

How to Flash Firmware in the Turnigy/FlySky/Eurgle/IMAX 9x Transmitter

This guide is intended as a cookbook solution to flashing new or alternate firmware to the various labeled brands of the OEM FlySky 9x. One way to do it is presented, which works well. Alternatives are generally not shown to prevent confusion. If you follow these steps exactly, you should have good luck. If you do something else, this guide will be of only general help. Everything on the list is available from US suppliers, except as noted. I don't know about shipping to other countries.

Disclaimer

While I believe the instructions in this document are accurate, I make no guarantee that you will be successful, and won't turn your 9x into a door stop. Anything you do to your 9x based on this document is entirely at your own risk. The 9x and it's processor are tough but not indestructible. You've been warned!! See the section on Troubleshooting for further warning information.

WARNING: If you have never soldered before, and particularly if you have never soldered on a printed circuit board before, now is not a good time to start!! Have someone experienced do the soldering on the mainboard of your 9x. A guy with the screenname, "s_mack", is in the process of producing a solderless board to reprogram the 9x. It isn't available yet, but should be soon if you are leery about soldering.

Work in a static free environment. Semiconductors are sensitive to static discharge. Use a grounded soldering iron (see requirements) to prevent accidental damage from that source. You are directly connecting, in some instances, to the microprocessor in your 9x.

Requirements

Hardware

You will need the following:

Turnigy/FlySky/Eurgle/IMAX 9x Transmitter

Windows based computer with Windows 2000 or higher and one free USB port

AVR Pocket Programmer, available here: https://www.sparkfun.com/commerce/product_info.php?products_id=9825

It looks like this:



OR,

USBASP AVR Programmer (buy this one, **ESPECIALLY** if you are using 64 bit (x64) versions of Windows Vista or Windows 7):

Available for worldwide shipping from Australia:

http://www.protostack.com/index.php?main_page=product_info&cPath=23&products_id=24

The **same** programmer is also available on ebay, here:

http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=330457885370&ssPageName=STRK:MEW NX:IT#ht_2802wt_835

It looks like this:



The following programmer is available in Europe. I have not used it, but have been told it works well, but this guide does not cover the specifics of using it:

<http://shop.myavr.com/index.php?sp=article.sp.php&artID=200006>

USB cable like this for the Pocket Programmer:

http://www.monoprice.com/products/product.asp?c_id=103&cp_id=10303&cs_id=1030302&p_id=3896&seq=1&format=2

Since the Protostack/ebay programmer plugs directly into your USB port, I **highly** recommend getting a **male/female usb cord**, so you don't accidentally bend/break your programmer or usb port while plugged into your computer:

http://www.monoprice.com/products/product.asp?c_id=102&cp_id=10303&cs_id=1030304&p_id=4885&seq=1&format=2

15 to 30 watt GROUNDED pencil soldering iron like this:

<http://www.radioshack.com/search/index.jsp?kwCatId=&kw=15%20watt%20soldering%20iron&origkw=15%20watt%20soldering%20iron&sr=1>

Rosin core solder

This cable/connector combination: http://search.digikey.com/scripts/DkSearch/dksus.dll?lang=en&site=US&WT.z_header=search_go&lang=en&site=us&keywords=awp10-7241-t-r&x=0&y=0
This is a color coded flat cable/connector combination which makes it really easy to figure out which wire is which. They show as zero in stock, but they make them up custom and ship usually in one day.

Cable connector: http://search.digikey.com/scripts/DkSearch/dksus.dll?WT.z_header=search_go&lang=en&site=us&keywords=awp10-7241-t-r&x=0&y=0
This is used to cap the above cable when not in use.

Low temp hot glue gun and low temp glue gun sticks. I bought mine at Walmart.

Various small drills/files/phillips screwdriver. I also recommend **head mounted magnifiers** to make the soldering easier to see.

Software/Drivers

AVR Pocket Programmer Windows driver (32 and 64 bit), available here:
<http://www.sparkfun.com/datasheets/Programmers/pocketprog-driver.zip>

or

Protostack/ebay programmer drivers here:

32 bit/x86:

<http://www.protostack.com/download/USBasp-driver-0.1.12.2.zip> (don't use the 64 bit driver from this file, use the signed one below instead.)

64 bit/x64:

<http://er9x.googlecode.com/files/USBasp%201.12.0.1%20podpisany%20cyfrowo.zip>

The author of the er9x firmware for the 9x has GREATLY simplified programming by writing a 9x specific program, **eePe**:

<http://eepe.googlecode.com/svn/trunk/eePeInstall.exe>

Note: If you are going to use the Sparkfun AVR Pocket Programmer on a Vista /Windows 7 64 bit (x64) system, you also need to read the Appendix!

Hardware Installation

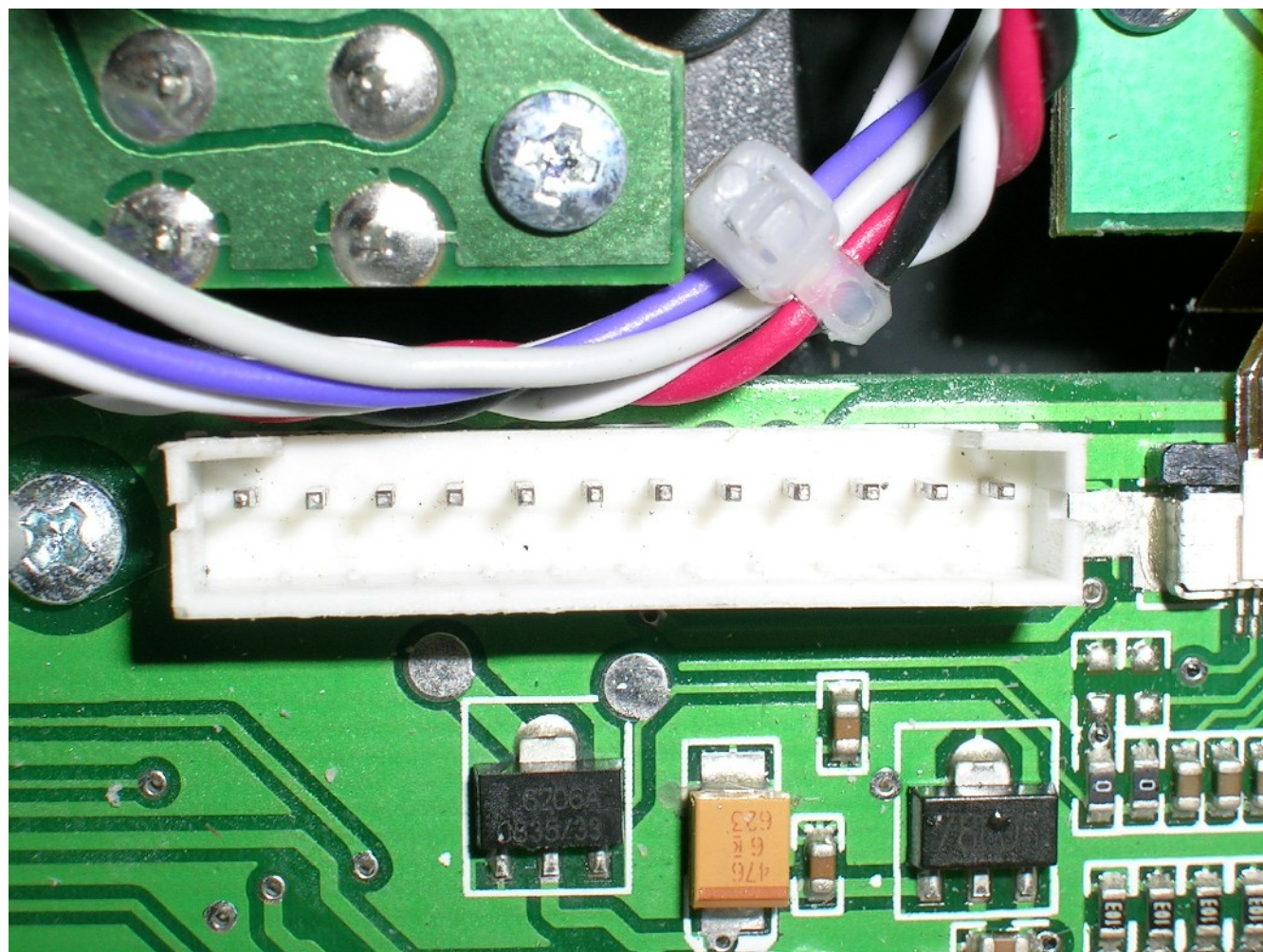
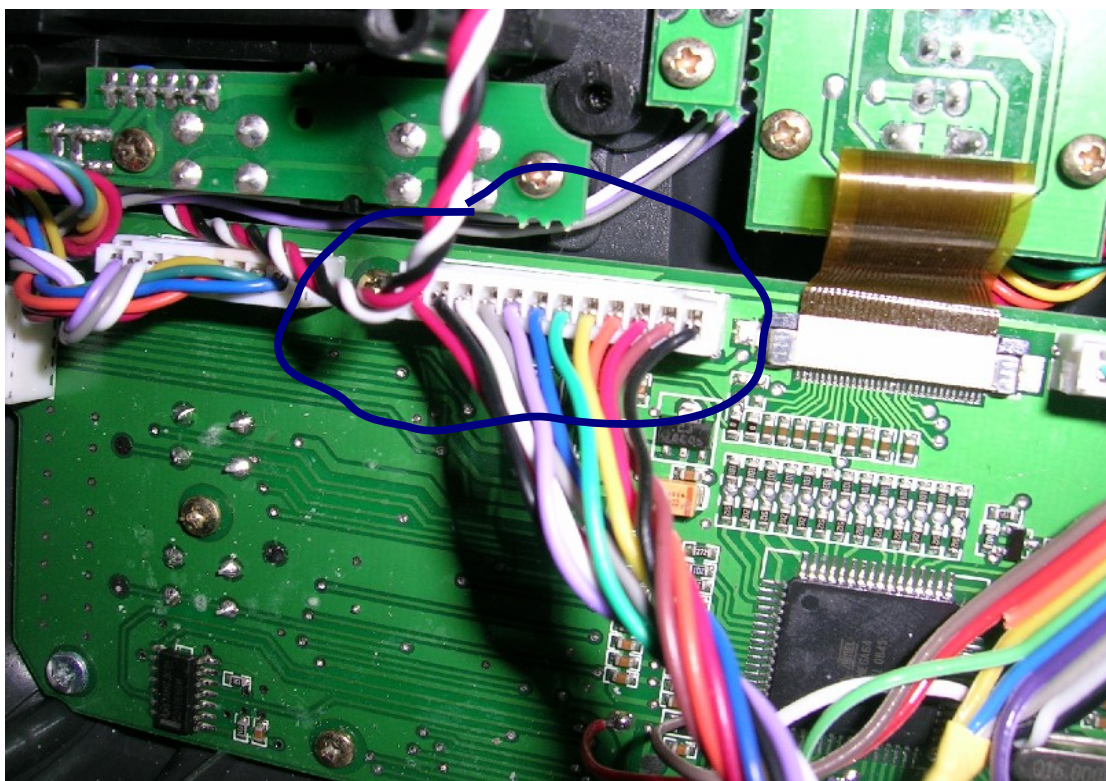
Plug in the 15-30 watt **GROUND**ED soldering iron and let it warm up for 15 minutes or so. Do NOT, under any circumstances, use any iron above 15-30 watts!! You can easily ruin the main board in your 9x if you do!! Please be SURE to use a grounded iron. You are working around semiconductors, and stray voltage or static can harm them.

