builders don't include any optical disks (CD-ROM or DVD-ROM) in their systems. And it almost goes without saying that floppy disks and other removable media are obsolete and can be skipped.

Apart from the operating system, you should be able to load all necessary software by network download. The operating system itself can be installed from a USB thumb drive. Newer versions of Windows can be purchased on a pre-loaded thumb drive, and you can also use your existing desktop PC to create an installable thumb drive image from Windows DVD-ROM install media.

12. Power Switching

When you push the ON button your virtual cab, you want everything to turn on automatically: all the TVs, the audio system, the feedback devices, etc. Likewise, when you're done playing, you don't want to run around shutting everything off separately; you just want to press the OFF button and have the whole thing shut down.

There are some challenges to achieving this kind of power integration, but they can all be overcome with a little planning and setup work. This chapter covers what you need to know to achieve single-button on/off control.

Soft power control through the computer

The key to whole-system power control is to let the computer control the power. Modern PCs are designed for "soft power" control, which means that the operating system software controls the power to the motherboard. This is how Windows shuts off power when you select "Shut Down" from the Start menu.

Using the soft power control on the PC motherboard itself is easy. You just need to wire a pushbutton to the power control pins on your motherboard. These pins are usually part of the "Front Panel" or "F_PANEL" header. If you were going to install the motherboard in a regular desktop case, the case would have a 10-pin connector that you'd plug into this header on the motherboard.

The exact location of the front panel header varies by motherboard, but it's usually easy to find. Look for a 10-pin connector near one of the edges of the motherboard. It's usually labeled F_PANEL or FRONT PANEL.

Intel defined a standard arrangement for this header a long time ago, so almost all motherboards use the same setup. You should check your motherboard's documentation to be sure, but the power switch pins are almost always the **red pins** in the diagram above - pins number 6 and 8 in the standard numbering.

To power up the PC via the soft power control, all you have to do is connect those two red pins together for a moment. To turn the power off through Windows, just momentarily connect them again.

To connect a pushbutton to the soft power controls, you simply connect one terminal of the pushbutton to one of the red pins, and the other terminal to the other red pin. It doesn't matter which order they're connected in.

The standard place on a virtual cab to mount the power button is on the bottom of the cabinet, at the lower right corner. See the "Floor" section in Chapter 21, Cabinet Body for the standard location in the WPC cabinet plans.

Type of connector use with F_PANEL

The standard F_PANEL connector uses two rows of pins with 0.1" pin spacing. This is a standard type of connector. To learn more about how to build a plug to connect here, see "0.1" pin headers" in Chapter 80, Connectors. I'd recommend building a connector using a 0.1" crimp pin housing, as described in that section.

Controlling everything else through the PC

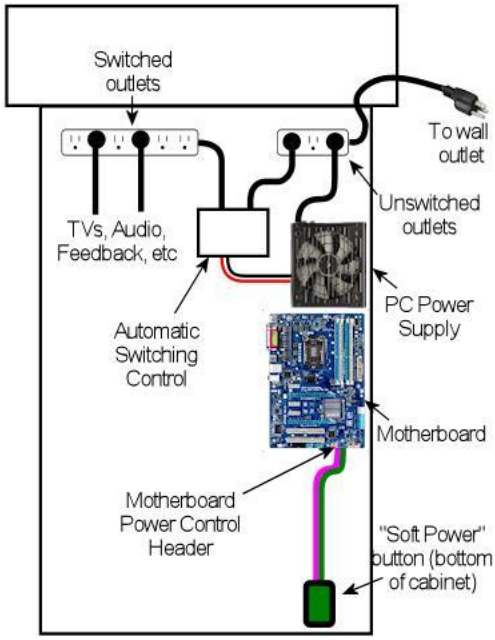
If you have a desktop PC, you've probably noticed that your monitor shuts itself off whenever you power down the computer, and turns itself back on when you boot up the computer. You don't have to worry about switching your monitor on and off separately; it just follows the PC's lead.

That's the template for how the cab should work. We want the TVs and everything else in the cab to follow the PC's lead when turning on or off.

Unfortunately, we can't count on the other devices in a pin cab to switch themselves on and off with the PC the way a computer monitor does. Computer monitors only do this because they're specifically designed for it. They work by way of the video signal. When there's no incoming video signal, a monitor assumes that the computer is turned off, so it goes into standby mode. Regular TVs usually don't have this feature, nor do audio amplifiers or the miscellaneous power supplies we use in a pin cab for feedback devices. So we have to give them some help, by adding our own power control machinery that we design to mirror the PC's power state.

There are two main options for implementing this, but they both work the same way: they control power to the **AC outlets** for the TVs and other devices, according to whether the PC is on or off.

The basic idea is that we install a power strip inside our pin cab, with a set of outlets for plugging in the AC power cords for the various devices. But we have two kinds of outlets in the power strip: one special outlet for the PC, and a bunch of outlets for everything else. The special outlet for the PC is always powered - that is, you effectively leave the PC plugged into the wall outlet all the time. The other outlets are **switched**, meaning that they only receive power when the switch is on. When the switch is off, they don't receive any power at all, so it's exactly like completely unplugging whatever's plugged into them. We use those switched outlets for the TVs, audio system, and all feedback devices.



Conceptual outline of how the PC can control power to other devices. The PC is plugged into an outlet that always receives AC power, allowing it to be switched on and off through the PC's "soft power" button. The TVs, audio system, and feedback devices are plugged into a second set of outlets that are controlled by an automatic switching control. The automatic control cuts AC power to the secondary outlets whenever the PC is off, forcing all of the other devices to turn off.

The final piece of the puzzle is how the outlet switch is controlled. It's not a "switch" in the sense of a wall switch that you operate manually. Rather, it's an electronically controlled switch, controlled by the PC's power state. When the PC is on, the switch turns on automatically. When the PC is off, the switch turns off automatically.

Since everything except the PC is plugged into switched outlets, all of those devices have no choice but to follow the PC's lead. When the PC is off, the switch cuts their power off at the source, so they have to turn off immediately whether they want to or not. They don't need any in-built circuitry to know whether the PC is on or off; we override all of that and control their power at the source.

Let's look at the two main ways we can implement this "switched outlet" setup.

Option 1: Smart power strip

You can buy pre-packaged "smart power strips" from Amazon or Best Buy that implement the sort of behavior we've been describing. This is supposed to be the easy way to implement a switched outlet, since it's plug-and-play. Just plug your computer into the designated outlet and the strip does its magic.

The reason I hedged by saying it's *supposed to be* the easy way is that it doesn't always turn out to be that easy. Some people run into snags with it, which we'll come to in a moment.

If you want to buy a smart power strip, try searching online stores for "smart power strip" and "green power strip". ("Green" because they save energy by cutting power to idle devices.) The specific product I used in my cab is an APC P7GB, which works well for me.

A retail smart power strip will have a specially designated "master" outlet or "computer" outlet, which is where you plug in the PC. The smart switching feature works by monitoring this special outlet to see if any power is flowing through it. When the sensor detects power flowing through the master outlet, the strip figures that the PC is on, so it turns on power to the other outlets. When the master outlet isn't drawing any power, the strip assumes that the PC is off, so it cuts power to the other outlets.

The good thing about this design is that doesn't require any special cooperation with the PC. It doesn't need any special connections to the PC or any special software. It works purely by monitoring the PC's power usage through its main power plug.

The weakness of the design is that nearly all PCs draw a little bit of power even when they're off, so the sensor in the smart strip that detects power usage in the master outlet has to be calibrated to allow for that. The smart strip can't just wait for the power level to drop to absolute zero, because that never actually happens. To make matters worse, there's no "standard" idle power level; every PC is a little different, and it can even depend upon what's connected to the computer, since some USB devices draw power through the PC even when the PC is off. The maker of a smart strip doesn't know what kind of PC you're going to use with it, so they can't tailor the threshold level to your particular model. They just pick a level based on averages across many models. This *usually* works, but not always. Some PCs are so energy-efficient that they always stay below the threshold levels, so a smart strip might never detect that those PCs are on. Other PCs draw enough power when turned off that they remain above the threshold levels at all times, so a strip might never detect that those PCs are off.

Every strip has its own power threshold level, and every PC has its own power characteristics, so it's not easy to predict if a given strip will work with a given PC. The only way I know to find out is to test the specific pairing. So if you're going to test a smart strip, buy it from a store with a good return policy, in case it doesn't work with your computer.

Another disadvantage of the packaged smart strips is that they usually have a mix of switched and non-switched outlets, which means that they don't have very many switched outlets. For example, my APC P7GB only has three switched outlets out of 7 total, which isn't enough for all of the things I need to plug into switched outlets. That's easily solved by plugging a regular dumb power strip into one of the switched outlets to create more switched outlets, but that takes up extra space in the cab, so it would be tidier if the smart strip had more switched outlets built-in.

Option 2: DIY switched outlets

Note: there's a retail product called the IoT Power Relay that's almost exactly like the DIY solution we're about to describe, but it comes pre-built, saving you the work of finding the component parts and assembling them. You might also prefer it for safety reasons, if you're uncomfortable working with high-voltage wiring. See Option 3 below for more details.

A second way to implement automatic power switching is to build it yourself. This is more complex than buying a retail smart power strip, but it's more reliable and more flexible. It eliminates the problem that some smart strips have with properly sensing the on/off status of the computer. If you have any problems getting a smart strip to work with your computer, you can use this approach instead. This approach also

makes it easier to add more switched outlets; the smart strips usually only have three or four switched outlets, which might not be enough for a decked-out pin cab. (With a smart strip, you can always plug a dumb power strip into one of the switched outlets add more switched outlets, but that takes up more space in the cabinet. If you build your own DIY switcher, you can start with a dumb strip that already has enough outlets for your needs.)

You'll need three things to build your own switched outlets:

- A small power strip (the ordinary "dumb" kind) with 2 or 3 outlets, to provide the **unswitched** outlets for the PC and the switched power strip
- A second ordinary power strip, with 6 or so outlets, to provide the **switched** outlets
- A 12VDC relay that can switch large power loads of at least 120VAC and 20A

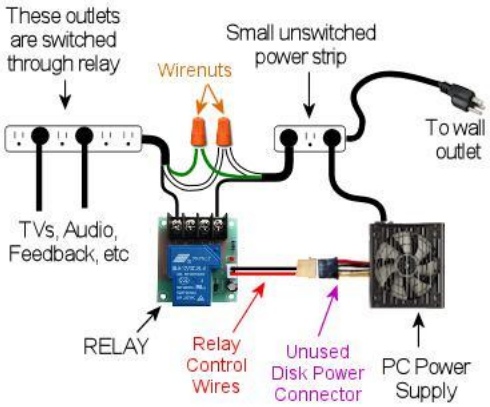
For both power strips, I recommend buying strips equipped with surge suppressors. The primary strip will be running your PC, and the secondary strips will be running your TVs, so both would benefit from surge suppression.

Relays that switch large loads are also known as **contactors**. You can find suitable devices on eBay, built into little circuit boards that simplify the wiring. Here's a picture of what to look for:

Search on eBay for "12V contactor board". You should be able to find listings that look similar to the picture above. (You don't need to find an exact match - the picture is just to give you an idea of what they look like.) The most common type currently listed has output limits of 250VAC and 30A, which is safely above our minimums. Make sure the control signal is listed as exactly 12VDC.

Be sure that your relay or contactor has a diode installed across the coil. This is important because it protects the 12V power supply and your PC electronics from the voltage spikes caused by the relay's magnetic coil. If you use an eBay contactor board, it'll probably have such a diode pre-installed, but you should visually inspect the board to make sure. If you're using a plain relay or contactor you bought as a separate component, you'll have to install a diode yourself. See Chapter 53, Coil Diodes for wiring instructions.

Here's the basic wiring diagram:



The theory of operation is simple. When the PC is ON, the PC power supply sends power to the disk connectors. This provides 12V to the relay, which turns the relay on, which in turn connects AC power to the switched power strip. When the PC is in one of the "soft off" modes, the PC power supply turns off power to the disk connectors, which cuts the 12V power to the relay. This switches the relay off, which cuts AC power to the switched power strip. This is equivalent to unplugging everything connected to the switched power strip, so all of the TVs and other devices will turn off.

