

These notes describe how to use the USB configuration program to program the EEPROM for operation as an N2PK VNA USB interface.

Below are the step by step instructions followed by some notes on the program.

NOTE THAT we here assume that you are using XP. This process has been checked on XPpro and on a clean installation of XPhome. The drivers should load and run correctly on the 32 and 64 bit versions of XP, Vista and Windows-7.

My thanks to Ian GM3SEK for taking the time to put together the first version of these instructions and to Paul N2PK for pointing out that it would be a useful utility. Finally thanks to Ivan for procuring the code signing key for the signed drivers.

There are two sets of drivers – the original unsigned drivers that use ezusb.sys and n2pkvld.sys and the newer signed drivers that use cyUSB.sys.

New signed drivers

Vista-64 and Windows-7 64 bit require drivers to be signed. In making the change to signed drivers the opportunity was also taken to move to the newer cyUSB suite of drivers from Cypress.

The older unsigned drivers use a two part driver comprising an initial download driver, n2pkvld.sys, whose sole purpose is to load the VNA specific code into the usb controller chip and a second driver, ezusb.sys, which then provides access to the vna functionality. There are multiple versions of n2pkvld.sys for the different versions of hardware and each has its own inf file to facilitate loading the drivers.

The new drivers use just one driver, cyUSB.sys. This driver knows how to download a script file to the controller chip so is used for both the initial loading of firmware and subsequent operation of the device. There are four versions of the driver, two for XP and two for Vista/Windows 7 and in each case a 32 bit and a 64 bit driver. You do not need to worry about this; the inf files will automatically select the correct file. There are also several script files that contain the device firmware; again you do not need to worry about this as the correct version will be automatically selected. The only thing you need to choose is the directory where the drivers are located – XP or Vista-Windows7.

The signing of the drivers was kindly performed by Ivan. Signatures have been applied to the catalogue (.CAT) and driver (.SYS) files. The signatures on the CAT files each sign four files, their respective inf file, the two referenced sys files and the corresponding spt file.

Along with the new drivers is a new version of USBconfigure which understands how to communicate with the new drivers and a new version of vnadll.dll. To use the new drivers you MUST use the new dll (it understands both new and old drivers). The od

vnadll.dll is less than 40k in size, the new driver over 80k if you want an easy way to check which one you are using.

STEP BY STEP (old unsigned drivers)

Do not connect the VNA to the USB interface until you have successfully completed step 13!

1. Unzip the driver files into a suitable location. This will install the 'usb configure.exe' program and also create driver subdirectories called 'ezusb' and 'N2PK drivers'.
2. Plug in the USB card. Windows will identify there is a new USB device, and may then search for several minutes to find drivers; you have to wait until it gives up. (If you see any messages about "This device will go faster", ignore them and they will go away. See step 10 for more details.)
3. When the "Found New Hardware Wizard" eventually appears:
 - 3.1 Choose "No, do not connect to Windows Update" and click Next.
 - 3.2 Choose "Install from a list or specific location" and click Next.
 - 3.3 Choose "Search for the best driver in these locations", check "Include this location" and Browse to the 'ezusb' subdirectory created in step 1. Click OK and then click Next.
4. The USB drivers are installed. Ignore the "EEPROM missing" error message and click "Next". It will then tell you that the drivers have not passed Windows Logo testing. Ignore this and click "Continue Anyway" (all this means is that the chip has been found correctly and either the EEPROM is missing or it is blank). When the process completes, click the "Finish" button. You are returned to the Windows desktop and the "Found New hardware" flag pops up at the lower right.
5. Go to the directory you created in step 1 and double-click the 'usb configure.exe' program. If you get a "Compressed Files warning", click "Run". The "N2PK USB Interface Control" dialog opens.
6. Go to the "Setup" tab and use the "Browse" button in the "Vend_Ax filename" field to find the 'Vend_Ax .Hex' file in the 'ezusb' subdirectory. Click "Save Settings".
7. Go to the "Program FX2 Chip EEPROM" tab and click "Download". Watch the messages at the bottom. If the download succeeds, the final line will say "SUCCESS: Program downloaded" and it will enable the "Read" and "Write" buttons on that tab. Click "Read" and you should get 8 bytes of FF. At this point you are communicating correctly with the USB card.
8. In the "Write EEPROM" field of the same tab, select the "N2PK" option from the drop-down list and click "Write Data". That should result in a success message, and the EEPROM will have been programmed. Now click the "Read" button and check that it

reads back the same string that was programmed in (see "Data to be written" at bottom of tab).