Plug in the hot glue gun and let it warm up.

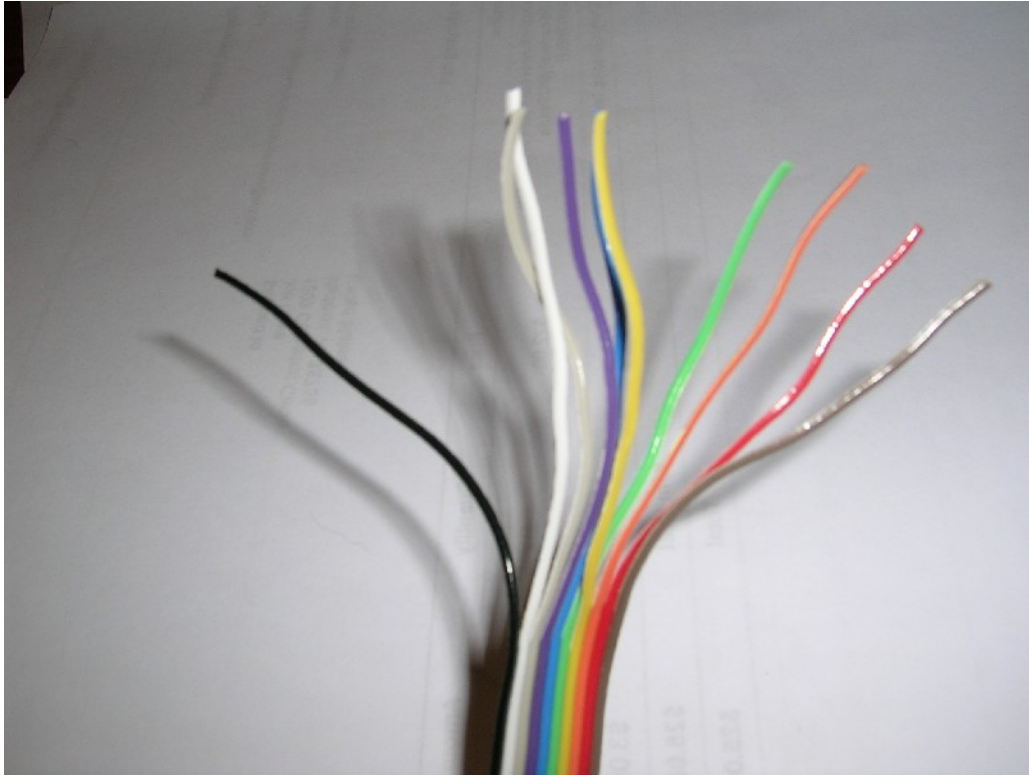
Remove the 6 phillips screws from the back of the transmitter case and retain them for reinstallation. There is no need to remove the transmitter module or battery pack first.



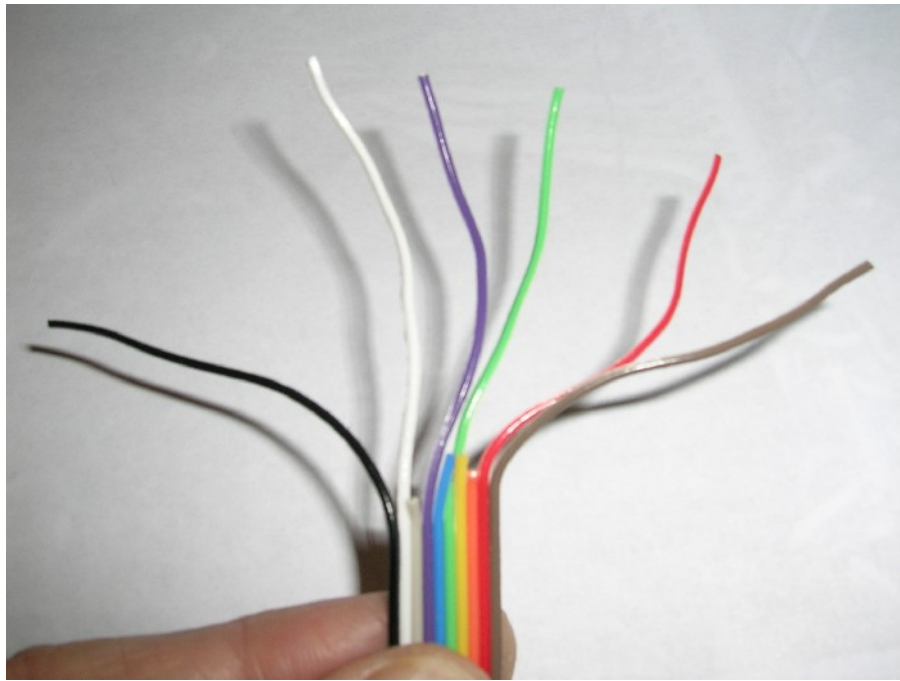
Carefully separate the case front from the back, noting the cable that connects the front and the back of the case. Carefully remove the connector on the cable from the transmitter back to the main printed circuit board (PCB) on the front half of the transmitter. Set the back of the case aside for now.