Most of the connections shown are just a matter of plugging in power cords: plug the PC power supply into the unswitched outlet, plug the TVs and other devices into the switched outlet. But there are three DIY steps required:

Step one: Find an unused disk connector on your PC power supply. Connect the **control wires** from the relay board to the disk connector wires as follows:

- **Red** relay wire → **Yellow** power supply wire
- **Black** relay wire → **Black** power supply wire

See How to connect 5V and 12V devices in the Power Supplies chapter for instrutcions on how to connect wires to the disk connector plug.

Step two: Make absolutely sure everything is unplugged for this step, because we have to cut into the AC power wiring.

On this step, we're going to cut the power cord in half for your **second** power strip: the one with 6+ outlets that's going to become the switched power strip. You don't have to cut it *exactly* in half, though; you should cut it where it will be most convenient for your physical layout. To figure out where that is, you should take a moment to do a rough fit in your cabinet to determine where you're going to situate the two power strips and the relay. Look at the diagram above and observe how the power cord from the second strip is going to be split into two parts, with the relay in the middle. Find a good point to cut the power cord so that you'll have a little slack on both sides of the cut when all of this is assembled.

Inside the power cord, you're going to find three internal wires. They should be color coded black, white, and green. The black wire is the one that we're going to connect to the relay. This is the "hot" or "line" wire that carries the voltage, so it's the one we want to interrupt to switch the outlets off.

The white and green wires are going to simply connect directly across both halves of the split power cord. In the diagram, we showed them connected by wire nuts, because we're assuming that you're going to have to cut the cord in half all the way through, severing all three wires inside. If you're really careful, you might be able to save that step by cutting only the black wire in half and leaving the white and green wires intact. If you can manage that, there's no need for the wire nuts. If you do end up having to cut the cord fully in half, though, reconnect the white and green wires by stripping a bit of insulation off the ends (about 1/2" worth), feeding the ends into a wire nut, and twisting them together until their securely in place. Make sure there's no exposed bare wire sticking out of the nut when you're done. As shown in the diagram, connect green to green and white to white - all we're doing here is undoing the cut and restoring the green and white wires to their original condition. You might want to wrap the nuts and some of the surrounding wire in electrician's tape when you're done to secure everything in place.

The black wires connect to the input and output terminals on the relay. It doesn't matter which black wire goes to the input and which goes to the output; either way is equivalent. The relay terminals might be labeled **input** and **output** or **K0** and **K1**. Many of these boards have four terminals; when they do, each pair of terminals is simply connected together. For example, there might be two terminals labeled K0; these are wired together inside the board, so you can just pick one of the two to connect one black wire.

Step three: Secure everything in place and cover the high-voltage wiring for safety.

Once everything is wired, permanently fasten the relay board to the cabinet floor (or wall) with screws. I'd also recommend using standoffs, to leave a little open air under the board. Secure the power strips in place. I'd also secure the cut power cord portions, perhaps with wiring staples, to ensure that the wire nut joints aren't jostled or stressed and that the black wires can't be accidentally pulled out of the relay terminals.

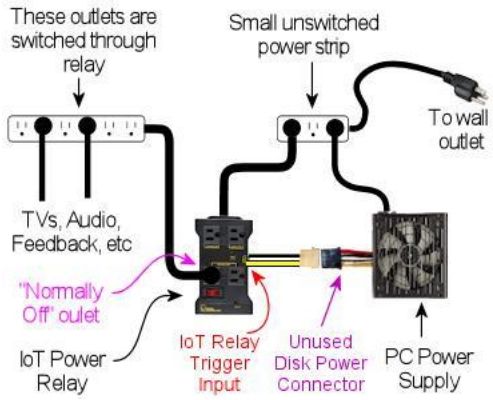
Finally, you'll have to improvise a cover for the entire relay assembly, so that there's absolutely no exposed metal or wire. The black wires will carry AC line voltage, which is hazardous high voltage. You don't want to allow anything loose in the cabinet to come into contact with the AC wiring, and you don't want any risk of touching it yourself while working in the cabinet. Remember that **the AC line voltage will be live on these wires whenever the cabinet is plugged in,** even when the computer is turned off. I'd recommend going to Home Depot and getting a plastic electrical junction box, of the type used inside the wall in your house wiring for switches and outlets. Get a box big enough that the relay board will entirely fit into it. Place it over the relay board and screw it into the cabinet so that the relay is permanently covered.

Option 3: IoT Power Relay

There's a retail product, called the IoT Power Relay, that implements the functionality described in the DIY option above, but without the need for you to buy individual components and assemble them. You can buy these from Amazon and other online retailers; search for **IoT Power Relay**. As of this writing (February 2021) they sell for about $27.

The IoT Power Relay is set up to trigger based on just about any AC or DC voltage, so you can set it up exactly as described above for the DIY option, using the 12V wires (yellow and black) from one of your primary PC power supply's unused disk connectors as the trigger source. Note that the IoT Relay's trigger input is polarized, so you have to connect yellow and black in the correct order. Be sure that the yellow wire from the disk plug connects to the "+" terminal of the IoT Relay trigger input, and the black wire from the disk plug connects to the "-" terminal of the relay trigger input.

Once you have that wired up, just get an ordinary "dumb" power strip, and plug it into one of the IoT Relay's "Normally Off" outlets. The Relay may have outlets marked "Normally Off", "Normally On", and/or "Unswitched", depending on which revision you get. For our purposes, you can ignore everything except the "Normally Off" outlets. Those are the ones that switch ON when the trigger voltage from the main power supply switches on. Note that you don't even need an extra dumb power strip if you only need two switched outlets, since all versions of the IoT Relay have at least two switched outlets built in. For most cabs, though, that's probably not enough - you'll probably have four or five things that you want to plug into the switched power strip (secondary ATX power supply, 24V power supply, DMD or DMD video panel, backglass TV, audio amplifier). The extra power strip is just there to provide those additional outlets.



The TV Power Memory Problem

Now we come to the eternal bane of pin cab builders everywhere: power memory, or more typically, power forgetfulness.

If you've been following along for the first part of this chapter, your cabinet is now set up (or you at least have a plan) so that everything in it will turn on and off automatically with the computer. This happens thanks to our "smart strip", which controls AC power to every outlet (apart from the computer's own outlet) according to computer's power status.

The "power memory problem" in a nutshell is that many TVs won't turn on with this setup. Instead, they'll go into "standby" mode, where they'll stay dark while awaiting an IR remote control command. A TV in standby mode won't show a picture even if it's receiving an active video signal. This is bad for our "smart strip" system, because the smart strip makes the TV *think* it's being plugged in anew each time the PC is powered up. If the TV is designed to go into standby mode each time it's plugged in, the TV will effectively remain off, defeating our wonderful one-button power control.

How to tell if your TV has the problem

The only reliable way to determine if a particular TV has the power memory problem is to test it. If you're still shopping and want to test a TV before you buy it, you really have to find the exact model you're considering in a showroom or friend's house and test that specific TV. Don't count on similar models from the same manufacturer working the same way; it's not consistent across product lines.

The thing that really makes it hard to shop for this feature is that it's almost impossible to find good information about this online. You won't find it listed in a spec sheet or Amazon product page, and most people won't even know what you're talking about it if you ask. Your best bet is to ask on the virtual pin cab forums, because at least some people there will understand the question; even so, there are so many TV models that it's always hard to find someone who owns the exact one you're considering.

If you do have a way to test a model in person (or by proxy), you can get a definitive answer using the following text procedure. Ideally, you should try this using the same video input on the TV that you're going to use when it's installed in your cabinet. For example, if you're going to connect it to your PC by HDMI, run the test with the TV set to view an HDMI video source. The reason this is important is that some TVs have different behavior on this test with different sources.

Here's the test:

- Plug in the TV
- Turn it on
- Let it run for a couple of minutes
- Unplug the TV **without** turning it off first
- Wait a few minutes
- Plug it back in

On that last step, if it turns back on and returns to showing the same video source as before, hooray! The TV has good power state memory. It should just work automatically with a smart strip in a pin cab, so you shouldn't need to pursue any of the solutions below.

If the TV goes into standby mode after being plugged back in, it has the problem. You'll need one of the solutions below if you want to use it in your cab and you want single-button power control to work properly.

Solutions to the TV power-on problem

Fortunately, the power memory problem can be solved. Here are several possible solutions, in order of DIY-ness.

Solution 1: Buy a TV that doesn't have the problem

The easiest solution to this problem is to not have it in the first place. You can simply decide when buying a TV that power memory is a must-have feature, and reject any models that lack it.

My guess is that about 50% of the people in the pin cab forums would agree with that approach, because they really don't want to mess with any of the workarounds. Personally, I **don't** like this approach, because power memory is hardly the most important thing to me about choosing a TV. I think it's much more important to consider picture quality, motion blur, input latency, physical fit for the cabinet, price, and probably a few other features, before worrying about whether it has power memory. You might rule out some otherwise superior candidates if you consider this a deal-breaker. I'd only consider power memory a "nice-to-have" feature, meaning I'd only use it to decide between sets that are otherwise equals. The power memory problem is solvable by the other means we'll see below, so it's really not the end of the world if your TV needs a little help powering on.

Solution 2: Keep the remote handy

Of the 50% of cab builders who *don't* think power memory is the king of all TV features, I'd guess that about 50% of them throw in the towel on single-button power-up if their TVs don't have it. Because there's always the easy manual solution: keep the remote handy and press the On button every time you power up the cabinet.

This really isn't a terrible solution. I'm too much of a perfectionist to accept it for my own cab. It's not the inconvenience of it that's the problem for me; it's just that it makes the project feel a little unfinished. But in practical terms, it costs no significant amount of time and is only a minor inconvenience. If you can live with the rough edge, and the solutions below seem like more trouble than they're worth, you can stop here and call it done.

Solution 3: Tape down the On button

For some TVs, you can get away with a simple hack. It's inelegant (which is, after all, the proper definition of "hack"), and it doesn't work at all on most TVs. But it's worth trying, because if it does happen to work on your TV, it's a really simple solution that you can implement in a matter of minutes.

Here's the idea. On some TVs, if you keep the on/off button pressed down *all the time*, the TV will turn on and stay on whenever you plug it in. If your TV works this way, you can improvise some simple mechanical way of keeping the button pressed down permanently.

Before you start thinking about how to stick the button down, test your TV to see if the trick works for it:

- Unplug the TV.
- Manually hold down the On/Off button.
- Keep holding down the button while you plug in the TV.
- Keep holding it down continuously for a couple of minutes.

Don't let go even briefly on that last step. The point is to test to see if holding the power button down for 10 seconds or 30 seconds or 60 seconds activates some special hidden action, like powering the TV back off, or rebooting it, or bringing up a service menu. "Long press" gestures often do something special like that on modern electronics, since everything these days needs to have a way to reboot it in case of software crashes. 30 seconds is almost always enough for a "long press" to take effect, but I'd give it a couple of minutes just to be sure.

If the TV turned on and stayed on, **and** you didn't activate some special hidden action by holding down the button for a long time, the hack will work.

To implement the hack, you just need to fashion something mechanical to hold down the button permanently. For some models, it's as easy as wrapping some duct tape around the bezel to apply pressure to the button. If that doesn't work for your TV's geometry, try taping a small object (a few pennies, perhaps) between the button and the tape, or try fashioning the right shape out of a paper clip or a little strip of sheet metal. If you have a 3D printer, maybe you can come up with the right shape for a custom plastic clip.

The big limitation of this hack is that it only works for certain TVs. Many TVs will respond by cycling repeatedly between On and Off or activating some special action. That's why you should try the test before worrying about how to implement the hack.

