

ITFreeTraining



Installing Devices

For the free video please see
<http://itfreetraining.com/ap/1d40>

In this video from ITFreeTraining, I will look at installing devices. Nowadays, the process is a lot simpler than it used to be, so unless something goes wrong, installing a new device should go smoothly.

Installing the Device

- Install the card/device
- Hopefully plug and play installs the device driver
- Recommend: Update the device driver
 - Check manufacturer's web site
 - Should be the most up to date

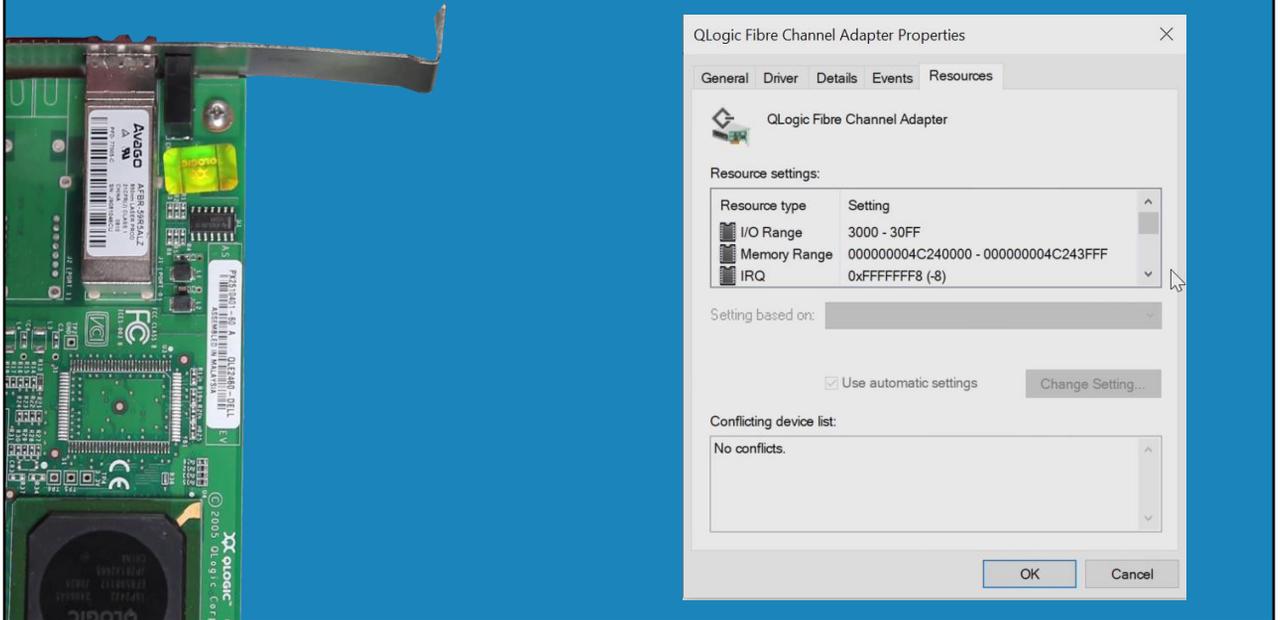
0:13 In this video I will install an expansion card; however, the process is much the same regardless of what you are installing. The first step is to install the card or the device. If the device does not support hot swapping, you will need to switch the computer off first. Some systems support hot swapping of expansion cards; however, this is not very common.

Once installed, hopefully plug and play will install the device and the associated drivers. Nowadays, plug and play is pretty good and in a lot of cases you will have a device driver available or you can download one if required. It is recommended to update the device driver to the latest version. The operating system will not always be able to get the latest driver. The best place to look for new device drivers is the manufacturer's web site, and these should be the most up to date ones.

In some cases, you may find that the hardware is supplied by someone else and the manufacturer uses it to build the device. For example, Nvidia sells the microchips, other companies then build the boards and circuits they go on. In cases like these, the most up to date driver will be the company that supplied the underlining hardware, and not the one who manufactured it. You may find that this is the case with chipsets. For example, if a motherboard has an Intel chipset, the Intel web site may have a more up-to-date driver than the manufacturer's web site. If you are not sure, always go to the manufacturer's web site first, as they should be able to supply you a device driver that will get the device working.