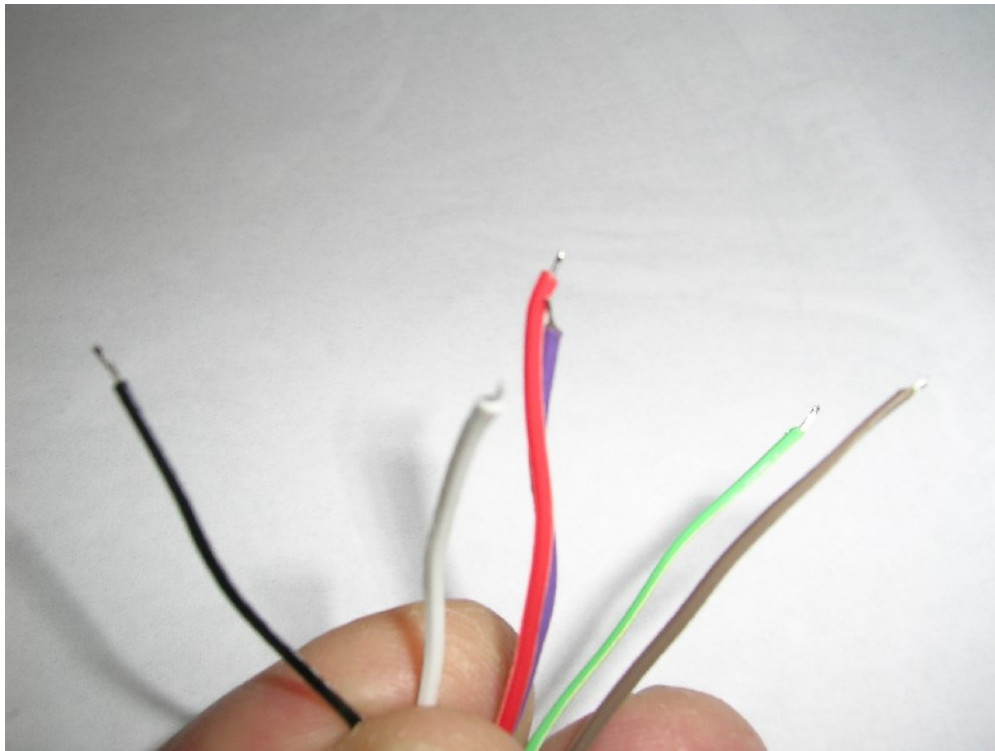
Take the color coded cable/connector combination you got from DigiKey. If you look VERY closely, you will see that there is a small arrow on one side of the connector. It SHOULD be on the end that the Brown wire is on. Cut off all but about 4 to 5 inches (10 to 12.5 cm) of cable attached to the connector. Scissors work well to cut the cable. Carefully separate the wires in the cable from each other, using your fingernails to separate them. Separate them back about 2 inches (5 cm) or so.



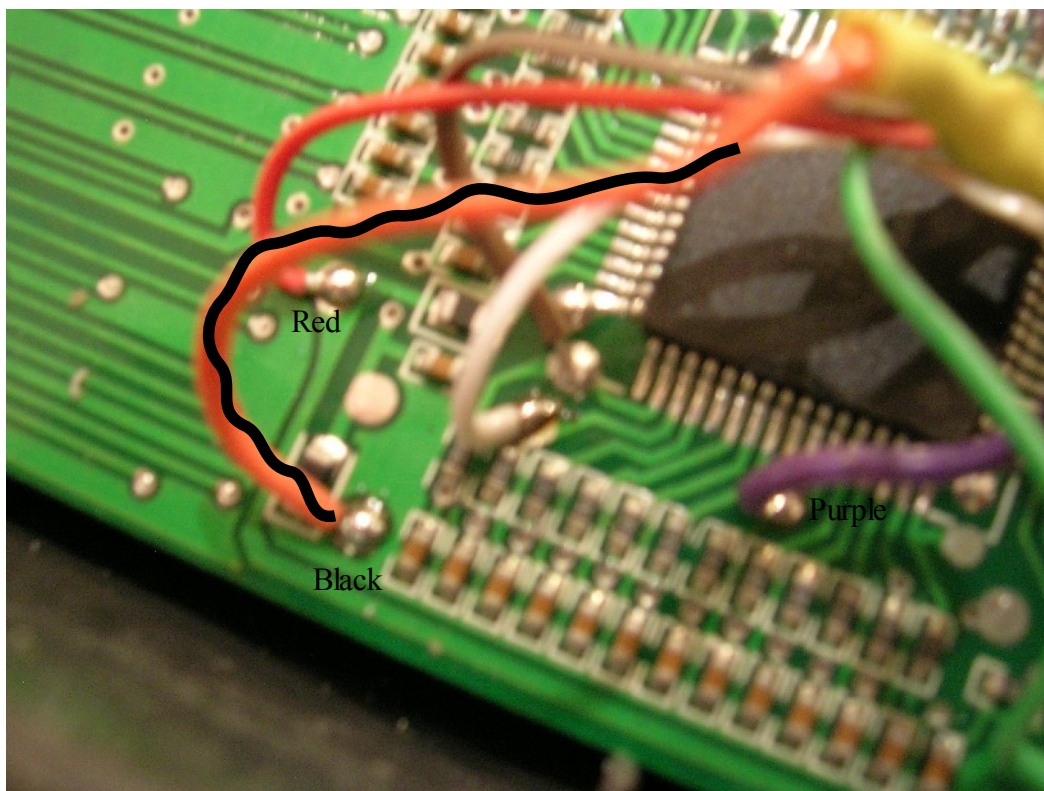
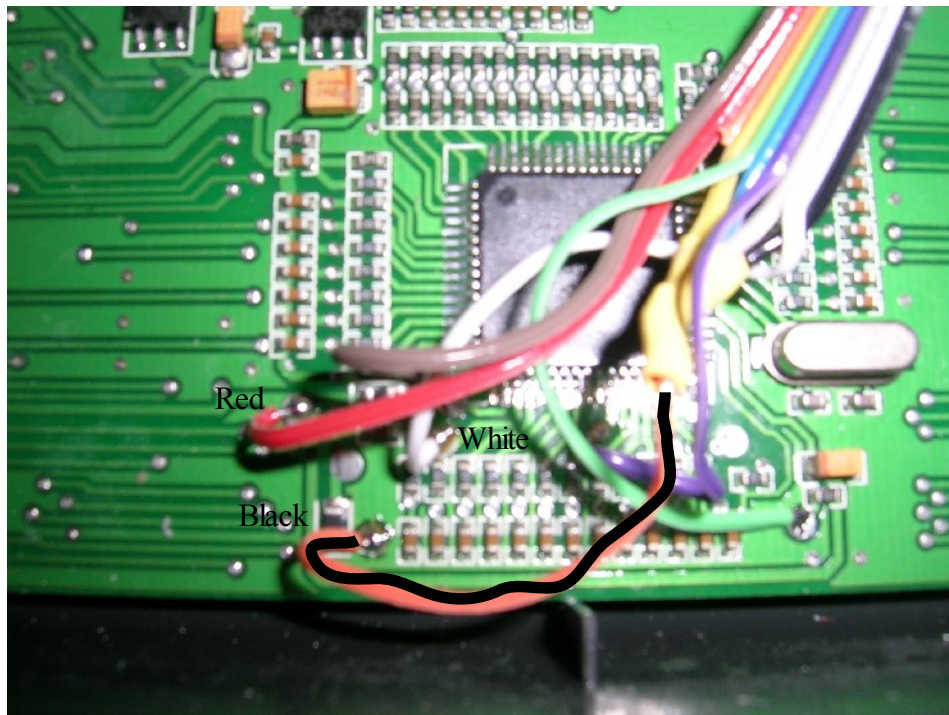
Cut the following separated wires back to the separation point. They are not required for hookup to the transmitter: Orange/Yellow/Blue/Grey. You should now have the following colors left: Brown/Red/Green/Purple/White/Black. These will be used to connect to the main PCB of the 9x.