Solution 4: Pinscape TV ON system

If you're using the Pinscape expansion boards, there's a feature built in to help deal with TVs that won't turn on automatically when plugged in. The Pinscape boards have a power sensor that tracks the power supply status, and two mechanisms for sending an ON command to the TV: a relay that can be hard-wired to the TV's On/Off button, and an IR emitter that can be programmed to send the TV's IR remote control command code to turn on. These features can be configured in the Pinscape software to send the TV ON signal (by relay and/or remote) after an adjustable delay interval after the rest of the system powers up, to give the TV a chance to "boot up" and make itself ready to receive commands.

See Chapter 114, TV ON Switch for full details.

Solution 5: eBay timer board

You can build your own equivalent of the Pinscape TV ON feature using a type of electronic timer circuit board available on eBay.

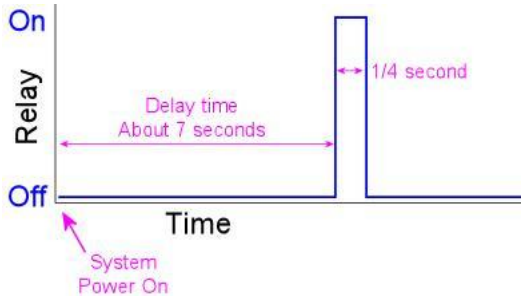
Note: I recommend against using this solution, because it requires taking the TV apart; it's only included here for reference. If possible, use the Pinscape IR transmitter solution instead. See Chapter 114, TV ON Switch. The IR approach is non-invasive and fairly easy to build. You can use it even if you're not using Pinscape for anything else.

This approach works by simulating a manual button press on the TV's On/Off button, shortly after the system power is turned on. We don't physically press the button, but rather simulate it electronically, by soldering wires to the button's switch contacts and connecting them briefly at the proper moment.

There are three important details required to make this work properly:

- We have to sense when the TV power switches from OFF to ON
- We have to wait a few seconds after that, to give the TV time to initialize
- We have to simulate a *momentary* button press only; we can't continuously hold down the button.

To accomplish all of this, we need a timer circuit. The circuit has to be triggered by the power coming on. It then has to pause for a delay time, long enough for the TV to get ready to accept command input, then it has "press the button" for just a moment. Here's what the timing looks like:



We're assuming that the timer is controlling a relay (an electronic switch). The "button press" is simulated by the relay toggling on briefly.

To implement this, we need the timer circuit itself, and then we need to connect it electrically to the TV's On/Off button.

Buying a timer: Suitable boards are available on eBay, but unfortunately it's rather difficult to find the needle in the haystack for this sort of item. The ones you're looking for are no-brand hobbyist products sold by Chinese companies, so there's not a particular store or product name I can point you to. You'll have to sift through the listings to find the right thing, but here's an eBay search term you can use as a starting point: "relay cycle timer".

To find the right timer, first make sure you find something with a relay. Most of the timer boards you'll find do use a relay, but some use solid-state switches (such as MOSFETs) instead. A relay is important for this application. Second, read through the descriptions and look for a list of "modes". The mode you're looking for should be described like this: "when the power turns on, the relay is disconnected, then delay T1, turn on the relay, delay T2, turn off the relay".

When you get the board, you'll have to program it according to the instructions (if any are provided) to set the correct mode and delay times. Set the initial delay time to about 7 seconds, and the second delay time to about 0.25 seconds. You can test that it's configured properly by cycling the power: each time you plug it into power, there should be about a 7 second delay, and the relay should click ON and immediately OFF.

Connecting to the TV: You'll have to be comfortable with taking the TV apart at this stage, because we have to connect some wires to the On/Off button.

There are no generic instructions for taking a TV case apart, so you're on your own for this part. Your goal is to open the case and expose the little circuit board containing the On/Off button.

Needless to say, use extreme caution with this step. In modern LCD TVs, the LCD panel and polarizing filter are very thin, brittle plastic sheets and often have no structural support other than the outer case, so it's very easy to crack them during the removal process or after the case is off. Removing the case will also void the warranty, so you're assuming the entire risk of breaking something by proceeding.

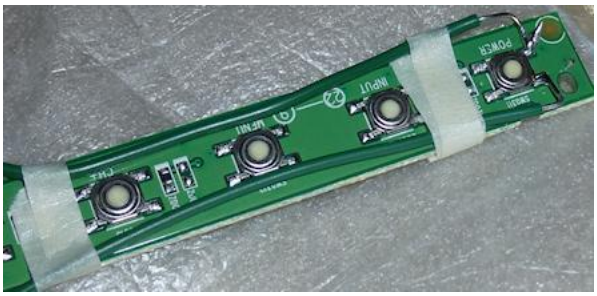
Once you get the case open, you should find a little circuit board located under the area where the buttons on the case are situated. It's usually long and narrow, and looks something like this:

The red arrows in the photo above show the soldering points for the button leads. The little squarish silver objects are the buttons. These are normally situated immediately under the exterior plastic buttons on the TV's bezel; pressing on the exterior plastic button has the effect of pushing down on this metal part, which is the real button.

Once you find this circuit board, identify which button corresponds to the On/Off button on the outer case. Do this by position: just find the inner button that's situated underneath the On/Off button on the case. You can also do this by counting buttons from right to left, since there should be the same number of silver buttons on the circuit board as plastic buttons on the case.

Next, identify the switch leads. There are probably four leads to these switches, one at each corner. On the TVs I've looked at, the leads are in pairs that are electrically connected together, so there are really only two wires here even though it looks like four. Put your multimeter in continuity test mode and check the leads in pairs. Find a pair that are **not** connected normally, but that become connected when you press the button. These are the leads you want to solder to.

The next step is possibly even more delicate and tricky than opening the case. You have to solder wires to the button leads you just identified. To do this, use fine hookup wire, 24 AWG or thinner. Strip a very short length of insulation from the ends, around 1/8". Melt a little solder onto the end of the wire. Position the end of the wire at the desired contact point. Now get out some tape (I used thin strips of masking tape here) and secure the wire to the board a couple of inches away from the contact point. The idea is to hold it in place at the desired position before soldering so that the solder can just flow over the junction with everything already positioned properly. Once everything is in place, heat the end of the wire for a few moments, long enough for the solder to melt and flow onto the switch lead. Remove the soldering iron carefully and try to hold everything very still for a few moments so that the solder can solidify over the junction point. If all went well, the wire should stick to the switch lead. The connection will be delicate at best, so you'll want to secure the wire with a couple more pieces of tape to minimize mechanical stress on it.



TV On/Off switch with wires soldered to leads

Repeat this process for the second lead. Once both are soldered and held securely in place with tape, test your work with the multimeter. Use continuity test again. Connect the meter leads to the free ends of the wires you just soldered. The meter should read open/no connection. Press the button, and the meter should read closed/connected. If that works, you're set. Put the TV case back together, taking care to run your newly attached wires out a suitable opening.

Now you just need to connect the newly attached wires to the timer board relay. Attach the wires to the relay **common (COM)** and **normally open (NO)** terminals on the timer board. (If the relay only has two switch terminals, those are the two to use!)

Finally, to power the relay board itself, connect its DC+ and DC- terminals to the appropriate voltage inputs from the **secondary** ATX power supply. For example, if it requires 5V for power, connect its DC+ input to the red +5V wire on the secondary power supply, and connect its DC- input to the black 0V/Ground wire on the secondary power supply. See Chapter 45, Power Supplies for Feedback for advice on connecting wires to the power supply.

Note that you **must** use a secondary ATX power supply to power the timer board (*not* the main PC power supply), and the secondary power supply must be plugged into the **switched** power strip. That's key to the whole scheme, because the timer board has to be powered up at the same time as the TV in order for the countdown to start at the same time the TV receives power.

Solution 6: DIY timer circuit

This works much like the eBay timer board described above, except that it saves you the trouble of tracking down the right item on eBay. The tradeoff is that you have to assemble your own circuit board instead. But you don't have to design the circuit: you can just build it from my plans.

As with the eBay timer board, I recommend against using this solution, because it requires taking the TV apart. If possible, use the Pinscape IR transmitter solution instead. See Chapter 114, TV ON Switch.

You can download the schematic, in EAGLE and PDF format, along with and an EAGLE printed circuit board layout, here:

mjrnet.org/pinscape/downloads/TVOnTimer.zip

Beta test warning: I haven't built or tested this incarnation of the schematic linked above, which is an EAGLE rendition of the original hand-drawn schematic I used to build the TV ON timer in my own cab. The circuit I built based on the hand-drawn original is well-tested (I've used it for several years without a hitch), but I could have made errors doing the EAGLE translation. I also haven't done a test run of the board design, although my experience has been that EAGLE PCB layouts work fine as long as the schematic is sound. If you're willing to be a beta-tester for these plans, please let me know how it goes!

Before ordering parts, check your TV's timing! If you need different timing, you will need to order different values for parts C8 and/or R10. These parts determine the initial delay time. The delay time can be calculated from these as:

$$1.1 \times R \times C$$

where R is in Ohms and C is in Farads. With the default values as shown in the shematic, the delay is 1.1 × 2.2M × 2.2uF = 5.3 seconds. Before you order parts,

test your TV to determine if it requires a longer delay time:

- Unplug it
- Wait a few minutes
- Plug it in
- Use a timer to wait for 4 seconds
- Press the On button

If the TV turns on, try the test a few more times to make sure the timing is reliable. If so, the default 5 second delay should work. If your TV ignores the first button press on some trials, it probably needs a longer delay time. Try the test again with longer wait times until you find the shortest reliable waiting period. I'd add a second or two to the result as a cushion. Now you can reverse the timing formula above to find new values R10 and/or C8. For example, if you need a delay of 7 seconds, you could keep the resistance value the same and calculate a capacitor value of 2.89uF. Round up to the next common size, which in this case is 3.3uF, which would make the actual wait time about 8 seconds.

Build the board: Assemble the circuit, following the schematic or using the printed circuit board (PCB) design provided in the plans. The circuit is complex enough that I'd recommend building it on the PCB rather than ad hoc. You can have the PCB manufactured by OSH Park for about $12 for three copies of the board, or at any PCB maker of your choice. You'll have two copies of the board left over to give to friends or use on your next cab!

Install the TV wires: The next step is to open your TV and solder wires to its On/Off button. The procedure is described in the section on eBay timers above.

Connect the board: Once you have wires connected to the TV's On/Off button, connect the other ends of the wires to one of the "K1" relay switches, on the Normally Open side. If you're connecting directly to the relay, connect to pins **4 and 8** or pins **9 and 13**. The relay in the spec is double-pole, meaning that it can switch two separate televisions on at the same time. That's why you have your choice of which relay pins to connect. If you have a second TV that needs the same treatment, you can simply connect it to the other pair of pins. If you use the PCB design, connect the TV wires to JP12 pins **1 and 2** (labeled "TV1" on the board silkscreen) or pins **3 and 4** ("TV2").

Connect power to the board: Finally, connect the power inputs to your **secondary** ATX power supply. As with the eBay timer, the scheme is predicated on the timer getting its power through a source that's switched on at the same time as the TV, because the power-on time is the start of the delay timer countdown. If you're building from the schematic, connect VCC to +5V (a red wire) from the ATX supply, and connect GND to ATX ground (a black wire). If you're using the PCB layout, connect an ATX red wire to the JP7 +5V (marked on the silkscreen), and connect a black wire to JP7 GND. See Chapter 45, Power Supplies for Feedback for advice on connecting wires to an ATX power supply.

This circuit design is designed for this single function, so there's no need to "program" it as with the eBay timers. All you have to do is plug it in and it should work.

Solution 7: Use a USB IR transmitter

I'm only going to provide an outline for this solution, because I haven't tried implementing it myself. You'll have to do a little product research to fill in the details.

You can buy a device for your PC that lets you plug an IR transmitter into a USB port. Software on the PC can then command the IR transmitter to send a signal. You can use one of these to send the ON command to your TV via IR remote during the Windows boot process.

I don't have any specific product recommendations, but your best bet might be to search for "winlirc transmitter" or "winlirc blaster". winlirc is open-source software that lets Windows send and receive IR commands, so a winlirc-compatible device with a transmitter should serve the function we need here.