Without further ado, I will now change to my testing computer running Windows 10 to have a look at how to install a new device and the device drivers for that device.

Demonstration



1:58 In this demonstration, I will install a QLogic fibre card into this computer, and then install the required device drivers for the card. You can see the fibre card to the left of the computer. When I have a closer look, I can see a sticker on the card with the model number QLE2460.

These model numbers can be difficult to find or are printed on a chip where the writing has faded and so cannot be read. If you cannot find it, I will look at a way you can find out this information.

Next, I will plug the fibre card into the computer.

Now that the card is installed, the next step is to start the computer up and log in. Once I am logged into Windows, I will right click on the start menu and select the option "Device Manager". Once Device Manager has opened, this will show all the currently installed devices on your computer. Hopefully, plug and play will install your device automatically; however, in this case the device has not been installed automatically.

You will notice that the new device has appeared under the section "Other devices". Whenever a device is detected, but Windows does not have a device driver for it, it will appear under this section. It is important to check this section for devices, particularly after installing Windows. If you find a device in here, locate a device driver for it and install it, and then the device will move out of this section to another section.

To get some more information about the device driver, I will right click on it and select the

option "Properties". This will show a lot of information about the device. This includes information about which slot it is installed in, which device number it is and device driver information. In this case, no device driver is installed, so no information about the device driver can be displayed.

To install a device driver, you can press the button "Update Driver". This option will perform a search for the missing device driver. In this case, I know this search will not be successful, so before I can install the device driver, I will first need to download it.

The sounds like an easy task; however, in some cases, you may not even know what the device is. There may not be any distinctive marks on it or any way of identifying the device. Part of the way plug and play works is that every device that has ever been made has a unique hardware ID. If you don't know what the device is, we can use this hardware ID to work out what the device is.

To find out the device ID, I will select the tab "Details". This tab gives us a lot of information about the device. To get information to help identify the device, I will select the option "Properties" and then select "Hardware IDs".

There will be a number of hardware IDs shown. Starting from the bottom, you will find the least specific ones. For example, the first hardware ID will indicate the manufacturer and the type of card. As we go up it will become more specific giving manufacturer and version number of the card.

Generally, you want the top one, but if you are not having any luck you could try one of the others. Rather than writing this down, I will right click on the top hardware ID and copy it to the clip board.

In this case, I know what type of device it is because I could physically look at the card. However, if you can't physically look at the device, for example the device is embedded in the motherboard, having this device ID can really help.

The next step is to work out what device it is. To do this, I will open Chrome and perform a Google search for the hardware ID. The results will hopefully give you some idea what the device is. You will notice that the results are a number of different sites which have downloads for the device.

A word of caution here; there are many web sites out there that will let you download device drivers. Personally, I would not trust these and I would only trust the manufacturer's web site. You don't know if these others have added something to the device driver; but if it is an old device and the manufacturer no longer exists, this may be your only option.

You will notice that the second option is a QLogic Fibre Channel Adapter. We just got a really good hint at what the device driver is. The next step is to go to the QLogic web site and get the device driver from them. To find the site, I will Google QLogic followed by download to find where to download the device drivers.

You will notice the first result is the QLogic web site. I trust this site enough to download device drivers from it, so I will select it.

Once the website is open, I will select the option for “Search by operating system” and the option for “Fibre Channel Adapters”. In some cases, there won’t be a device driver for the operating system you are using. Starting with Windows Vista, Microsoft started developing a good specification for device drivers that did not change too much after that. This essentially means that an older device driver will probably work on a newer operating system. In the old days, this was not so much the case. This also means that newer device drivers may work with an older operating system.

Once I have selected the device driver, I next need to press “Agree” to accept the license and the device driver will be downloaded.

The device driver will be downloaded to the local computer. I will now open the downloads folder, and open the zip file containing the download. You will notice that there is an executable file. I can run this, and the device driver will be installed. In some cases, the executable will run a setup and install additional software with the device driver. In some cases, you may want this additional software and in other cases you may just want to install the device driver.