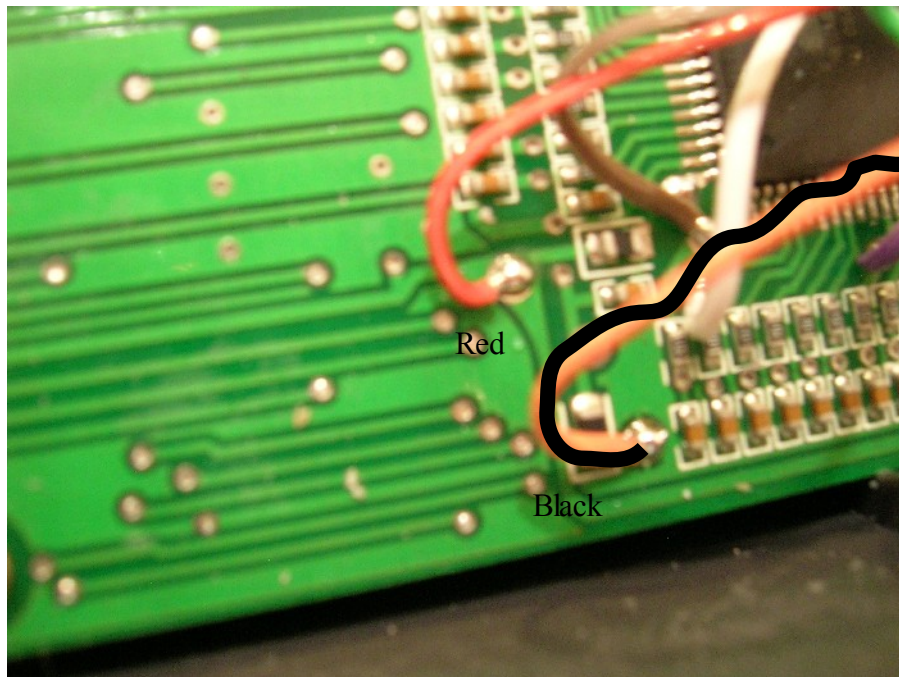


Carefully strip the insulation no more than 1/16" (1mm) from the end of each of the remaining wires, and tin the ends with solder.

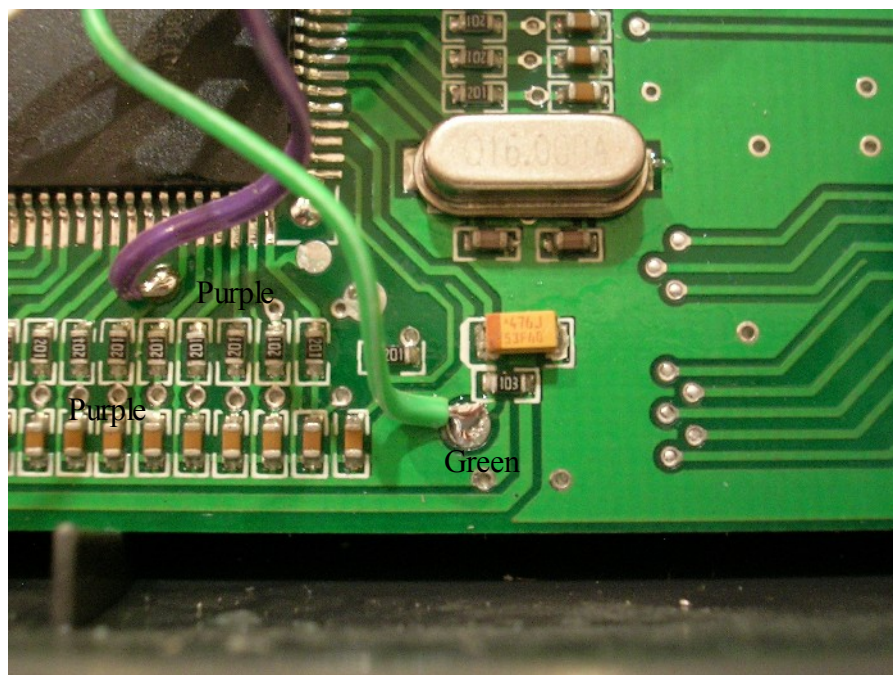


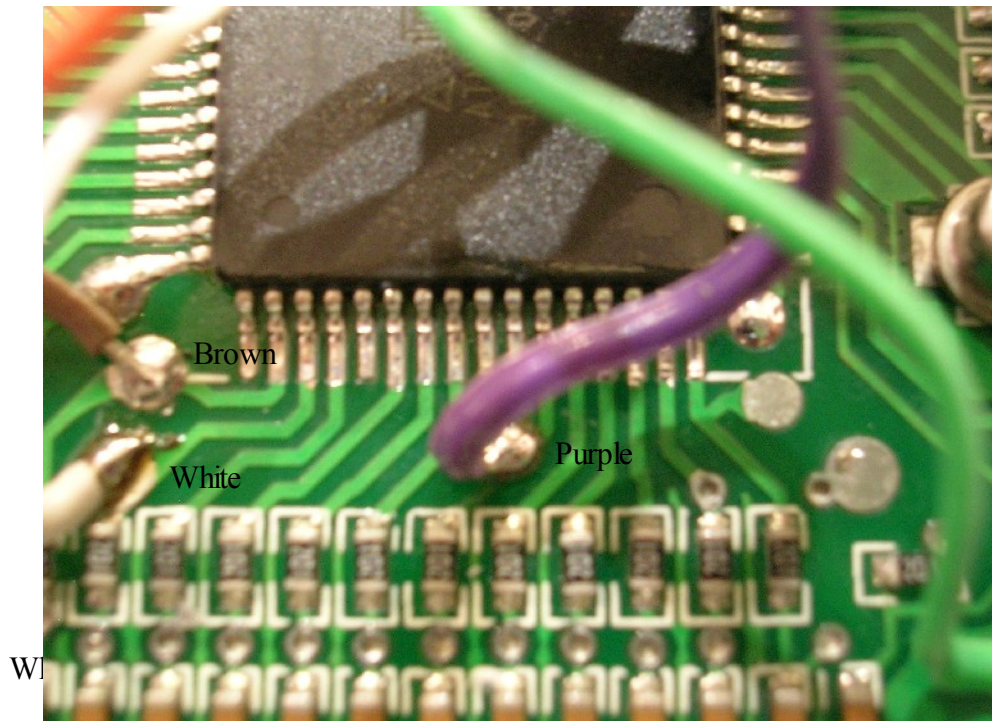
Look at the pictures below, and note where each of the colored wires will be soldered to the main PCB. Note that I colored in the black wire to show the correct color code.



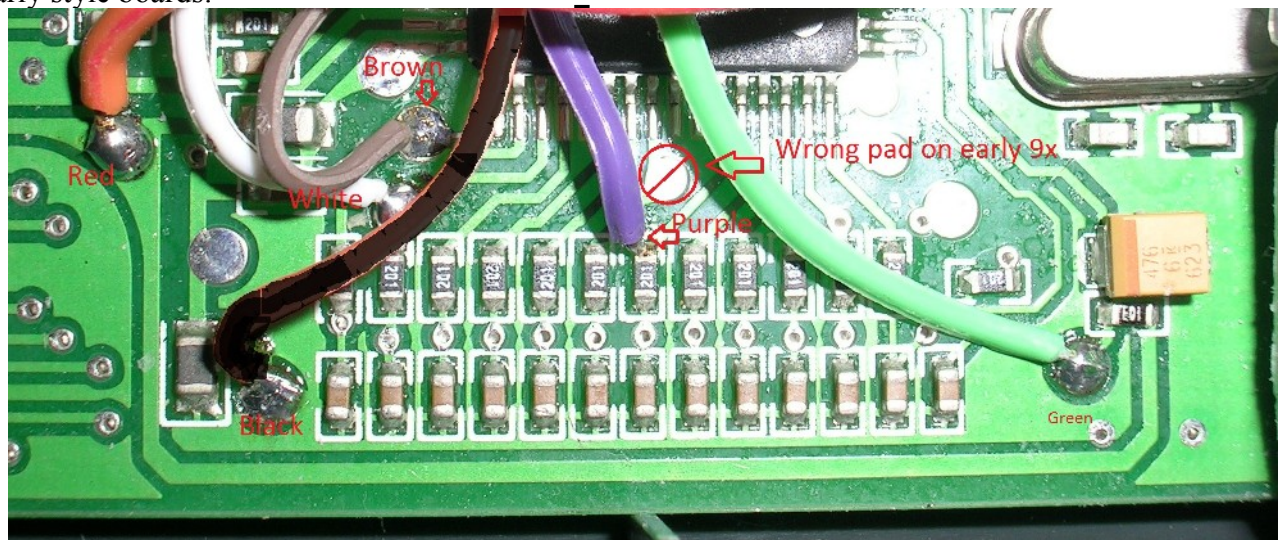


Below is where the purple (violet) wire goes on a V2 transmitter.