If the EEPROM does not read back correctly, go to the drop-down list, select the "Erase (write FFs)" option and repeat the "Write Data" operation to (try to) reset the EEPROM. Do not pull the USB plug out while the EEPROM is not reading back correctly, or you will lose control of the EEPROM.

When all is OK in step 8, close the 'usb configure.exe' program. Note that at this point the program running on the USB controller chip is the "Vend_Ax" programming utility and not the VNA program.

9. Pull the USB plug out and reconnect it. When the "Found New Hardware Wizard" appears:

9.1 Choose "No, do not connect to Windows Update" and click Next.

9.2 Choose "Install from a list or specific location" and click Next.

9.3 Choose "Search for the best driver in these locations", check "Include this location" and Browse to the 'N2PK drivers' subdirectory created in step 1. Click OK and then click Next.

If the wizard does not appear, open Control Panel, select the "System" Icon and from there select the "Hardware" tab. Then click the "Device Manager" button. Select the first item in the list and from the Action menu select "scan for hardware changes". This should then cause Windows to find the device.

10. The N2PK drivers are installed. Ignore the "Has not passed Windows Logo Testing" message and click "Continue Anyway". When the process completes, click the "Finish" button. You are returned to the Windows desktop and the "Found New hardware" flag pops up.

If the "Found New Hardware Wizard" appears again, repeat steps 9 and 10 until Windows can routinely find the USB interface when it is plugged in.

Note: from now on, whenever you plug the USB interface into a USB1 port (as used in older PCs), you will see an irritating message that "This device can perform faster". Ignore it.

Note 2: If the PC wants to reboot let it. The driver will in fact be working correctly but it is simpler to get it over with. You may also find that if you plug it in to a different USB port it will reinstall the drivers again. This should only happen once per port though.

11. Re-run the 'usb configure.exe' program (see step 5). At the "Compressed Files warning", click "Run". The "N2PK USB Interface Control" dialog opens.

12. Go to the "Communicate with Adapter" tab and click "Check VNA interface". You should get an "N2PK VNA Interface found" message, and the "Read VNA Interface" and "Write VNA Interface" buttons will be enabled.

13. Click "Read VNA Interface" - still with no VNA connected - and you should read the message '00 00 X0 C0 00'. Where X is 8 or 4.

Now proceed to connect the interface to the N2PK VNA. Follow the instructions in Win4VNA as to how to enable the use of the USB interface with the application.

STEP BY STEP (new signed drivers)

Do not connect the VNA to the USB interface until you have successfully completed step 13!

1. Unzip the driver files into a suitable location. This will install the 'usb configure.exe' program and also create driver subdirectories called 'XP' and 'Vista-Windows 7'.

2. Plug in the USB card. Windows will identify there is a new USB device, and may then search for several minutes to find drivers; you have to wait until it gives up. (If you see any messages about "This device will go faster", ignore them and they will go away. See step 10 for more details.). Note: it is possible that, if previously unsigned drivers were loaded or Windows was instructed not to install the drivers, that no such response will be obtained. In this case, run Device Manager, locate the entry for the interface and select "update drivers" manually.

3. When the "Found New Hardware Wizard" eventually appears:

3.1 Choose "No, do not connect to Windows Update" and click Next.

3.2 Choose "Install from a list or specific location" and click Next.

3.3 Choose "Search for the best driver in these locations", check "Include this location" and Browse to one of the subdirectories created in step 1 as appropriate for the operating system. Click OK and then click Next.

4. The USB drivers are installed. Ignore any "EEPROM missing" error message and click "Next". It will then tell you that the drivers have not passed Windows Logo testing. Ignore this and click "Continue Anyway" (all this means is that the chip has been found correctly and either the EEPROM is missing or it is blank). When the process completes, click the "Finish" button. You are returned to the Windows desktop and the "Found New hardware" flag pops up at the lower right.

5. Go to the directory you created in step 1 and double-click the 'usb configure.exe' program. If you get a "Compressed Files warning", click "Run". The "N2PK USB Interface Control" dialog opens. Make sure that the title of the dialog box indicates version 2.2 or later of the program.

6. Go to the "Setup" tab and use the Browse" button in the "Vend_Ax filename" field to find the 'Vend_Ax .Hex' file. Click "Save Settings". Click on the "Find devices" button and select the device indicated. Note that if you have any other Cypress controller chips plugged in at this point then it may find these as well; be careful not to reprogram them inadvertently.