Once you find a suitable device, install it on your PC and arrange the IR emitter so that it's within range of your TV's remote receiver. Now you just need to set up a script on the PC that sends your TV's ON command while Windows is booting. You should be able to do this by creating a .CMD file containing the command line

sequence to send the IR command, then placing a shortcut to the .CMD file in your Start menu's Startup folder.

13. I/O Controllers

One of the big things that elevates the virtual cabinet experience above ordinary desktop computer pinball is the ability to use real pinball controls: flipper buttons, coin slots, a plunger, "nudging" by actually nudging the cabinet. An equally big enhancement is feedback devices that create tactile effects, lighting effects, and mechanical sound effects that aren't just coming from a speaker.

If you're new to virtual pinball, you might wonder how all of this is possible, since normal PCs don't have any provisions for connecting any of these unusual devices. There's no "flipper button" connector on a Dell desktop. The secret ingredient is something called an I/O Controller ("I/O" for "Input/Output"). These are special hardware devices that plug in to the standard PC ports (usually USB) and provide the special wiring needed to connect buttons, accelerometers, plunger sensors, solenoids, lights, and so on. They provide the physical bridge between the PC and the unique pinball hardware.

This chapter gets into the details of what these devices do, and offers some suggestions for what to buy. The subject can seem overwhelming at first, because there are lots of product options, and they all have different combinations of features and functions. We'll try to make it easier by breaking things down by function, and giving you a comprehensive list of the products available and which functions they offer. Towards the end of the chapter, you'll find a product/feature matrix that shows everything at a glance.

I/O controller functions

Let's start by looking at the main categories of functions that these devices can handle.

Button input: A device that lets you wire regular pinball buttons to the PC is called a "key encoder". These devices are pretty easy to set up. You just run a pair of wires to each button (flipper, Start, etc) and connect them to the encoder. The encoder attaches to the PC with a USB cable. When you press a connected button, the encoder emulates either a keyboard key press or a joystick button press. As far as the PC software is concerned, you're just typing on the keyboard or using a joystick.

Nudge input: This type of device uses an electronic accelerometer to sense the motion of the cabinet. Good accelerometers are sensitive enough and accurate enough to detect when you nudge the cabinet and to measure how hard each nudge is. The pinball software can use this information to apply a corresponding acceleration to the virtual ball - in proportion to the strength of the nudge, so you can get realistic reactions for soft nudges, hard nudges, and a continuum in between. Nudge devices usually connect to the PC via a USB cable and emulate joysticks, so a physical nudge looks to the PC software like a momentary deflection on a joystick handle. The strength of the nudge is indicated by the magnitude of the deflection, which is what allows the software to differentiate between soft nudges and hard nudges.

Plunger input: This is a very specialized type of input device, because it has to use some type of position sensor to track the motion of the plunger, and then translate the readings from that sensor into a format that the PC can understand. Plunger devices usually attach to the PC via a USB cable and emulate joystick input. This is the same way most nudge sensors work, so we need a way for the PC to tell the two apart. This is usually accomplished by using different "joystick axis" assignments for each device.

Feedback output: Output controllers let you connect feedback devices to the PC so that the software can control them. As with the other devices, these usually use USB connections to the PC. Unlike the various input controllers, which all emulate ordinary PC input devices (mainly keyboards and joysticks), output controllers all need special software on the PC. Fortunately, the required special software is already integrated with the main pinball player programs.

Available devices

Now let's look at the available devices. Some of the devices fall neatly into single categories, and others can perform multiple functions.

Pinscape Controller, running on just the KL25Z (no expansion boards). Open source software, DIY hardware; about $15 for the main microcontroller board. Key encoder, plunger input, nudge input, feedback device control.

> This is an open source project that can handle all of the I/O controller functions with a single device. The main hardware required is a KL25Z, which is a $15 microcontroller that comes fully assembled and ready to use. By itself, the KL25Z can handle button input and nudging (it has an excellent built-in accelerometer). Plunger input and feedback device control require additional hardware that's described later in this Build Guide. The Pinscape software does just about everything the various single-function commercial devices do, with some added bells and whistles of its own. It includes fully assignable button inputs (using keyboard keys and/or joystick buttons), a "shift button" feature, LedWiz emulation for universal software compatibility, "night mode", high-precision nudge input, high-precision plunger input, and numerous other features. It's highly configurable via its setup program (which runs on Windows, and is free and open-source), and the firmware itself is open-source, so you're free to customize it if you need to do anything beyond what the configuration options allow, or if you want to add whole new features. The firmware includes built-in support for several types of plunger sensor technologies, so you have a choice of different plunger setups.
>
> The standalone KL25Z can handle button, nudge, and plunger input with little more work than attaching wires. It gets a little more complicated if you want to use it with feedback devices, because it needs some additional electronics to do that, as explained in Chapter 49, Pinscape Outputs Setup (Standalone KL25Z).

Pinscape Controller with expansion boards. Open source software and hardware design; components cost about $100 for a full build. Key encoder, plunger input, nudge input, feedback device control.

> This is an extension of the basic Pinscape Controller project that adds a set of circuit boards, primarily to provide more feedback device outputs. The boards make it possible to control a much larger number of feedback devices than the KL25Z can control on its own. The boards also provide built-in handling for high-power devices, so that you can connect things like motors, solenoids, replay knockers, fans, and flashers without any additional booster circuits. The hardware design is open-source, so you can build everything yourself from components, which add up to about $100 for a full-featured build. You can also opt to build only sections of the boards if you only need a subset of the features, which reduces the cost accordingly.

Zeb's Boards plunger kit. Commercial, about $140 from zebsboards.com. Plunger input, nudge sensor, key encoder.

> This kit comes with the control board and plunger sensor that attaches to a standard pinball plunger (available separately for about $30). In addition to plunger input, Zeb's kit also handles nudging via an on-board accelerometer, and provides key encoding for up to 20 buttons (with fixed key mappings). Zeb's plunger gets the best user reviews of the commercial plunger options. It uses a high-precision sensor for the plunger that provides realistic plunger motion in the pinball simulation.

VirtuaPin plunger kit. Commercial, $140 to $160 from virtuapin.net. Plunger input, nudge sensor, key encoder.

> The VirtuaPin kit comes with a control board and plunger sensor, and optionally includes the physical plunger assembly. Like other commercial plunger kits, the VP kit is very easy to set up, with little assembly required beyond attaching the sensor to the plunger. The control board has an excellent on-board accelerometer for nudge sensing, and has wiring for up to 16 button inputs. Button inputs are hard-coded as joystick buttons and can't be assigned to keyboard keys. If you're picky about realism in the plunger, be aware that this kit uses an IR proximity sensor to detect plunger position, and these sensors have relatively poor distance resolution. Some users have reported that the plunger animation can be choppy.

i-Pac 2 and i-Pac 4. Commercial, $39/$59 from ultimarc.com. Key encoder.

> The i-Pac devices are full-featured key encoders. Their target market is video game cabinet builders, but they work equally well for virtual pinball, since the needs are basically the same. Buttons are fully assignable (via a setup program

on the PC) to keyboard keys and joystick buttons. The devices have a "shift button" feature that lets you assign two meanings to each physical button by holding down a designated shift button to activate the second meaning.

i-Pac Ultimate I/O. Commercial, $99 from ultimarc.com. Key encoder, feedback device control.

This is a hybrid of the i-Pac and PacLED devices that provides button input encoding and feedback device control. The key encoder features are just like the i-Pac devices, with 48 button inputs. The feedback output controller is designed specifically for attaching 32 small (20mA) RGB LEDs. For a virtual pinball cabinet, you'll want to attach other devices that require higher power, so you'll need external booster circuitry, such as Zeb's booster board. One warning: as of this writing, this device's output controller feature isn't as well supported in the standard virtual pinball software as the LedWiz and PacLed devices, so you might encounter some difficulty setting up the software to take advantage of it. The button input feature will work seamlessly, though.

LedWiz. Commercial, $45 from GroovyGameGear.com. Feedback device control.

The LedWiz was the first output controller widely adopted among virtual pinball cabinet builders, and as a result, it's the most universally supported option. This device is aimed at video game cabinet builders, so it was designed especially for controlling LEDs (thus the name), but it's not limited to LEDs. It can control just about any type of device. The caveat is that it has a low limit on how much current it can control per device (500mA), so you can't connect high-power devices directly. You can work around that by adding an external booster board to increase its power limits. That 500mA limit is adequate for most types of lights, including flasher LEDs and button lamps. A booster is needed for most mechanical devices, like knockers, motors, and solenoids.

PacLed-64. Commercial, $59 from ultimarc.com. Feedback device control.

This device is well supported by the newer open-source pinball software systems (including Visual Pinball and PinballX), but it's not as compatible with older systems like Future Pinball as the LedWiz is. It provides 64 outputs for small LEDs. Like the LedWiz, this device was designed for video game cabinet builders, but its power handling is even more limited and isn't sufficient for high-powered lights like flashers and strobes. So you'll need to combine this with a booster board for almost anything in a virtual pinball cabinet.

SainSmart USB relay boards. Commercial, about $20-$40. Feedback device control.

SainSmart makes USB-controlled relay board with 8 relay outputs. Software on the PC can send USB commands to turn attached devices on and off through relay switches. The relays can be used to control devices that use high power levels, so they're good for devices like solenoids, contactors, and replay knockers. However, these boards aren't a good choice for lighting devices, since relays on simple on/off switches and thus can't control brightness. For lights (especially flashers and button lights), you'll want to be able to control the intensity level of each output. The other slight disadvantage of relays is that they add a small lag time for switching devices on and off, which can make the device response slightly out of sync with the game action. Most people don't find this noticeable, though.

Warning! DOF is currently only compatible with the **8-relay** Sainsmart boards. Sainsmart makes the boards in different sizes, from 4 to 16 channels, but DOF **only** works with the 8-relay version.

Warning! There seem to be some no-brand devices out there that look ridiculously similar to the Sainsmarts, with the same blue lays laid out the same way, but which aren't compatible at the software level. That means they won't work with the existing pinball software, unless you can do some additional programming to add support yourself. I'd avoid look-alike boards that aren't clearly branded as Sainsmart products.

Zeb's Boards booster board. Commercial, $75 from zebsboards.com. Feedback device add-on.

This board lets boost the power from 16 outputs on an LedWiz or PacLed output controller. The booster board itself isn't an output controller, so you can't use it

alone; it has to be used in conjunction with one of the output controllers. The booster board raises the power level on 16 of the output controller's ports to 6A, which is enough to control anything in a pin cab, including high-power devices like replay knockers, shaker motors, gear motors, fans, beacons, and solenoids. If you need more than 16 boosted ports, you can add more of these boards to boost an additional 16 ports per board.

SainSmart (non-USB) relay board. Commercial, $20 to $40. Feedback device add-on.

These boards are similar to the SainSmart USB relay boards, but they're not controlled by USB. Instead, they're controlled by individual inputs to the relays. You can connect the relay control inputs to the output ports of an LedWiz or PacLed unit to boost the power handling capability of the controller via the relays. You can then attach a high-power device, such as a replay knocker or solenoid, to the relay. The controller unit will switch the relay on and off, and the relay will in turn switch your high-power device on and off. This is a simple way to boost the power handling of an LedWiz or PacLed unit. Note that the relay switching adds a small amount of lag time, which can make the feedback response slightly out of sync with the game action, although most people who have set these up don't find this to be noticeable.

Zeb's Boards output kits. Commercial, $550 to $900 from zebsboards.com. Feedback system including controller *and* feedback devices.

These kits offer turnkey feedback setups that include not only the output controller device but also all of the feedback devices themselves, all fully assembled and wired. Everything comes pre-mounted to a couple of modular panels for easy installation in a cabinet.

Recommendations

For the DIYer: I'm biased, obviously, but if you like building things yourself, my pick would be Pinscape. For a fully decked-out system with all the feedback devices, go with the expansion boards. For the input features only (buttons, plunger, nudging), the standalone KL25Z is all you need. I'm pretty sure Pinscape has all of the features of the best-of-breed commercial products (plus some extra features they don't have), equal or better performance, and a lower price tag. And the open-source design puts you in complete control. You can change anything that's not to your exact liking; and if you take "DIY" especially seriously, you can use my code as a starting point and rewrite as much of it as you want from scratch.