Note: There are two versions of the main PCB in the 9x. The later ones (aka Turnigy V2) have a pad where the Purple wire will be connected. **The vast majority of the 9x's out there are the late style.** The early ones do not have the pad; instead they have a pad on the wrong trace. If you have an early style board, you can still do the mod, you just have to be very careful with your soldering, as you will be soldering to the end of the surface mount resistor shown. The early style is shown in the picture below with the purple (violet) wire soldered to the end of the resistor. Note the red circle. On the early style, you cannot use that pad. The CORRECT connection is to the 7th resistor from the left on the early style boards:

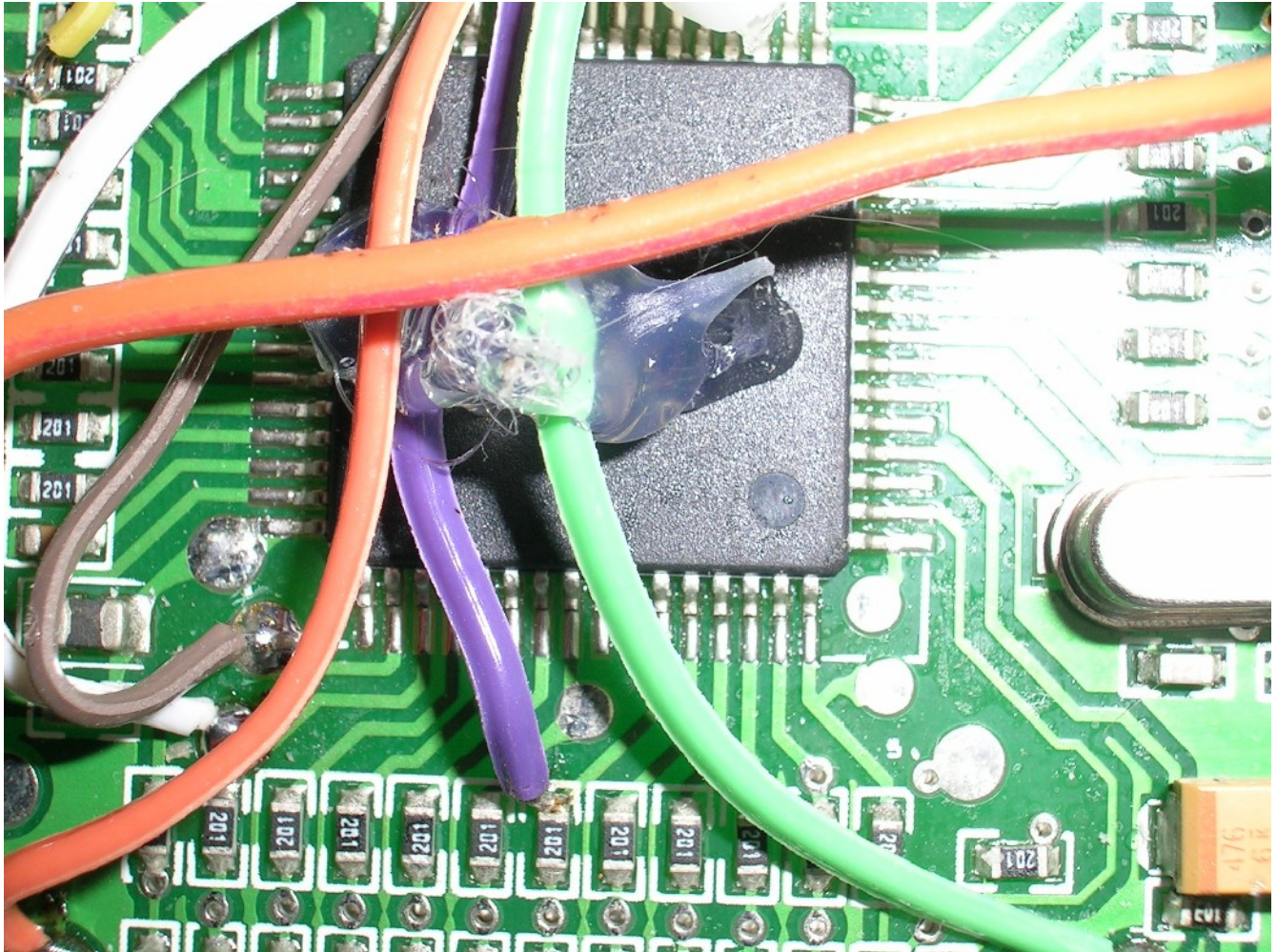


Carefully tin each of the round pads where the wires will attach to the PCB. Just a small amount of solder is necessary, but it helps to quickly solder the leads to the PCB. Make sure your soldering iron is **COMPLETELY** warmed up. Touch the iron to the pad, then add solder. Work quickly, and don't leave the iron on the pad for more than a second or so. If you have an early board, you don't need to tin the

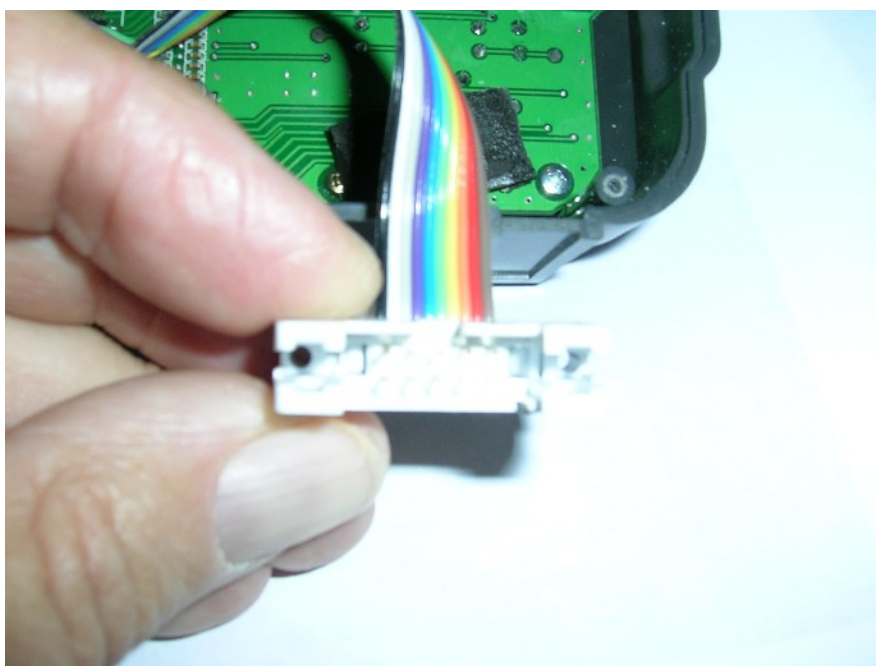
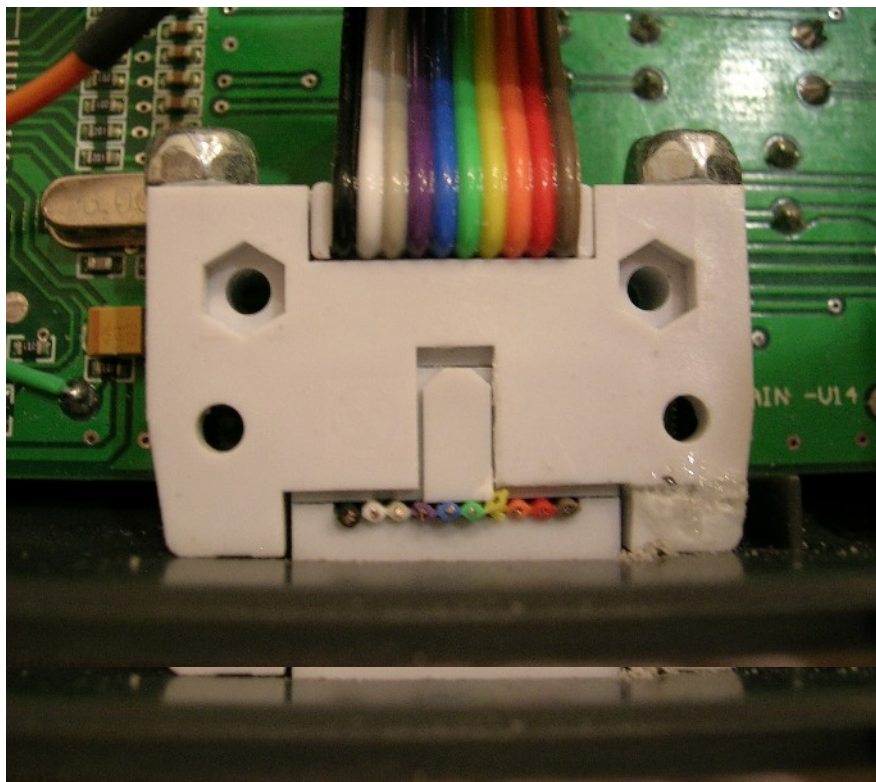
connection at the resistor. Quickly solder each of the color coded wires to the indicated spot on the PCB. If you have an early 9x, just touch the very tip of the soldering iron to the end of the resistor and the Purple colored wire simultaneously, and remove the iron quickly.