It is pretty simple to install the device driver using the executable, so I will look at how to install the device driver manually. Sometimes you may want to do this, as the executable may also install additional software that you do not want.

Since the device driver is zipped up, I will first decompress it to access the files inside. To do this, I will right click on the file and select “Extract All”.

Once the files have been extracted, I next want to update the device driver using the files from this location. So that I don’t have to remember the whole path, I will select it and copy it to the clipboard; I will next go back to Device Manager, right click the device, and select the option “Update Driver”.

From the update device driver screen, I could select the first option to “Search online for an updated device driver”. This is often a good first step in the troubleshooting process because if it downloads a device driver, you will then know what the device

is. However, in this case, I know the device driver will not be found online, so I will select the second option "Browse my computer for driver software".

The next step is to paste the path that I copied previously for the driver location. There is an option ticked to include subfolders. I recommend leaving this option ticked as it is common for a manufacturer to have their setup files in the root folder and the actual device drivers in a sub-folder.

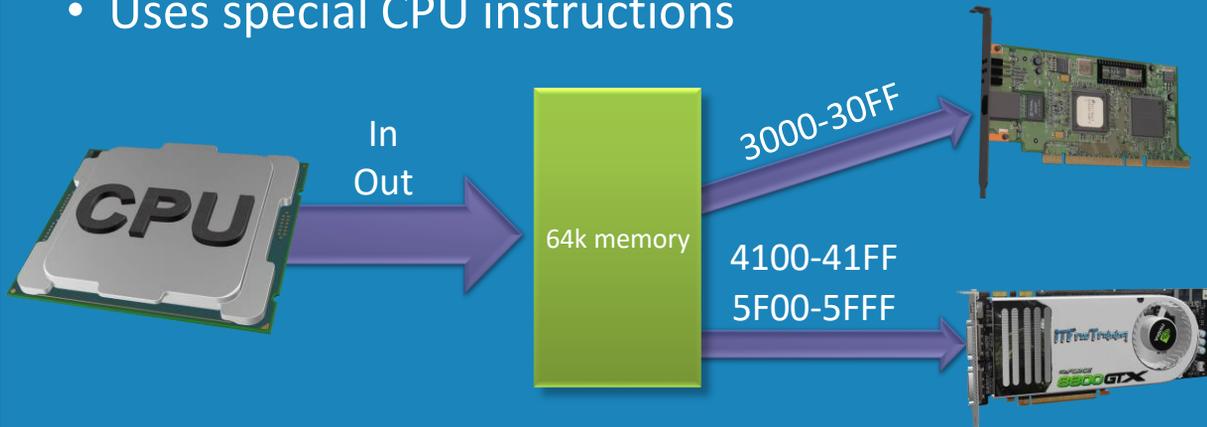
Once I press next, Windows will install the missing device driver to my system. Once complete, you will notice that, in Device Manager, the device has moved from "Other devices" to "Storage controllers". To find out more information about the device, I will right click on the device and select the option "Properties".

You will notice that the device is now reported as working properly. There are also other tabs you can select to get more information about the device driver; for example what version it is. In this case, I want to find out what resources the device driver is using, and to do this I will select the last tab "Resources".

You will notice that when I scroll through the resources, there is an I/O range, memory range and a lot of interrupts. These are all allocated automatically by plug and play. In the old days, you would sometimes get problems with certain devices and conflicts; however, nowadays plug and play is pretty reliable, so you don't generally get problems like that anymore. Given the three different resources listed here, I/O range, memory range and interrupts, I will now have a closer look at what each of these do.

I/O Range (Port Mapped I/O)