If you want "no compromises": Again, I'm biased, but I think the answer here is Pinscape. It has the most full-featured and highest performance implementation I'm aware of for each of the components. It's highly configurable through its Config Tool, so you can set it up exactly how you want it. And again, it's open-source, so if there is anything you want it to do that it doesn't already do, you can add it; or if there's anything it does do that's not quite the way you want it, you can change it.

If you're uncomfortable with DIY: You'll probably be happier with the commercial options if you're not comfortable building this sort of thing yourself. The commercial products come ready to install, with only some basic setup required. The big challenge is figuring out which devices you need, since their functions overlap in somewhat confusing ways. Here are my recommendations for some common scenarios:

For a simple feedback system with lights only: If the only feedback devices you want are lighting devices (flashers, strobes, and button lights, for example), I'd recommend an LedWiz as the output controller. The LedWiz is inexpensive, and for just lights, it's simple to set up, since that's exactly what it's designed for. A single LedWiz has plenty of ports for a pin cab's lighting needs. The LedWiz is a good choice for lighting devices because it can display a range of brightness levels, which allows for fades, flash patterns, and RGB color mixing effects. The LedWiz isn't as ideal for high-power devices like solenoids and motors, since it can only handle limited power to each port; while it's possible to use it for these devices, you need additional hardware add-ons, which largely negates the whole "it's simple" advantage.

For a simple feedback system with solenoids and motors only: If you want a feedback system consisting only of tactile effects (replay knocker, flipper and bumper solenoids, shaker motor), get a SainSmart USB relay board. I'd get the

16-output type so that you have plenty of outputs for extras you might want to add later. The SainSmart board is the easiest thing to set up for high-power devices. The downside is that relays are strictly On/Off switches, so the SainSmart can't display different brightness levels if you use it to control lights - it can only turn them fully on and fully off. That makes it good for devices like solenoids and motors, but not so good for lamps and LEDs, where you need brightness control to get the full range of effects. The other disadvantage is that the relays are mechanical, so they can eventually wear out; some people on the forums have reported having to replace their SainSmart boards every couple of years due to relay failure.

For a plunger-less system: If you don't want to include a plunger in your setup, use a KL25Z running Pinscape as the input device. You don't need the expansion boards if you're just using the input features. The installation work for buttons and nudge input is pretty much the same as for any of the commercial options, and Pinscape is a lot cheaper and has more features.

For a turn-key plunger: If you want a plunger but don't want to build the electronics yourself, buy Zeb's plunger kit. It's easy to set up and gets generally good reviews from users.

For a turn-key feedback system: If you're the opposite of a DIYer, and you don't want to do a lot of planning or parts sourcing or assembly work, buy one of Zeb's pre-built feedback kits. They're expensive, but they'll save you a lot of work, and they'll eliminate any anxiety you might feel about the things going wrong if you build it yourself.

Feature matrix

Here's a summary of the key features of the available controllers, to help you decide on a combination of devices for your system based on the features you plan to include.

Device		Key Encoder			Plunger		Nudge	Feedback Output			
Name	Type/ Price	# Buttons	Assignable	Shift Button	Sensor Type	Precision	✓	# Outputs	Power Limit	Brightness Control	Booster Required
Pinscape (standalone)	Open source $15	24+	✓	✓	Multiple options	High	✓	22+	4mA	✓	Yes
Pinscape w/expansion boards	Open source ~$100	24+	✓	✓	Multiple options	High	✓	65-128	4A[1]	✓	No
Zeb's Boards plunger kit	Commercial $140	20	-	-	Poten-tiometer	High	✓				
VirtuaPin plunger kit	Commercial $140	16	-	-	IR	Low	✓				
i-Pac 2	Commercial $39	32	✓	✓							
i-Pac 4	Commercial $65	56	✓	✓							
i-Pac Ultimate I/O	Commercial $99	48	✓	✓				96	20mA 6@1A	✓	Yes[2]
LedWiz	Commercial $45							32	0.5A	✓	Yes[3]
PacLed-64	Commercial $59							64	20mA	✓	Yes[2]
SainSmart USB relay board	Commercial $20-$40							4-16	12A		No
Zeb's Boards booster board	Commercial $75							16	6A	✓	No[4]
SainSmart relay board (non-USB)	Commercial $20-$40							4-16	12A		No[4]
Zeb's Boards output kits	Commercial $550-$900							16-64	1A-6A	✓	No

Footnotes:

1. The 4 Amp limit applies to the general purpose outputs on the power board. There are 32 of these on each power board. In addition, the main board has 16 flasher/strobe outputs that can handle 1.5A each, and 16 outputs for button LEDs that can handle 20-50mA each. The typical setup uses one main board and one power board, which gives you 65 total outputs, plenty for a decked-out cab. If you need more, you can add extra power boards for another 32 of the high-power outputs per, up to the software limit of 128 total outputs.

2. This device's outputs are designed to drive low-power LEDs, which it can do without any extra booster circuitry. A booster board is needed to drive anything needing higher power, such as flasher LEDs or mechanical feedback devices.

3. The LedWiz can handle 500mA per output, which is sufficient for most types of lights, including LED flashers and button lamps. A booster board is required for most

non-lighting devices, such as contactors, replay knockers, solenoids, fans, shakers, and gear motors.

4. This device works in conjunction with one of the output controllers (LedWiz, PacLed-64, etc). It can't be used alone; it has to be used in combination with an output controller.

14. PC Hardware Setup

If you bought an assembled PC, all you have to do at this stage is unpack it. If you're building a PC from parts, you might want to do a preliminary test build at this point to make sure everything's working, and to confirm that you have everything you need.

If you're going to install your PC equipment in your pin cab without a standard PC case, you can do your test build with the parts spread out on a tabletop. Be sure to do your work on a non-conductive surface like wood to avoid accidental shorts.

Static electricity precautions

Many of the parts in a PC are sensitive to static electricity, particularly the CPU and memory chips. That's why everything comes packaged in those silvery plastic anti-static bags.

Your body can accumulate a significant static charge, enough to damage semiconductors, so you have to be careful handling these parts to avoid zapping them when you touch them.

The way to protect against damage when handling static-sensitive parts is to frequently "ground" yourself, meaning that you electrically connect yourself to the earth. Professional engineers do this with something called a grounding strap, which is a conductive bracelet that you wear on your wrist and connect by a wire to your house's ground wiring. That keeps you connected to the earth ground the whole time you're working. You probably don't have one of these unless you do a lot of electronics work, but you can achieve the same thing by simply touching a metal surface that's connected to earth ground periodically while you're working.

Where do you find a grounded metal surface to touch while working? When you're working on a PC, there's one that's always close at hand: the power supply case.

Here's what I suggest. Before you start any other work, get your power supply out of the box and **plug it in**. Put it on the table where you're going to do your work. Find a **bare metal surface** on the power supply case. If there are no bare metal surfaces, an unpainted screw head on the case will work. The key is bare metal attached to the case.

As long as the power supply is plugged in with a three-pronged plug, you can now ground yourself at any time by touching that bare metal surface.

You don't have to keep in contact with the grounded metal all the time, although it's ideal if you do (which is why they invented those grounding bracelets). But do ground yourself frequently while working, at least every few minutes, and every time you return to the work station after walking around. Walking around is is a great way to accumulate static charge. Another good time is whenever you're about to open an anti-static bag or start handling a new part.

Assembling the motherboard

You should find detailed installation instructions for assembling your motherboard in its packaging. If you bought a retail-packaged CPU, it should also include its own instructions for installing it in the motherboard.

You should follow the setup instructions in your motherboard's documentation, since every board is a little different. In general, though, here are the steps:

- Install the CPU in its socket. Be especially diligent about static electricity precautions while handling the bare CPU.
- Install the CPU fan. Most motherboards have clips or sockets for securing the fan, so this is usually just a matter of fitting the fan into place.
- Connect the CPU fan wiring. The CPU fan should have a short wire connector that mates to a "CPU FAN" socket or pin header on the motherboard.
- Plug in the memory (RAM) chips.
- Insert the video card into its motherboard slot. Note that some motherboards have multiple slots that are physically capable of holding the video card, but

one slot might be better than the others because it has a faster data connection. Check your motherboard documentation to see if one slot is designated as the special slot for the video card. The documentation might not put it in these terms; it might instead list the PCI "x" speeds for the different slots, such as x1, x4, or x16. The highest "x" speed is the fastest slot, so it's the one to use for the video card.

- Connect the storage device (hard disk or SSD).
- Connect the power supply to the motherboard. There are usually two connectors from the power supply that plug directly into the motherboard.
- Connect the power supply to the video card. Most higher-end video cards have a dedicated connection directly to the power supply. Your power supply should have the special mating plug attached to one of the wires in the bundle of cables coming out of the supply.
- Connect the power supply to the hard disk or SSD. These devices have their own power connection. As with the video card, the hard disk/SSD has a dedicated power connector that will mate with one of the connector wires coming out of the power supply.
- Connect a video monitor to the video card.
- Connect a keyboard and mouse to USB ports on the motherboard.
- If you have a wired Ethernet network, connect a cable to the network plug on the motherboard.

Video card precautions

If you're assembling everything on a tabletop without a standard PC case, the video card's connection to the motherboard will be fragile, since it won't have the structural support that a normal case provides. The PCI slot is really only meant to provide the electrical connection; it's not meant to provide structural support or anchor the card in place.

You'll definitely need to secure the card physically in your eventual installation in the pin cab. For testing purposes, though, you can work with this flimsy setup as long as you're careful not nudge anything while the power is on. Even a little nudge, like someone bumping into the table, can be enough to momentarily interrupt the electrical connection in the socket. I'd avoid that; in most cases the only harm will be to make the PC reboot itself immediately, but these cards really aren't meant to be hot-plugged (taken in and out with the power on) and could be damaged by this.

Power it on

Once you have everything assembled as described above (and according to any additional instructions in your motherboard's documentation), you're ready to give it a test run.

Modern PC motherboards have "soft power" controls, so even though it's already connected to the power supply, it won't actually turn on until you press the "on" button. If you have a case, the "on" button is the one on the case. If you're working without a case, though, you have to find the "Power Switch" pins on the motherboard. Check your motherboard documentation to find the right pins. The motherboard manual will tell you that these are the pins to connect to the "Power Switch" connector wires from your case.

Once you identify the "power switch" terminals, you turn the PC on simply by shorting these two pins together for a moment. They're usually right next to each other on the motherboard, so if you don't have the right kind of connector handy, you can simply touch a metal screwdriver tip to both pins at the same time. (Be careful not to touch any other pins while doing this, of course.)

BIOS Setup

When the PC first powers up, you should see a brief message flash on the screen telling you to press a key on the keyboard to enter the BIOS Setup. It's usually one of the function keys, often F8 or F12, but it varies by motherboard. You'll just have to watch for the message to find the right key. You should also be able to find this information in your motherboard's documentation.

You should be able to reboot by pressing Alt+Ctrl+Del on your keyboard, which should give you another chance to press the magic BIOS Setup key. You can also power cycle by shorting the "Power Switch" pins together again to turn the PC off, then wait a few seconds and do it again to power up again.

You have to press the magic BIOS Setup key at just the right moment after the power comes on. It usually works to tap on the key rapidly while the machine is powering up.

The BIOS Setup lets you configure the machine's hardware and verify that everything you physically attached (memory, disk, video card) is being properly recognized by the motherboard. It's worth running the setup as a very basic test, since the fact that it runs at all confirms that the video card, keyboard, CPU, and memory are all working.

15. Windows Setup

Once you have the PC hardware set up, the next step is to install Windows.

We won't try to provide a Windows installation tutorial here, since the Web has much more comprehensive information on that than we could provide here - and you probably won't need to look at any of that anyway, since Microsoft has managed to make the process fairly automatic in most cases. But we do have some recommendations for settings specifically for pin cabs.