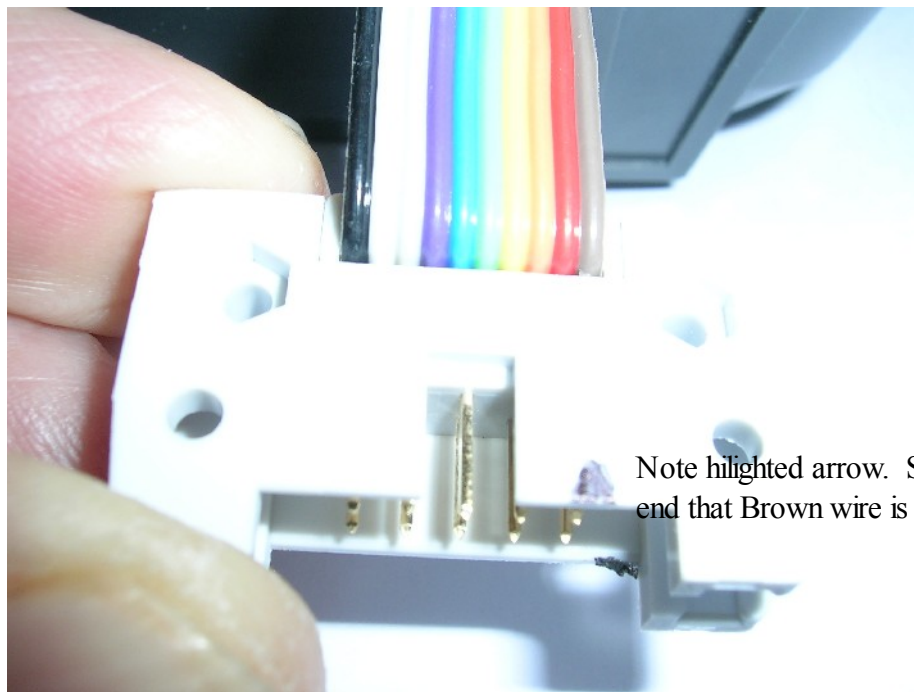
Double check that you have soldered each wire to the correct location on the PCB. There are extra round pads that we don't use, so it is easy to make a mistake.

Gather the individual wires together in a bundle, and hot glue them to the top of the main processor. This acts as a strain relief on the solder joints. I didn't do a very good job on this one, but you get the idea. **This is an early style board!**



Note the position of the connector in the transmitter case on the pictures below.





Note hilighted arrow. Should be on the end that Brown wire is on.

Measure the size of the opening of the 10 pin cable connector from the programmer and add about 1/16" (1mm) to the dimensions all around. Drill and file a hole in the bottom of the case to allow the cable from the programmer to pass thru the hole. Note that the side of the connector on the cable from the programmer that the cable comes out of will require extra clearance. Drill two holes to accommodate two long 2-56 screws, and mount the cable connector to the bottom of the transmitter case with the 2-56 bolts and nuts. Ensure that the cable from the programmer can still be inserted. The photo below looks much worse than it does in real life; just love digital closeups!





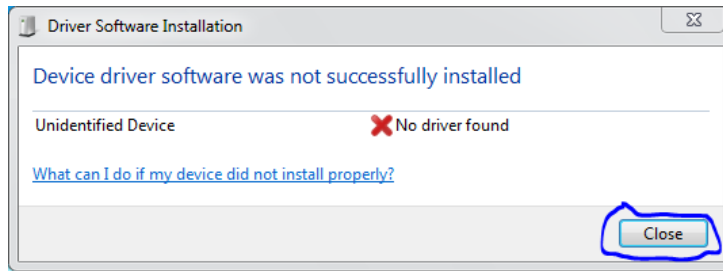
You can unplug the soldering iron and hot glue gun as you are done with them.

Take the back of the transmitter case and plug the cable connector from it back into the main PCB. Fit the two halves of the case together, and while holding them, turn on the transmitter and make sure everything works normally. If they don't, recheck your solder joints for shorts and bridges.

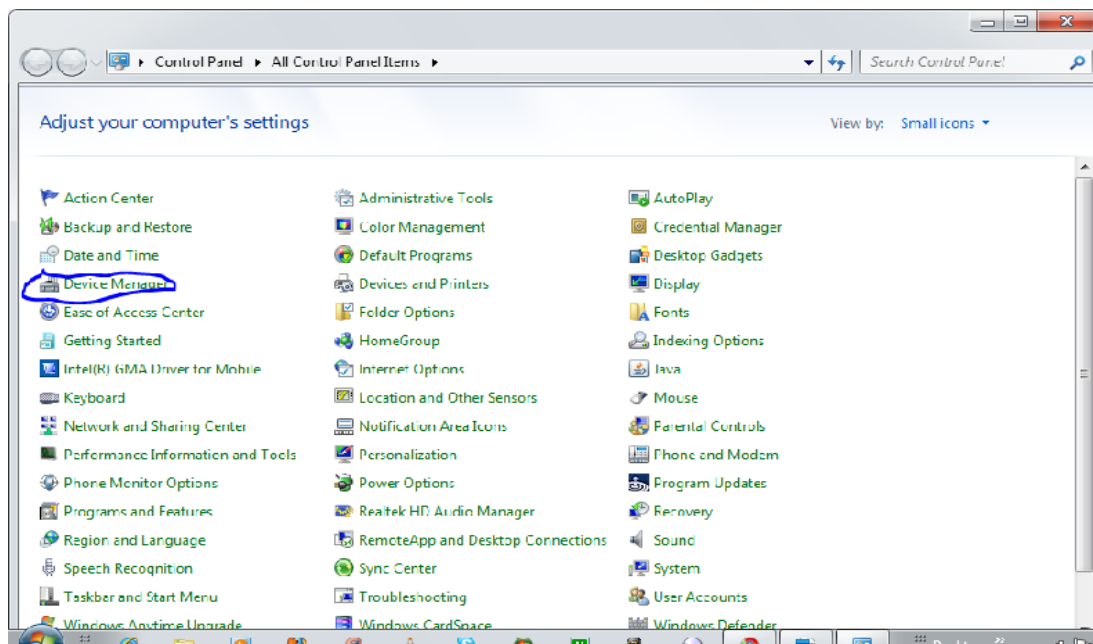
Software Installation

The drivers for the AVR Pocket Programmer and Protostack/ebay programmers do not have installers, so they will need to be installed manually. Unzip the driver package, making sure everything in it is in a single folder. Installation varies a bit for the various flavors of Windows. Some of the alternate programmers might have installation programs. Follow their instructions instead.

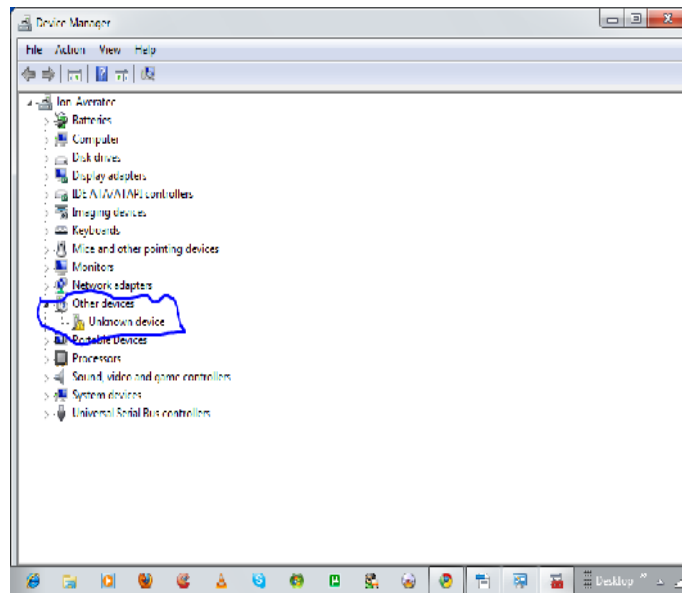
On Windows 7 and Vista, plug in the the programmer with the USB cable to the computer. Windows should say that it is installing the driver firmware. After it churns for awhile, it should come up and say that the driver was not successfully installed.