- Limited to 64 kilobytes across all devices
- Uses special CPU instructions



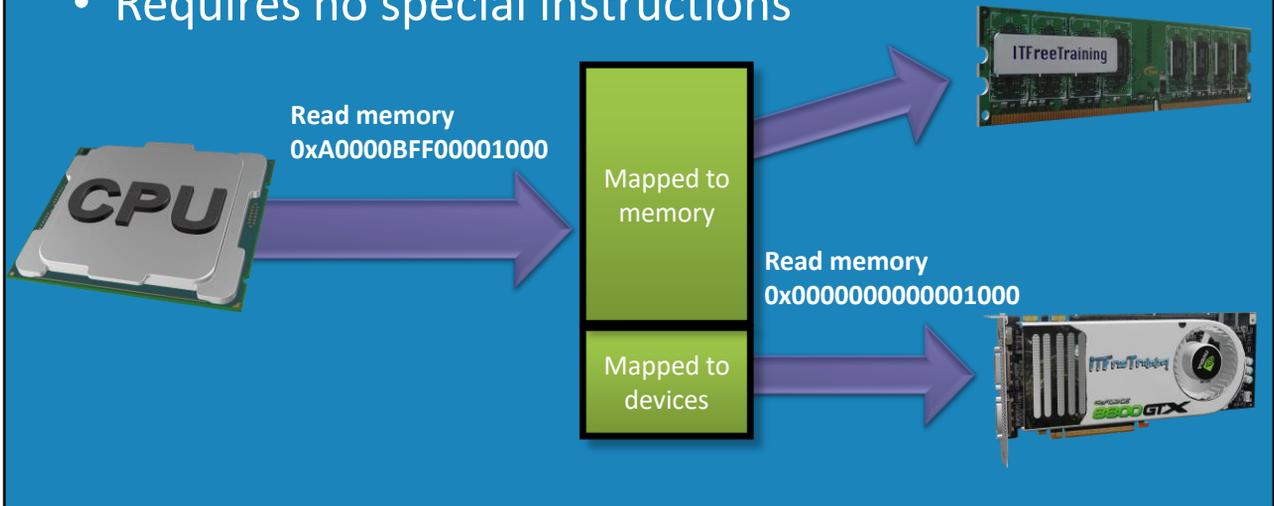
10:51 The first one that I will look at is the I/O range. For the CompTIA exam you don't need to know much about these, but I think it is good to have an understanding of how a computer works, so when you see an I/O range in a device driver you know what it is referring to.

The I/O range is limited to 64 kilobytes which is shared across all devices on the computer. So, each time a device is added, part of this 64 kilobytes can be allocated to that device. I say "can be" because it is up to the device to request what it needs. Plug and play will look at the request from the device and decide how much of the I/O range it will allocate to the device. There are limits to how much a device can request and devices generally don't need too much I/O range to operate. Considering this, it is unlikely the computer will run out of I/O range even with a large number of devices. Again, this is another thing in modern computers that looks after itself and you generally don't need to worry about it.

In order for the computer to access this I/O range, it uses special CPU instructions. These were added in the first IBM PC and are still used today. In those days, using the I/O range was a good simple way to access the device. Accessing memory on the device, although not impossible, was not as easy as it is today. I/O range is not a bad system for sending commands to a device; however, due to its limited size, it is not good for transferring a lot of data to the device. To get around these problems, let's have a look at how memory transfers are handled.

Memory Range (Memory Mapped I/O)

- Device memory directly accessible by CPU
- Requires no special instructions



12:32 Next, I will look at the memory range. A memory range allows a device's memory to be directly accessible by the CPU. If I consider a CPU, the CPU essentially accesses memory by using an address. This is where it gets interesting, because the memory address used can be mapped to a completely different physical location than the computer's memory.

So, if you consider the memory addresses the CPU has access to, the physical location of the memory could be anywhere. For example, it could be in the memory inside the computer or memory on a device in the computer.

Since the memory is mapped, the CPU accesses memory in the device just like it would any other memory in the system. Thus, the CPU requires no special instructions. For example, if the CPU wanted to access memory on the video card, it would access the memory range that was allocated to the video card. The memory controller knows this memory is on the video card and reroutes the request to the video card (in the process, changing the memory address to the physical memory address of the memory on the video card).

This process is all done seamlessly. As far as the CPU is concerned, when it is accessing memory on the video card, it is no different than when it accesses memory in the computer. The process and instructions are the same.