So continue below after you've gone through the basic Windows installation procedure.

DON'T turn off UAC

User Account Control (UAC) is a Windows security feature added in Windows Vista, and present in all Windows versions since, that makes Windows ask for your permission when an application tries to do something that affects core resources in the operating system, such as altering a system registry setting.

You might see advice on the forums telling you to turn off UAC to avoid those prompts. I recommend ignoring that advice. **Leave UAC enabled.**

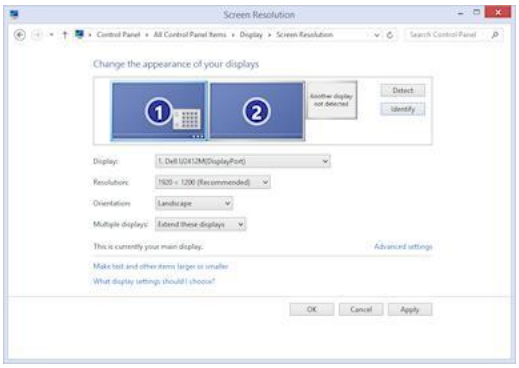
I recommend **against** disabling UAC because doing so can cause software compatibility problems. UAC is an integral part of modern Windows systems, and removing it actually changes the way Windows works internally, which can break some application software. If you're technically inclined and curious about the details, see Mark Russinovich's article in Microsoft's TechNet Magazine, June 2007, Inside Windows Vista User Account Control. For a less technical explanation, try a Google search for "Why not disable UAC". Disabling UAC also increases your vulnerability to system damage from malware and unintentional software bugs.

Any advice you see about disabling UAC is likely outdated. The notion comes from the early days of Windows Vista, when UAC was first introduced. A lot of software at the time wasn't properly designed for the tighter security rules added in Vista, so some people resorted to disabling UAC in an attempt to keep their old software running. As time has gone on, though, most software has been updated to work properly, and UAC itself has been improved to make it less intrusive.

Arrange monitors

If you're using multiple monitors, Windows will combine their display areas into a single large virtual desktop.

Windows lets you adjust the way the monitors are arranged within the virtual desktop via a control panel. The idea is that if you have two physical monitors sitting side by side on a desk, you can arrange the virtual desktop to match. To reach this control panel, bring up the Display control panel, then click Adjust Resolution.



This control panel shows a diagram of how the physical monitors are lined up across the virtual desktop. You can drag the monitors in the diagram to rearrange them.

How you arrange your monitors is mostly up to you, but there's one important rule you should follow:

The main display should be at the upper left of the virtual display area.

Note that the "main display" isn't necessary display #1. The numbering is just a way to identify the monitors and is somewhat arbitrary. The "main display" is simply the one you designate as such using the "Make this my main display" button in the control panel.

I recommend the following layout:

- Make your playfield monitor the main display
- Arrange the monitors in a single row
- Make sure the main display is at the left end of the row

Some versions of Windows only allow certain monitors to be the main display, so you might not have the option to make your playfield monitor the main one. If you can't, you should still arrange things so that the main monitor is at the left end of the row.

The reason I recommend this arrangement is that some software, notably VPinMAME, can have odd problems if the main monitor isn't at the left extreme of the virtual desktop. Windows internally assigns the "origin" of the pixel coordinate space to the top left of the main monitor, so any monitor that's to the left of this (or above it) in the virtual desktop area will have negative pixel coordinates. Some software (like VPinMAME) gets confused by the negative coordinates. If you don't follow this advice about the layout, VPinMAME won't be able to properly remember your screen layout, because it incorrectly interprets negative coordinates as errors.

Turn off "Sticky Keys"

Most pinball software uses the Shift keys to control the flippers. Windows has an accessibility feature called "Sticky Keys" that locks the shift keys on if you press them several times in a row. The feature is well-intentioned - it's there to help people who have difficulty pressing several keys at once - but it interacts horribly with pinball games. It can make the flipper get stuck in the up position after a bunch of rapid flips.

Sticky Keys is an accessibility feature, so you'll find it on the "Ease of Access Center" control panel, which you can find in the main Control Panel window.

On most versions of Windows, you can also find this control panel by pressing Windows+S ("Search") and typing "Sticky Keys" into the box. Look for "Change how your keyboard works" or "Make the keyboard easier to use" in the search results.

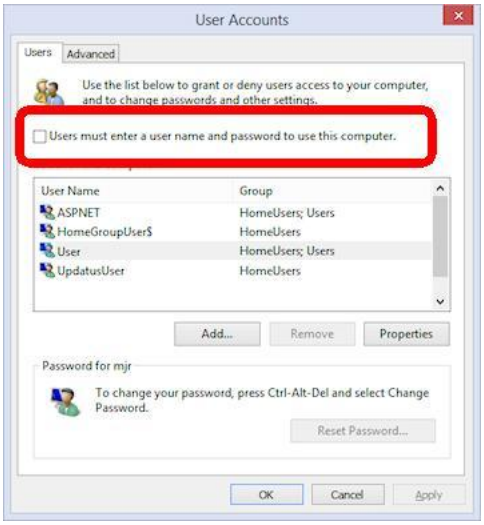
Once you find the dialog, look for the checkbox labeled "Turn on Sticky Keys". Make sure it's un-checked.

Automatic login

Windows normally asks you for a username and password every time you start up the computer. This is great for office or work PCs. It's not great for pin cabs, where you want to be able to turn on the machine and get straight to playing pinball. It's a little crazy to have to get out the keyboard and log in first.

Fortunately, Windows lets you disable the login requirement, so that the machine boots straight to the Windows desktop each time you turn it on. Here's the procedure:

- Press Windows+R ("Run Program")
- Type **netplwiz** into the Run box and press Enter. This should bring up the advanced user accounts control panel (titled "User Accounts" on most Windows versions).



- Look for the checkbox "Users must enter a user name and password to use this computer" (circled on the screen shot above). Un-check this box.
- Click OK to confirm the changes.
- A new dialog should appear asking you to enter the credentials to use to sign in automatically when Windows reboots. Enter the username and password you wish to use and click OK.

The next time you boot, Windows should automatically log in to the account you selected and go straight to the desktop.

Remove (or don't install) anti-virus software

On any gaming PC, it's best to minimize the number of background tasks running. Background tasks take CPU time away from the main program that's running. This can have a visible effect on the animation in a game, since even a very short interruption in the animation updates can cause momentary glitches and stutters.

Probably the most important background task to get rid of is third-party antivirus or anti-malware software. Virtually all of these programs use significant system resources and will noticeably hurt game performance. If you built the PC yourself and did a fresh install of Windows, you can simply elect not to install any third-party security software. If you bought a pre-built PC, and the vendor larded it up with "free trial" security software, I'd remove it all.

It might seem crazy in this day and age to run a PC without any security software, and I certainly wouldn't recommend going without on an ordinary PC, but a pin cab isn't an ordinary PC. The difference is that you'll probably only use it for playing pinball - not for browsing random Web sites, opening random emails, or downloading random programs. As long as you're careful about what you install, your risk of encountering any malware should be small. Stick to the well-known pinball programs and add-ons, and always get them from reputable sites.

An exception: you can (and should) leave the built-in Windows security features enabled, particularly Windows Defender and the Windows Firewall. Those have a negligible impact on system performance, and they provide a good baseline level of protection.

Backing up your system data

Everyone knows how important it is to back up the data on a PC, in case you ever need to recover from hardware failures, accidental file deletions, or malware attacks. It's a lot of work to set up all the software on a pin cab, so backups are as important for a pin cab as for any other PC.

The approach I've used for a long time is to back up to external USB hard disks. Those are reliable and fairly inexpensive, and most of them come bundled with

backup software. More recently I've added cloud backup as a second layer of protection. There are several good on-line backup services that run about $10/month for reasonable storage quotas.

Here are some things I consider important when setting up your backup plan:

- It should be **automatic**. It should run on a schedule so that you don't have to remember to run it yourself. It's too easy to put it off or forget about it entirely if you have to do it manually. The cloud backup services make this particularly easy.
- The media should be **offline** between backup sessions, meaning not physically connected computer you're backing up. This will protect your data in case of a hardware failure (such as a power spike that fries everything connected to the computer) or a system-wide malware infection. If you back up to an external USB disk, simply unplug it from the computer after each backup.
- Better still, the media should be **off-site**, at a physically separate location. This will protect your data in case of a whole-house disaster like a fire or flood. This is a big benefit of cloud services.
- Backups should be **versioned**. Versioning is particularly critical for malware protection, because an infection might not be immediately apparent, so your most recent backup might include infected files without your knowing it. Keeping multiple versions lets you go back in time to a point before the infection. Versioning is also a nice safety net in general - it lets you go back to an older working configuration if something goes wrong with a software update, for example.
- The backup software should do a whole-disk scan. If you have to manually choose the files that get backed up, you'll inevitably miss something important. I always prefer starting with a default that includes everything on the disk, and then manually selecting files to exclude.
- The backup scan should **include the Windows registry** as part of the backup, since Windows itself and many application programs store a lot of important configuration data there.

16. Pinball Software Setup

Now let's look at the software needed to transform this from a plain Windows PC to a pin cab machine.

The main software you need, of course, is the pinball simulator. To take advantage of the special features of a cab, you also need some add-ons to display the backglass artwork and control the feedback devices. You'll also want a "front end" program that provides an interactive menu for selecting tables to play. We'll look at the options for all of these in this chapter.

Should I use "Run as Administrator" for everything?

The simple answer is No. You might find advice in the forums or FAQs or other guides saying that you should routinely run everything in Admin Mode. My advice is to ignore that other advice, because it's usually outdated or misinformed.

What is "Admin Mode" anyway? Microsoft divided things into "Admin" and "User" spaces to protect the internals of the system from being accidentally damaged by software bugs or user errors, or intentionally damaged by malware. Programs running in User Mode have some restrictions on what they can access, but Admin Mode programs can access everything. Admin Mode is supposed to be reserved for special system programs that have a legitimate reason to modify your system internals - programs like installers, disk management tools, and system control panels. Everything else is supposed to run in regular "User" mode.

So why do so some people on the forums tell you that you *should* use Admin Mode routinely, when Microsoft says you shouldn't? The reason is mostly "history". In the old days, there were a few isolated software components in the pinball simulation ecosystem that really did require Admin Mode to work properly. The snag is that Windows erects protective barriers around an Admin Mode program. Those barriers prevent it from interacting with regular User programs. But the pinball ecosystem is made up of a bunch of programs that were designed to interact with each other. So if you run program X in Admin Mode, and program Y needs to interact with it, then you *also* have to run program Y in Admin Mode. I think you can see where this idea that "you've got to just run everything in Admin Mode all the time" came from - it was a blunt instrument, but it was a way to get around these program interaction problems that Admin Mode created.

Okay, so if "Admin Mode everywhere all the time" is a simple way to solve thorny problems, why am I saying you *shouldn't* use it? The main reason is that, while it might solve some problems, it creates others. Microsoft doesn't want you to use Admin Mode routinely for everything, so you're always somewhat fighting with Windows if you do. It also reduces your system's security by defeating all of the protective mechanisms that Microsoft designed into Admin Mode in the first place.

The right solution - from a security perspective, and in terms of simplicity - is to stop using Admin Mode for *any* of your pinball software. If you run everything in regular User Mode, everything will be able to interact as it was designed to, with no hassles at the system level. Remember how I said that this whole Admin Mode fiasco is historical, because it was a requirement for certain components *in the old days*? Fortunately, it really is mostly relegated to the past now. Those old Admin Mode requirements were almost all due to software bugs, not actual engineering requirements, and all of the cases that I'm aware of have been fixed in modern versions. As of 2020, I don't think that any of the common pin sim components require Admin Mode, as long as you've updated to current versions.