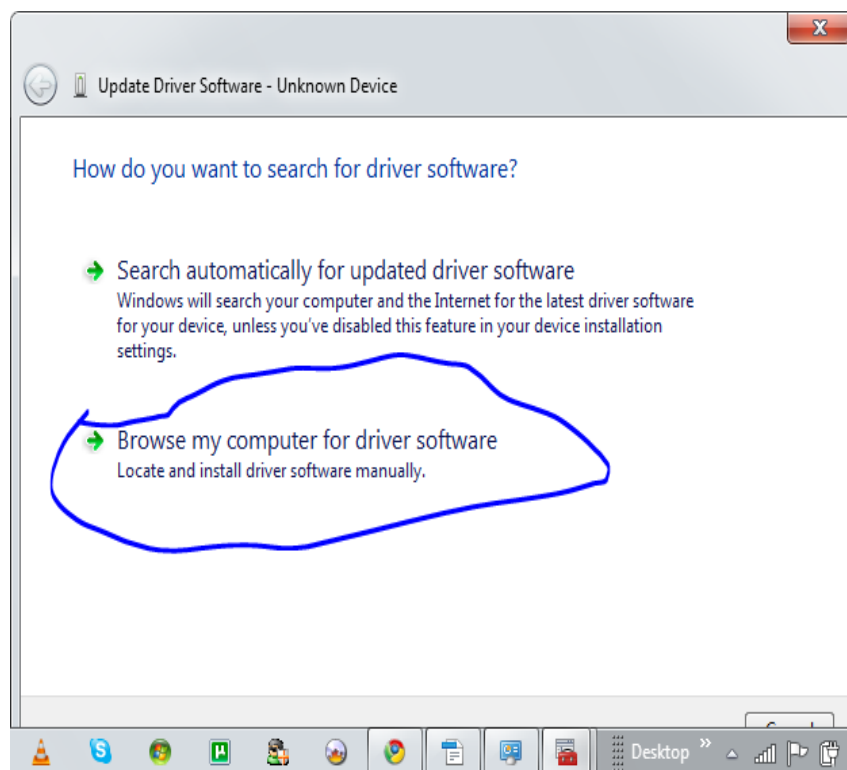
At this point, go to Start/Control Panel, and open up “Device Manager”.



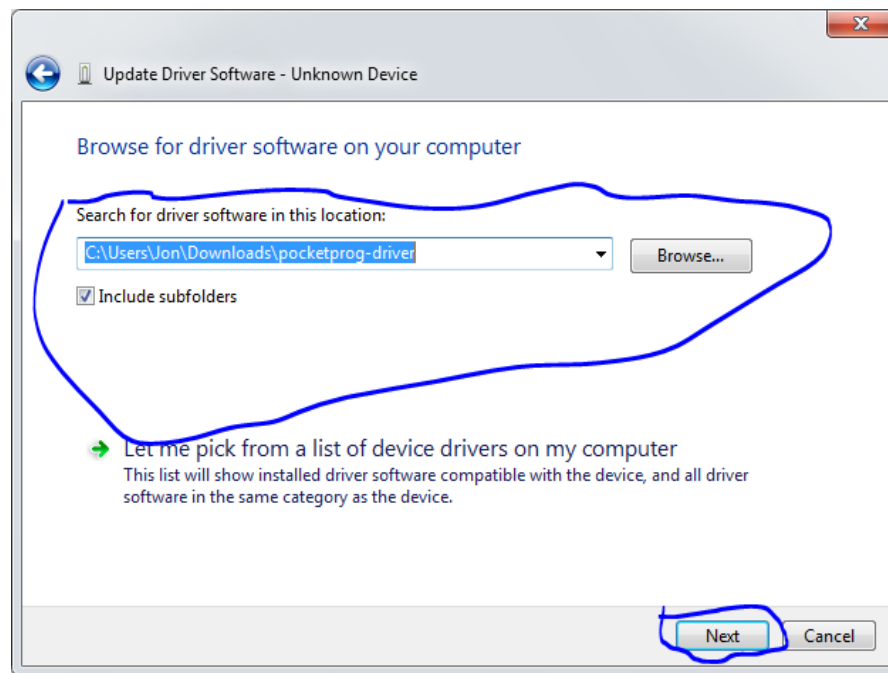
About halfway down the list you should see a “?” and Unknown USB device.



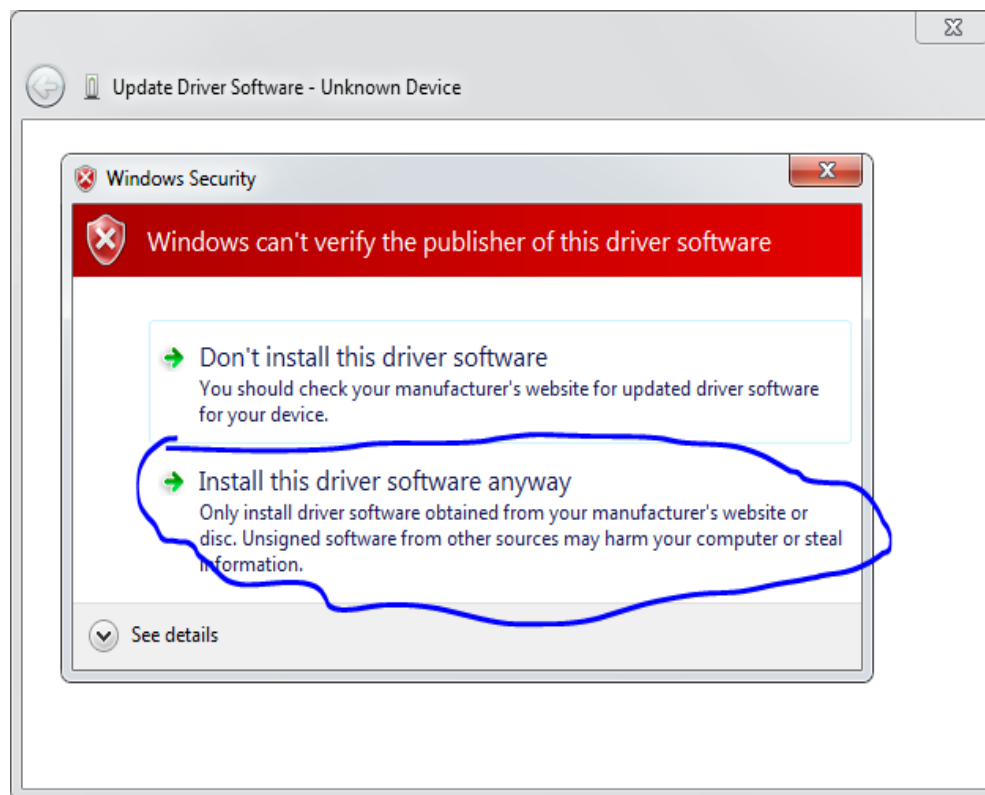
Right click on it, and select “Update Driver Software..”. You will get the Window below:



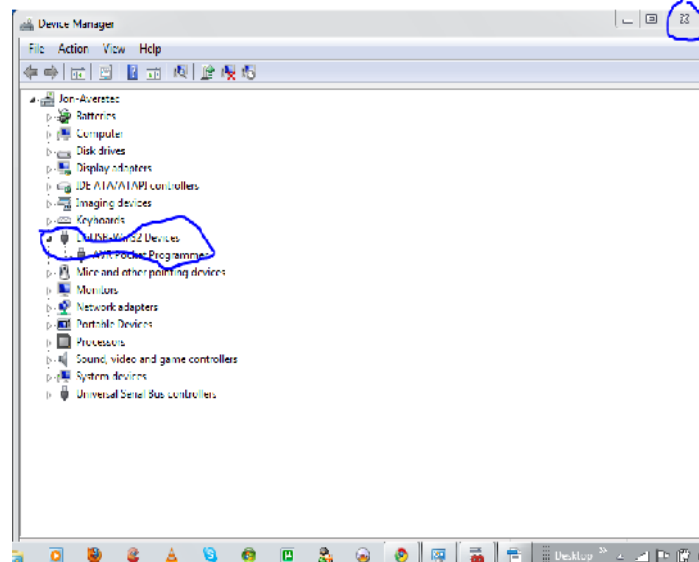
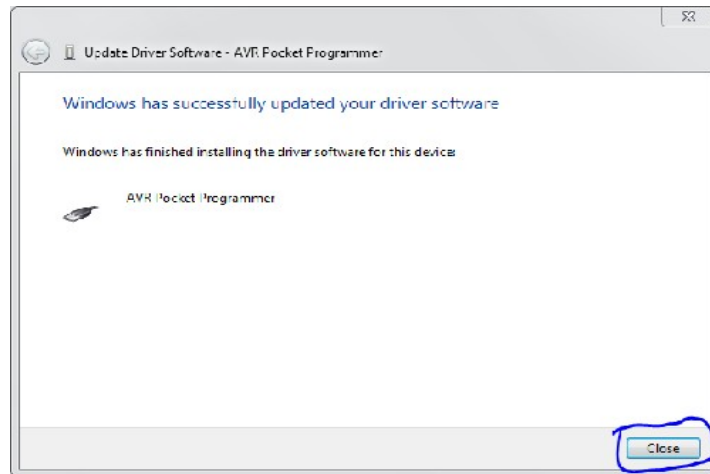
At the resulting window, browse for the folder that you unzipped, and select ok. This example shows the Pocket Programmer, but the install steps are the same for the Protostack/ebay programmer.