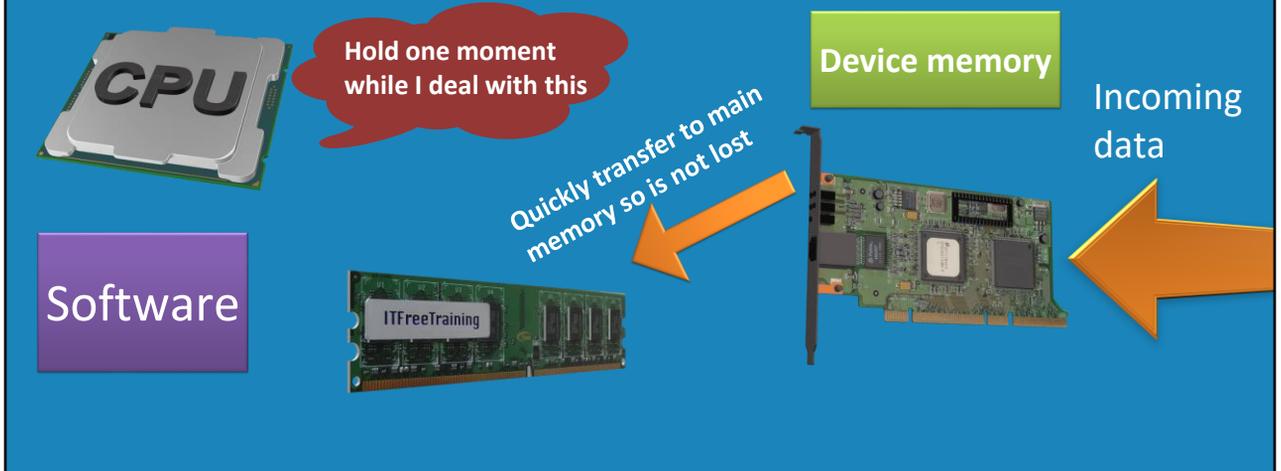
So, the question remains, should a device manufacturer use I/O range or memory range? Essentially both methods allow data to be sent or retrieved from the device. There are two differences: I/O range requires its own instructions in order to access a device and is limited to

64 kilobytes of memory. Memory range on the other hand is not limited to a small amount of memory and does not require special instructions in order to access a device. At the end of the day, it is up to the manufacturer of the device to decide which one they want to use. Generally, you will find that most devices will use a combination of the two.

Now that we can transfer data to and from the device, we next need a way to control when this occurs, and this is where interrupts come into play.

Interrupts (IRQ)

- A signal to the CPU immediate attention is required
- CPU will pause what it is doing to action the interrupt



14:46 An interrupt is essentially a signal to the CPU that immediate attention is required somewhere else. When this occurs, the CPU will pause what it is doing and action the interrupt. To understand this process better, it is best to consider an example.

Let's consider that you have a network card. The network card is connected to the network and thus data is being transmitted to the network card. On the network card there is some memory that the incoming data can be stored in.

The network card has a limited amount of memory on it, and if that memory fills up, the network card either has to discard the data it has previously received or discard the new data. Either case involves losing data, which is not desirable.

You may be thinking that an easy solution would be to increase the amount of memory on the network card. This would help prevent data being lost. However, the thing to consider is that if data has arrived at the network card, generally you want to know about this sooner rather than later. If there is a delay before the data is actioned, this will increase the latency of your applications.

So, the solution to this problem would be to poll the device to see if there is data waiting or use an interrupt. Polling essentially means checking the device periodically to see if data is waiting. For example, consider you are waiting for a package. You could go out every few minutes and see if the package has arrived. This would soon become a waste of your time. Also, there will be a delay from when the package is delivered to when you go and get it. The

way to decrease this delay would be to check more often, which then becomes an even bigger waste of your time.

The simple solution is to install a doorbell. When the package is delivered the delivery driver can press the doorbell to let you know. When this occurs, you stop what you are doing to get the package; you action the request right away and don't waste time checking to see if it has been delivered. This is an example of how an interrupt would work in a computer. If you consider a device like a mouse, when it moves you want to know right away. Any delay and the movement data is no longer useful as potentially the mouse has moved elsewhere. When the mouse is not moving, you don't want to know about it because you can assume it to be in the same place—this is a perfect example of a good use for an interrupt.

Considering the network card example, the CPU will work away executing code. When the network card has data, it will signal the CPU by using an interrupt. The CPU will pause what it is doing in order to action the interrupt.