7. Go to the "Program FX2 Chip EEPROM" tab and click "Download".

Watch the messages at the bottom. If the download succeeds, the final line will say "SUCCESS: Program downloaded" and it will enable the "Read" and "Write" buttons on that tab. Click "Read" and you should get 8 bytes of FF. At this point you are communicating correctly with the USB card. If you navigate away from this tab in the program at this time then you will find you can no longer communicate with the card until you reset it by disconnecting it and reconnecting it.

8. In the "Write EEPROM" field of the same tab, select the "N2PK" option from the drop-down list and click "Write Data". That should result in a success message, and the EEPROM will have been programmed. Now click the "Read" button and check that it reads back the same string that was programmed in (see "Data to be written" at bottom of tab).

If the EEPROM does not read back correctly, go to the drop-down list, select the "Erase (write FFs)" option and repeat the "Write Data" operation to (try to) reset the EEPROM. Do not pull the USB plug out while the EEPROM is not reading back correctly, or you will lose control of the EEPROM.

When all is OK in step 8, close the 'usb configure.exe' program. Note that at this point the program running on the USB controller chip is the "Vend_Ax" programming utility and not the VNA program.

9. Pull the USB plug out and reconnect it. When the "Found New Hardware Wizard" appears:

9.1 Choose "No, do not connect to Windows Update" and click Next.

9.2 Choose "Install from a list or specific location" and click Next.

9.3 Choose "Search for the best driver in these locations", check "Include this location" and Browse to the appropriate subdirectory created in step 1. Click OK and then click Next.

If the wizard does not appear, open Control Panel, select the "System" Icon and from there select the "Hardware" tab. Then click the "Device Manager" button. Select the first item in the list and from the Action menu select "scan for hardware changes". This should then cause Windows to find the device.

10. The N2PK drivers are installed. Ignore the "Has not passed Windows Logo Testing" message and click "Continue Anyway". When the process completes, click the "Finish" button. You are returned to the Windows desktop and the "Found New hardware" flag pops up.

If the "Found New Hardware Wizard" appears again, repeat steps 9 and 10 until Windows can routinely find the USB interface when it is plugged in.

Note: from now on, whenever you plug the USB interface into a USB1 port (as used in older PCs), you will see an irritating message that "This device can perform faster". Ignore it.

Note 2: If the PC wants to reboot let it. The driver will in fact be working correctly but it is simpler to get it over with. You may also find that if you plug it in to a different USB port it will reinstall the drivers again. This should only happen once per port though.

11. Re-run the 'usb configure.exe' program (see step 5). At the "Compressed Files warning", click "Run". The "N2PK USB Interface Control" dialog opens.

12. Go to the "Communicate with Adapter" tab and click "Check VNA interface". You should get an "N2PK VNA Interface found" message, and the "Read VNA Interface" and "Write VNA Interface" buttons will be enabled.

13. Click "Read VNA Interface" - still with no VNA connected - and you should read the message '00 00 X0 C0 00'. Where X is 8 or 4.
Now proceed to connect the interface to the N2PK VNA. Follow the instructions in Win4VNA as to how to enable the use of the USB interface with the application.

UPGRADING FROM UNSIGNED TO SIGNED DRIVERS

The current drives in both signed and unsigned packages are the same – there is no need to update. If however you want to then you can't simply use the "update" function from the drivers. To force it, delete (or preferably rename just in case) the file ezusb.sys and n2pkvld.sys files that are present in the windows\system32 or window\system32\drivers directory. When next you plug the board in it will not be able to load the drivers and will ask afresh. Now follow the above instructions for loading the drivers. Also note that it is possible that Windows will fail to prompt for the new drivers; in this case run Device Manager, browse manually to the interface, and manually install / update the drivers.

PROGRAM NOTES

Note that if you wish you can also send commands to the USB chip from the configuration utility but beware – it is easy to block the communications if you are not careful. If this happens just pull the USB cable out close the application and plug it in again.

The interface also allows you to download programs to the USB chip directory for test purposes. This is useful for testing variants of the FX2 chip application but note that if the USB cable is pulled out and reconnected the chip will revert to the current N2PK VNA firmware.

Also one other point. In playing with the interface it is possible that (for example after programming the EEPROM and disconnecting/reconnecting the chip it will be allocated a

different driver name (for example ezusb-1 instead of ezusb-0) You can find out if this has happened and change the name from the setup screen.