If you do encounter any up-to-date pinball-related programs that say "Admin mode required", you should take a critical look at them and make sure the requirement is real, not just a misunderstanding. There's still a lot of confusion about this, so you can't always trust the FAQs and guides. My personal policy is that I simply won't run programs with unnecessary Admin Mode dependencies until the developer fixes them. I realize that not everyone can bring themselves to be so ruthless, when faced with a fun new feature that they really want. If you find a program that you can't live without, and there's just no way around its "requires Admin Mode" problem, I'd at least try to hold firm on one thing: don't let it "infect" the rest of your system with its Admin Mode requirement. One concrete thing you can do is to use PinballY as your front end. It has the ability to launch Admin Mode programs *without* running in

Admin Mode itself. A major cause of the Admin Mode infection is that none of the other front ends can launch Admin Mode programs unless you also run them in Admin Mode, and of course if you do that, everything they launch will be in Admin Mode. And as I said, that might *seem* like it works for a while, but it's likely to eventually cause its own problems.

Customization log

Before you do anything else, I think it's a good idea to create "customization log" file. This is just text file for your own use - you can create it with Notepad and leave it empty for now. Put it someplace where you'll be able to find it easily in the future, such as right on the Windows desktop on your cab PC.

The point of this file is to jot down all of the special customizations you make to Visual Pinball and other software. VP in particular forces you to make some customizations in ways that you'll have to repeat each time you update to a new version. For example, some customizations require that you hand-edit VP's shared script files, and those changes will be lost on each update because VP will overwrite the scripts with its own updated copies. That's not a very friendly design on VP's part, I know, but it's just the way some things in VP work.

I'll mention this file again in other chapters when these sorts of changes come up, with a suggestion that you make a note in your customization log file. For now, just create the file so it'll be ready when you need it. In the future, whenever you make a change that warrants inclusion, add a note about it to the file. When it comes time to update VP or other software, you can refer back to this file to reinstate any customizations that got lost in the update process. The same goes if you ever have to rebuild your Windows system due to a system upgrade or disk failure.

Free pinball players

There are three main free pinball player programs for Windows:

- Visual Pinball 9
- Visual Pinball 10 (also known as VP X)
- Future Pinball

Visual Pinball is the essential program for a virtual cab. VP is an open-source project with an active developer community and frequent updates. Hundreds of tables are available, including re-creations of a pretty good percentage of all of the real pinball machines across the decades, plus many original tables. VP has excellent support for the whole gamut of special pin cab features: backglass monitors, DMDs, feedback devices, plunger inputs, accelerometer nudging.

I counted VP 9 and 10 as two separate programs because they're not compatible with each other's tables, so you really have to install both. There's also a much older version 8, plus a couple of different, mutually incompatible versions of VP 9. Some people like to keep all of these installed because, again, individual tables are all tied to specific VP versions, so you need all of the VP versions if you want the ability to play all of the tables out there. (VP isn't very good at compatibility.) Fortunately, there's a combined installer that sets up the whole collection of VP versions with a single download and a single install process.

Future Pinball is another free player, but unlike VP, it's no longer being maintained or updated. Its original creator abandoned it a long time ago and never released the source code, so it's basically a dead end. Even so, you might want to install it to gain access to its tables, since there are a few re-creation tables (particularly from the 1970s or before) where there's an FP version but no VP version.

Visual Pinball 9 and 10

Visual Pinball 10, or "VP X", is the latest version, and VP 9 is the previous version. You'll want to install **both** versions because they're not compatible with each other's table files, and you'll want to be able to run both kinds of tables.

You can recognize VP 9 tables by the ".vpt" filename suffix. VP 10 tables use the ".vpx" suffix.

The easiest way to set up both versions is to use the VP Installer. VP is actually a collection of about five programs that work together, and in the old days, you had to

go download them all individually and then go through a complex series of steps to configure them. The VP Installer bundles everything into a single download, and provides a Windows Setup program that configures it all automatically.

Here are the steps to install VP (both versions 9 and 10) with the VP Installer:

- Go to vpforums.
- On the navigation bar near the top, click Getting Started. This will pop up a menu. Under "Install Visual Pinball", click "VP Installer".
- Even though this is called the "VP X Installer", it's actually the combined installer for VP 9 *and* 10.
- Click the Download button and follow the instructions to download the file. If you see several version options, pick the one with the highest number, since it should be the latest. You might need to create an account and log in before you can start the download.
- Unzip the downloaded file into a temporary folder on your hard disk.
- Double-click the Setup program.
- **Important:** when the program asks for a destination folder, use a folder in your hard disk's root folder, such as **C:\Visual Pinball**.

  You can use a folder different name, but **don't** use anything within the Windows "Program Files" folder tree. Yes, that's the *normal* location for installing programs, but don't use it for VP. You'll create huge headaches for yourself if you do. The issue is that some VP components need to write files to their own install folders, and Windows has security restrictions against programs writing within the Program Files tree. The simple solution is to install VP somewhere else.

- If the program asks which DMD components to install, it's talking about the special "Dot Matrix Display" hardware devices that you can optionally install in your cab to re-create the plasma scoring display on 1990s pinballs. If you're using a video monitor (such as a small TV or laptop display) for this, or you don't have a DMD panel at all, use the default option. If you're using a special external DMD device (PinDMD 2, PinDMD3, or Pin2dmd), select the corresponding option.

  The VP Installer asks this question because each of the external hardware DMD devices require their own special software. The VP developers are working to combine all of this into a single unified system, which will eventually make it unnecessary to choose which one to use. If the installer doesn't ask this question, don't worry - it means you have a newer version with the unified software.

Future Pinball

Future Pinball isn't as essential as VP. It's an older system that hasn't been updated since 2010, and it's unlikely that it ever will be updated again, since its author abandoned the project without ever publishing the source code. I don't find its physics as convincing as VP's, and due to its age, FP's support for special cabinet features is limited.

Even so, many cab builders think FP is worth installing, since it's free and it has lots of tables available.

You can recognize tables written for FP by the ".fpt" filename suffix.

To install FP:

- Go to the Future Pinball site, futurepinball.com
- Click on the Download button near the top of the page
- Click on the Download link
- Run the downloaded .exe file, which will set up the program for you

Commercial pinball players

Some good commercial pinball games are also available. Here are the main commercial titles popular with cabinet builders:

- Pinball FX. A commercial pinball simulation available on Windows and other platforms. In 2018, this company acquired the Williams licenses that Farsight (see below) formerly held. They're gradually releasing table packs featuring re-creations of Williams/Bally/Midway titles. Pinball FX also offers a large collection of "fantasy" titles (original tables that never existed as real machines) from before they bought the Williams licenses, many based on popular media themes including the *Star Wars* movies and Marvel comics. Their older fantasy games had a decidedly unreal flavor, as they chose to fully embrace their video-game-ness by including elements that would have been impossible in a physical table. For some people that's a positive, since it makes the game action more diverse than in a real pinball machine, but it can be a negative if your tastes run more toward simulation and realism. Recognizing this, the FX developers say they've made changes to the physics engine in the new re-creations to make them play more realistically. This product has a Pin Cab mode available; to get it, you have to send a request to the publisher's tech support staff and provide proof that your cab is operated non-commercially.
- The Pinball Arcade by Farsight Studios. Detailed and accurate re-creations of real machines from the 1960s through the 2000s, available on Windows and other platforms. TPA *formerly* boasted a large collection of Williams/Bally/Midway titles that included many of the best pinballs ever made. But Farsight's license to those titles was terminated in 2018 (to be taken over by the Pinball FX developers), so the editions you can buy now only include Gottlieb and Stern titles. Gottlieb dominated the EM era, so there are some great classics in there if you like the older machines, and Stern has been steadily producing newer machines since Williams withdrew from the market, many of which are popular and well-regarded.

The commercial games are playable on pin cabs, but they cater mostly to desktop users, and have limited support for pin cab features (DOF, multiple monitors, real DMDs, etc). Pin cab users aren't a big enough market to attract much commercial support, and of course the open-source developers who created all of the pin cab technologies are unable to modify closed-source commercial products.

Cabinet enhancements

Visual Pinball and the other pinball player programs are basically PC video games. To take full advantage of a cabinet, there are some additional pieces of software that you need.

Backglass display software

To display backglass artwork when playing Visual Pinball games, you need an add-on program called B2S Backglass Server. B2S is installed automatically along with VP if you used the VP Installer. If you set up VP manually, you'll have to install B2S separately.

Getting B2S working takes a few additional steps beyond just installing the software. We cover the details in Chapter 17, Backglass Software Setup.

Tactile feedback and lights

If you're installing any feedback devices in your cab - solenoids, shaker motors, flashing lights - then you need some additional software called DOF (DirectOutput Framework) to control the feedback devices.

DOF is an add-on program that lets Visual Pinball and other software access your output controller. DOF acts as the coordinator between the simulated game and the physical feedback devices, to synchronize feedback effects with the game action: firing your flipper solenoids when the flipper flips, activating the shaker motor when the castle is destroyed, etc.

DOF is a fairly big subject, so it gets its own chapter: Chapter 46, DOF Setup.

PinVol

PinVol is a utility I wrote to make it easier to control the audio volume during play. It lets you adjust the volume using cabinet buttons, and its special ability is that it helps equalize the volume level across different tables. It remembers your volume settings for each table individually, and automatically restores the table-specific

settings whenever you switch tables. It has some additional special features for pin cabs, such as "night mode" (to reduce volume across all tables for late-night play) and individual level controls for multiple sound cards, all accessible from cabinet buttons.

You can find the download link and installation instructions on the PinVol page.

Game selectors, or "front ends"

When your pin cab is finished, you'll probably want it to give the appearance of being a full-fledged arcade machine, not a plain old Windows PC. When you turn on the power, you won't want to see the Windows desktop at any point; you'll want something that looks more like a video game instead. It's also important to be able to operate all controls with the basic set of pin cab buttons - flipper buttons, Start, Exit.

This can all be accomplished with a program known on the forums as a "front end", so-called because it's the first thing you see when you walk up to the pin cab. A front end program serves as a replacement for the Windows desktop. It provides a video game-style user interface that lets you browse through your installed tables, launch tables, and switch between tables. A good front end will let you operate everything with the pin cab buttons so that you don't have to reach for the mouse or keyboard.

The most widely used front end currently is PinballX, which is free but closed-source. The original front end, HyperPin (also free-but-closed-source), is still around, but it's not very widely used any more; most people consider PinballX's user interface to be more modern and more pin-cab-friendly. There are also two newer options: PinUp Popper, another free/closed-source program; and my own PinballY, free and open-source.

PinballY

This is my own project, brand new in late 2018. I tried to make it easy and quick to set up so that you can try it out without a lot of hassle. It's designed specifically for pin cabs, and has built-in integration with most of the pin cab ecosystem, including DOF, real DMD devices, joysticks (for button input), and multiple monitors. It's also highly customizable via a built-in Javascript scripting engine.

Downloads and more information are available at the PinballY Project Page.

PinballY is similar to the other front ends in terms of user interface appearance and functionality. The main reason I wrote it was that I wanted an open-source option (all of the other front ends I know of are closed-source).

PinUp Popper

This is a newer program released in early 2018. It's free, but closed-source. See www.nailbuster.com/wikipinup/doku.php for download and install information.

PinballX

PinballX is currently the most popular front end for pin cabs. It has a minimalistic user interface that's well designed for pin cabs, letting you access all functionality with just four buttons (flippers, Start, and Exit), but also letting you use other buttons if you have them (e.g., MagnaSave).

You can download PinballX from its home site, pinballx.com. It's free to download, but it's closed-source, and installed versions "expire" after a period of time, requiring you to update. Follow the Download link from the main page to download the installer.

After running the installer program, you have to run the **Settings.exe** program in the PinballX folder. PinballX needs to know a bunch of things about your system before it will work properly. You should go through at least the Basic settings. Pay particular attention to the following:

- Display Settings page: Assign the monitors you're using for the playfield (which PinballX calls the "main display"), backglass, and DMD (dot matrix display). Also set the rotations.
- Startup Settings: Set "Start with Windows" to Yes if you want the program to launch automatically when you boot the system.