In this particular case, the CPU will transfer the data from the network card to the main memory of the computer, thus freeing the network card up to receive more data. In many cases, the interrupt will also perform some processing of the data. In the case of the mouse, the input from the mouse will be converted to an x and y position and be stored in memory for the operating system to use. In the case of the network card, the packet data will most likely be converted to a data stream, and the software will be notified it is ready to be processed. The point being, an interrupt needs to be responded to quickly and be completed quickly. The CPU has paused executing programs on the computer in order to process the interrupt. The program does not want to wait too long, so the interrupt code does what it needs to do quickly and does not do more than it needs to.

As before, interrupts are allocated by plug and play so you don't need to worry about them. In the old days of computing, there were only 15 usable interrupts. In modern computers there is no limit like this. You will find that some old devices like serial ports will still use these legacy interrupt numbers, however for new devices the interrupts could be any number.

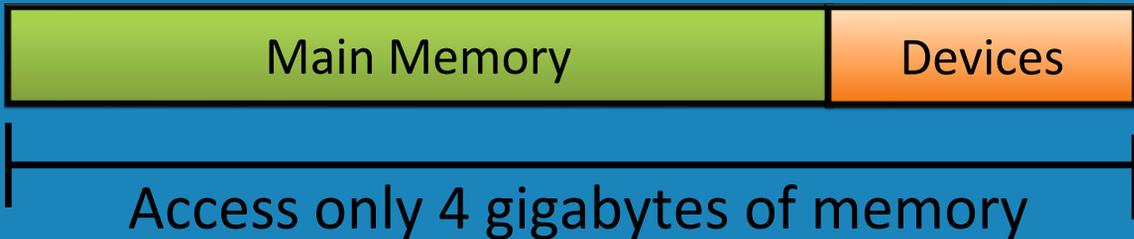
If you are having performance problems on a computer, this is probably a sign that there may be a hardware problem. This can show itself in two different ways: the CPU is spending a lot of time processing an interrupt, remembering that interrupts should be completed quickly, or the CPU is processing a very large number of interrupts. In either case, there may be a hardware fault causing these problems.

On modern computers, plug and play should just work. Plug and play has come a long

way since the early days, and the resources a card needs to operate should be allocated automatically without the administrator having to do anything. In some cases, you may be working on an old system, so I will next have a look at some of the problems you may encounter. For the CompTIA exam it is unlikely they will ask you about this; however, it is good to know if you get stuck fixing an old computer.

32-Bit Memory

- Maximum 4 gigabytes of addressable memory
- Shared between devices and memory
- Not all of 4 gigabytes of memory will be accessible



19:31 The biggest problem you may come across is with 32-bit systems, as these systems are limited to four gigabytes of addressable memory. This addressable memory is shared between devices and memory.

If you are using up to about three gigabytes of memory this should not present any problems. However, if you have four gigabytes installed, not all of the memory will be accessible. This is because the memory of the devices will need to be mapped to some of the addressable memory. Essentially each spot in memory needs to go somewhere. It either goes to main memory or it goes to a device, it can't go to both. This is why you won't be able to access the full four gigabytes of memory on a 32-bit system.

In some cases, some devices will allow you to reduce the amount of addressable memory they use. These settings are usually configured in BIOS. The problem with this is the less memory that they can access, the slower the device will operate. Therefore, it comes down to a tradeoff between the amount of main memory you can access and how much will be available to devices. A lot of devices don't require much memory, but devices like video cards require more, and this is where you are likely to come across problems – all the more reason to try and get the computer upgraded to 64-bit.

That concludes this video on installing devices. I hope you have found this video informative. Until the next video, I would like to thank you for watching.

References

“The Official CompTIA A+ Core Study Guide (Exam 220-1001)” Chapter 3 Position
14046-14936

“MEMORY-MAPPED I/O VS PORT-MAPPED I/O – 2018”

https://www.bogotobogo.com/Embedded/memory_mapped_io_vs_port_mapped_isolated_io.php

Credits

Trainer: Austin Mason <http://ITFreeTraining.com>

Voice Talent: HP Lewis <http://hplewis.com>

Quality Assurance: Brett Batson <http://www.pbb-proofreading.uk>