PROGRAM DETAILS

When the program starts it opens at the “communicate” screen. This is used to send messages to and read messages from the VNA. This means however that the VNA interface program must be running on the FX2 chip before this can be done. Therefore either the EEPROM must have been programmed and the driver automatically downloaded, or the download process has been used to put the program onto the chip. In either case, reading the interface string descriptor 2 will yield a string that we can recognize. Specifically the string we look for is set on the “setup” screen under the name “Configuration string for N2PK VNA”. If we find this, then the read and write buttons are enabled. When ‘read’ is then clicked, a message is read from the interface (see the N2PK USB firmware guide for details). When Write is clicked, whatever is in the Write area is sent to the chip. Any message may be entered here as hex, or the Set and Raw buttons may be used to create a message. Again, see the firmware guide for details.

One thing to note. It is dead easy to send a message that will pause communications, and I am not certain that there is not a bug here somewhere so if it locks up, disconnect the interface briefly.

The second screen is “Download Program”. This allows any program (any VALID program) to be sent to the FX2 chip. Just browse for a file (only FX2 HEX files are supported) and click on download. The idea is that this screen would normally be used to send new FX2 executables to the chip whilst testing new firmware.

The third screen is the “Program FX2 chip EEPROM” screen. It is pretty well described above. Format Data button is a null function. It just tries to get you to think about any non standard value that you are going to write.

Final screen is “Setup”. These are the 3 basic configuration items that may be changed by the user. The most likely to be changed are the driver name and the location of Vend_Ax.hex. When a device is plugged in, the ezusb.sys or cyUSB.sys driver allocates it a driver name. These are usually “ezusb-“ followed by a digit 0..9. or a string that starts ‘N2PK USB device’ or ‘cyUSB device’. The difference is as follows. If you are running the old unsigned drivers, then Windows will be using exusb.sys to communicate and the name will start with “ezusb”. If you are running with the signed drivers then cyUSB.sys will be used and one of the other two names will be used. For an unprogrammed eeprom the Cypress GUID of {AE18AA60-7F6A-11d4-97DD-00010229B959} will be used and the name will start “cyUSB device”. When programmed as an N2PK VNA USB interface then the VNA specific GUID of {C63859BD-5C4B-474d-9572-CE604E611D73} will be used and the name will start with “N2PK USB device”. So, be careful, this can trip you up. The name will change after you have programmed the eeprom.

Typically you will only ever see '-0' at the end of the name but there is one situation that can cause confusion. When you reprogram the EEPROM then unplug the device and plug it in again, ezusb.sys may allocate it "-1" thinking that "-0" is still connected. You can search for devices using the "find devices" button. Vend_ax is used to program the eeprom. It is an FX2 program that will accept requests to program the EEPROM and must be running in the FX2 before you can try to program the EEPROM. The setup program allows the location of this file to be set. The final item is the string that needs to be present in configuration string 2 of the FX2 in order for us to be assured that the USB interface program is running on the FX2. When "store" is clicked, the current settings are stored in the registry and reloaded next time the program runs.

The drivers that exist are as follows:

Ezusb.sys	The driver used to communicate with the board using the old unsigned drivers
N2pk*vld*.sys	There are several versions of this, the original unsigned driver. These run when the board is plugged in and download the operating firmware to the controller chip. The first '*' may either be absent or 'v5' to signify one of the original boards or the new v5 family VNA boards. The second '*' may be absent or take the value '500' to signify a low power board or one that consumes 500mA in powering the whole VNA
CYUSBxp32.sys CYUSBxp64.sys	These are the 32 and 64 bit signed drivers for XP. There is no separate pre and post driver as there was with the unsigned drivers.
Cyusb_vista_w7_32.sys cyusb_vista_w7_64.sys	These are the vista and Windows 7 versions of the above, again in 32 or 64 bit versions. Whilst the XP drivers may be used under Vista or Windows 7, the opposite is not true, and it is also preferable to use these drivers not the XP drivers under Vista or Windows 7.

The .spt files are script files that contain the hex firmware; the signed drivers download them automatically to the USB controller chip on startup.

The .inf files are the ones used by Windows to install the drivers; they contain instructions on how to install and load. There are separate versions depending on operating system and hardware configuration (original / V5 board).

The .cat files are catalogue files that contain details on the signatures of the files needed by an inf file, the drivers and the .spt files.

Dave
G8KBB