- Keyboard Input Settings: set the key assignments to match the keys assigned to your cabinet buttons. If you're mapping the buttons to joystick buttons, you can assign those on the next page, Joystick Input Settings.
- Future Pinball, Visual Pinball: Set the directory paths for these programs. The "Working Path" field should be set to the folder containing each program.

Adding tables to the PinballX menu

PinballX doesn't go out and find your tables by itself. You have to enter each table into PinballX's menu list yourself. You do this using the Game List Manager program in the PinballX program directory. Before running this, make sure you configured the directory locations with the PinballX Settings program as described above.

The PBX installer will pre-populate the menu list with a few games for demo purposes, so the first thing you'll probably want to do is delete these. Simply click the Delete button next to each game in the list. Note that there are multiple game lists (Visual Pinball, Future Pinball, MAME), so you'll have to select each list with the drop list at the top of the window and delete its games.

Each pinball game you set up has a bunch of associated "media" items: a "wheel" image, which provides the title graphic shown in the menu when you navigate to the table; a playfield image; a backglass image; a DMD image; the advertising flyer for the game; an instructions card; video versions of the table and backglass images; and audio to play when you launch the game. You can set up each of these items individually, but that's extremely tedious, especially if you have lots of games to add.

Fortunately, there's an easier way.

The quick way to set up a game is to use the "Import Media Pack" button at the top. This lets you add a game, along with all of its related media items, in one operation. You'll still need to select the game's playable file (the .vpx file for VP 10, for example), but everything else will be set up automatically.

To set up a game using the "import" button, start by downloading the game's media pack. You can find media packs on vpforums. Select "Frontend Media & Backglass" on the navigation bar, then click "Complete Media Packs" under the Media Packs section. This will take you to a gigantic list of "HP Media Pack" files. The "HP" is for HyperPin, but PinballX knows how to read these same files. Navigate through the list to find the game you're looking for.

Each of these "HP Media Pack" files is an ordinary ZIP file. Don't unpack them. Simply download them to the Tables directory for the appropriate pinball player version. For example, if you're setting up a Visual Pinball 10 game, download the corresponding table pack to the Visual Pinball 10\Tables folder.

Now go to the PinballX Game Manager. Select the list for the appropriate pinball player at the top (e.g., select "Visual Pinball"). **Don't click Add Game** at any point. Instead, click Import Media Pack. Select the ZIP file you downloaded. This will automatically create a new entry for the game and populate it with the media items in the ZIP file. Now click on the Select button next to the Game field for the newly added item. Choose the playable game file from the list. Note that this will only show you a list of game files you've already installed in the Tables folder, so you'll have to actually download the game into the Tables before you can complete this step.

After you exit out of the Game Manager program and restart PinballX, you should now see the newly added game show up in the menu.

As you add tables to your system, you'll need to repeat this process for each one.

HyperPin

HyperPin was the original front end for pin cabs. It's an offshoot of the similarly named HyperSpin, which is a popular front end for home-brew video game cabinets. Since HyperPin came from the video game world, it was designed around an assumption that you have a big bunch of buttons. Pin cab builders tend to prefer a more minimalistic approach, with only a small set of buttons closer to what's found on most real pinball machines. This has always made HyperPin a little ill-fitting on a pin cab, since its UI depends on having a fairly large number of buttons that can be mapped to individual functions. A lot of early pin cab builders designed their cabs specifically for HyperPin by installing four or five extra buttons on the front panel dedicated to front-end functions. But most of us don't like the extra buttons on

aesthetic grounds, because they take away from the real pinball look. That's a big part of why so many pin cab builders migrated to PinballX when it became available.

The home site for HyperPin is hyperspin-fe.com. Click the Download button in the main navigation bar, then look for "HyperPin" in the Category list.

Where to find tables

Visual Pinball tables: The biggest collection I've seen of VP cabinet-mode tables is vpforums. Click "Visual Pinball Tables" in the navigation bar at the top. The popup menu has several sections; the ones you'll want to look in for pin cab use are "VP9 Cabinet Tables" and "VPX Tables" section. VP 9 requires tables to be designed specially for cabinet use, which is why it has a special section. VP 10 unifies cabinet and desktop modes, so it doesn't have a separate cabinet section - any VPX table should work in cabinet mode.

vpuniverse also hosts VP tables, although their collection isn't as extensive. Click the Downloads link in the navigation bar to find tables.

Future Pinball tables: As with VP 10, all Future Pinball table files are playable in cabinet mode. You just have to adjust the camera settings for each table to get it lined up properly for cabinet play. vpforums has a large collection of FP tables: click "Downloads" in the navigation bar, then look in the "Future Pinball Tables" section.

Backglasses: Some tables include the B2S backglass files with the Visual Pinball table files, but most don't, so you'll usually have to download backglass files separately. vpforums has a large collection of these: click "Frontend Media & Backglasses" on the navigation bar, then select "dB2S Animated Backglasses" under the Backglasses section.

PinballX & HyperPin media: vpforums has a large collection of media packs for the front-end menu program. Click "Frontend Media & Backglasses" on the navigation bar, then select "Complete Media Packs" from the "Media Packs" section. "HP Media Pack" files work in both HyperPin and PinballX.

17. Backglass Software Setup

If your pin cab has a separate backglass monitor, you'll want it to display the appropriate backglass artwork for the current game. And you'll want this to be more than just a still image, since the backglass is an active part of many games, showing information on score, bonus features, etc.

Fortunately, this is well supported in Visual Pinball. VP can display live, animated backglass artwork that synchronizes with the game action. VP requires an add-on program called B2S Backglass Server to do this. B2S works alongside Visual Pinball to display the animated backglass artwork, simulating the same backglass lighting effects, score displays, and animated elements that you'd see on a real machine. B2S is specifically designed for a cabinet setup where you have a separate monitor for the backglass.

To get all of this working, you have to install the B2S software itself, and then there are some extra setup steps required for each table. This chapter explains how to set up the B2S software and configure tables to use it.

B2S Installation

The first step to getting backglasses working is to install the B2S software itself.

If you installed Visual Pinball using the VP Installer program, B2S should have been installed automatically as part of the setup process, so you're already set.

If you didn't use the VP installer (that is, you installed VP manually from ZIP files or something like that), you'll have to install B2S separately. To find the download:

- Go to vpforums.org
- Click **Getting Started** the top navigation bar
- Select **Essential Files** > **Frontends and Addons**
- Find "B2S Backglass Server" in the file listing

There might be several versions of B2S Backglass Server in the file list. I'd recommend picking the one with the highest version number to make sure you have the latest update. Click the link, which will take you to the download page. That should contain links for downloading the file and for installation instructions. Note that you'll have to create a vpforums account to download a file, if you don't already have one.

Click the "instructions" link on the download page and follow the steps. Here's the basic procedure:

- Unzip all of the files from the downloaded B2S ZIP file into your Visual Pinball\Tables folder.
- Right-click the file B2SBackglassServer.dll. Select Properties from the menu. Check for a message under the "General" tab saying something like "This file was downloaded from the Internet and has been blocked." If you find this, there should be an "Unblock" button. Click it. If you don't find any such message, no action is required.
- Right-click the application file B2SBackglassServerRegisterApp.exe. Select **Run as administrator** from the menu.
- Check the README.TXT file from the downloaded ZIP file for any additional instructions or notes for the version you downloaded.

Download backglass files

Okay, you've installed B2S, loaded up a game in VP... and didn't see any backglass artwork. That's because installing B2S isn't the last step. You also have to do a little extra work to set up each table. This is work that you'll have to repeat for each new table you download, but fortunately it's only a one-time job for each table.

Most VP table files are distributed without any backglass artwork. Some authors bundle the two together, but in most cases you have to download the backglass file separately.

Fortunately, it's fairly easy to find backglass files. vpforums.org has a large collection:

- Go to vpforums.org
- Click **Frontend Media & Backglass** on the top navigation bar
- Click **Backglasses** > **dB2s Animated Backglasses**

Files are listed by table title, so just find the title of the table you're looking for and click through the links to download the file.

Are B2S files and .vpt/.vpx files paired?

No. You **don't** have to find the exact matching B2S version for your table. Any B2S for a given title should work with any .vpt/.vpx game for the same title. E.g., you don't need a specially paired B2S for "Funhouse_NightMod_ToyMod_991_v26.vpt"; any B2S for Funhouse should work.

This is the whole reason that the table files and backglass files are usually distributed separately. The B2S files and VP table files are more or less independent in terms of their operation and design. A particular .vpt or .vpx file should work with just about any .directb2s file for the same table. There's no need to find a matching set for a particular version of either.

2-screen and 3-screen versions

Some of B2S backglasses have "2-screen" and "3-screen" variants. The difference is how the speaker/DMD panel graphics are handled:

- A 2-screen backglass includes graphics for the speaker panel as part of the backglass window. This is ideal if you have a single large monitor in your backbox and no separate DMD video monitor or real DMD device.
- A 3-screen backglass separates the graphics for the backglass area and speaker panel area into different windows, so that you can position the two areas separately on your physical monitor layout. This is ideal if you have a 1990s-style backbox, with a 16:9 monitor at the top, and a separate speaker panel with its own DMD video monitor or real DMD device.

In cases where both types are available, choose the one that matches your physical cab setup. But it's also okay if only one type is available and it's the "wrong" one for your cab. You can always use either format. The worst that happens with the "wrong" format is that the geometry will be a bit distorted because the proportions were designed for a different monitor layout.

Installing the .directb2s file

After downloading a backglass file, perform these steps:

- Unpack it from the ZIP/RAR container if necessary
- Put it in your Visual Pinball **Tables** folder (the same folder where you keep your .vpt/.vpx files)
- Rename the file so that it matches the name of the .vpt/.vpx file you want to use it with, but keeping the .directb2s suffix (e.g., if the table file is called `Funhouse_NightMod_v2.vpx`, rename the B2S file as `Funhouse_NightMod_v2.directb2s`)

It's essential to put the .directb2s file in the same location as the table file and to rename it to match the .vpt/.vpx file name. That's how B2S finds the file. If the name and location aren't matched like this, B2S can't usually locate the file and won't show any backglass artwork.

(B2S does actually have some "fuzzy matching" that tries to find a matching file even if the name doesn't exactly match, but in my experience, that does more harm than good. B2S's fuzzy matching usually just picks something random and wrong, and the fact that it's doing it at all makes it that much harder to figure out why the wrong file is getting picked. The only way to make B2S pick the right file reliably is to give it the exact same name as the table file.)

Enable B2S on each table

There's one more step before the backglass artwork will appear during a game: you have to enable B2S mode for each individual game. This is a one-time step, but you have to do it separately for each table.

Most VP 10 tables will automatically use B2S if present, so this step isn't usually required for VP 10. However, if you try a VP 10 table and it doesn't work, it might be an unusual case that requires the VP 9 procedure below.

Some later VP 9 tables, from 2016 and later, also will automatically work with B2S.

Given that most VP 10 tables and some VP 9 tables will work "out of the box" without any modification, the first step is to simply fire up the game in VP and see if the backglass appears. If so, you're all set with that game. If not, try this procedure:

- Open the table in the VP editor (see Chapter 19, Customizing VP Tables)
- Open the table script (VP 9: **Edit > Script** menu command; VP 10: **View > Script** command)
- Look for a line like this:

```
Const cController = 0   ' 1=VPinMAME,
                         ' 2=UVP backglass server,
                         ' 3=B2S backglass server
```

- Note that the variable name `cController` might be slightly different, so just look for something that roughly matches that format.
- If you find such a line, change the "0" to the number listed for B2s (usually 3). Save and run the table. If it successfully displays the backglass, you're set.
- If you can't find the code above, look for something like this:

```
Set Controller = CreateObject("VPinMAME.Controller")
```

- If you find that, replace it with this:

```
Set Controller = CreateObject("B2S.Server")
```

- Save the game and try again.

If none of the above helped, and the table is a re-creation of an older EM (electro-mechanical) game from the 1970s or earlier, the game will need more extensive modification to make it work with B2S. The work needed is beyond the scope of this chapter. Your best bet might be to contact the author of the table and request a B2S-capable update, or see if someone else on the forums wants to take it on.