Click Next, and Windows should start installing the driver software. You may get the following warning:



Select as shown above. When it is done, close the resulting window, and the Device Manager window. This concludes driver installation.

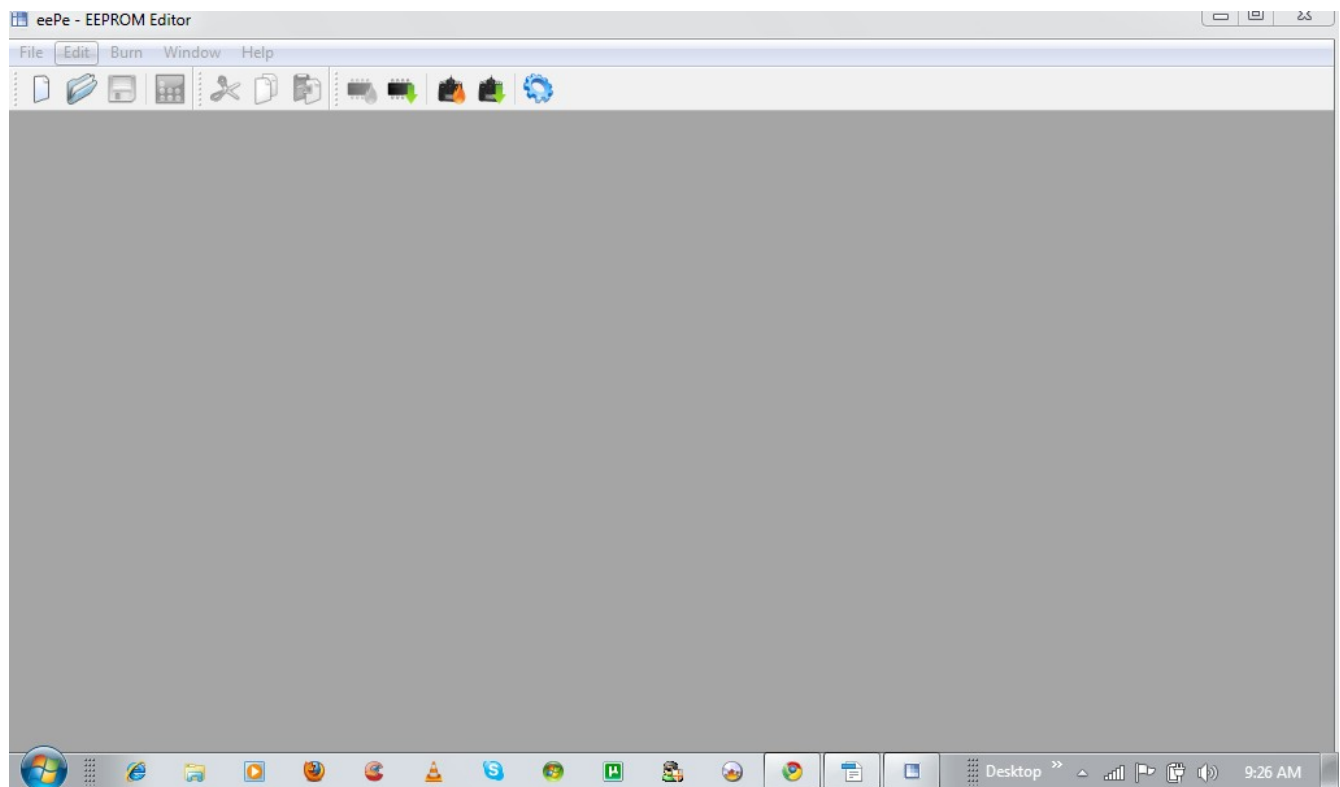


On XP or Windows 2000, when you plug in the programmer, it will give you the option of installing the drivers. Browse to the unzipped folder, and allow it to install.

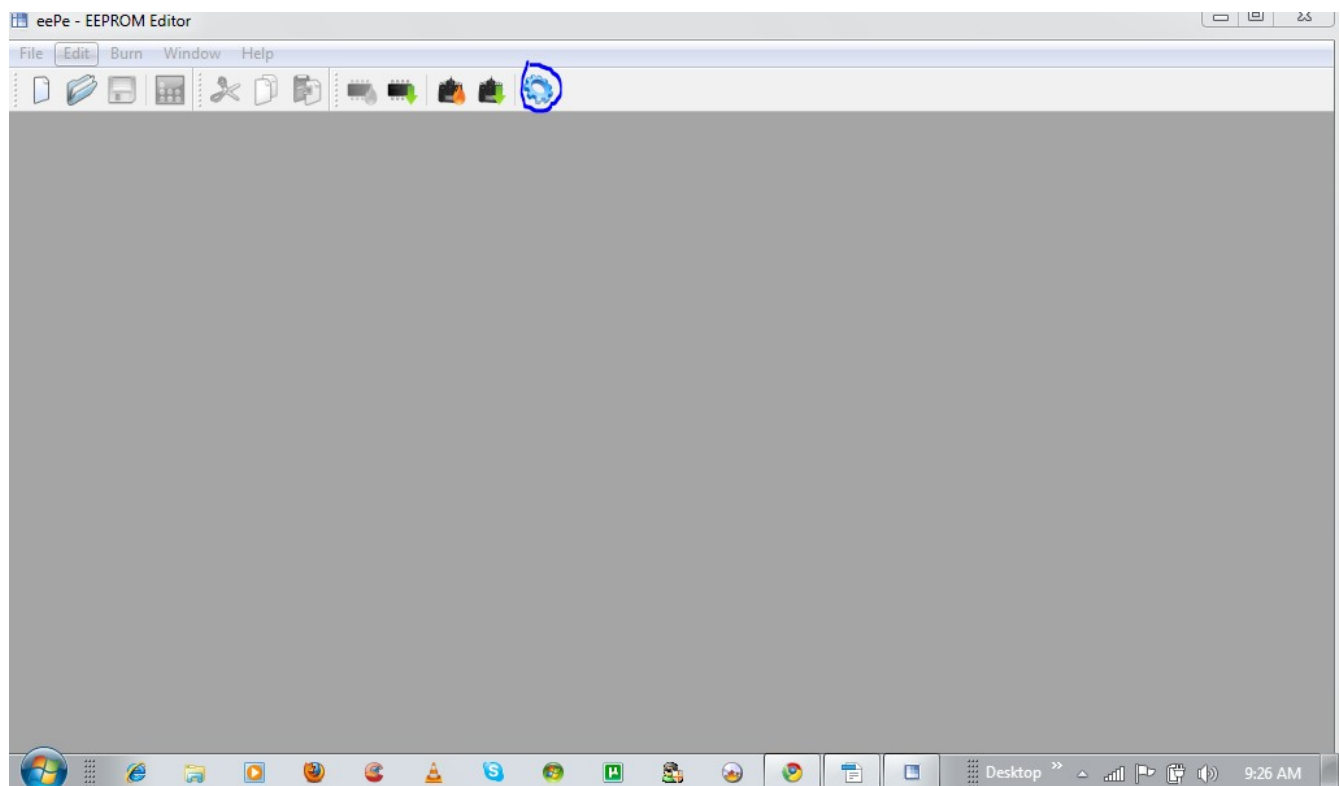
Install eePe

Installing eePe is a breeze, and SO much easier than what we had to do before Erez wrote it. In fact, it is so simple, I'm not going to show a lot of installation window shots. Just browse to where you downloaded it, and double click on the file. Use the default setup options and locations, and let it install. It is that easy!

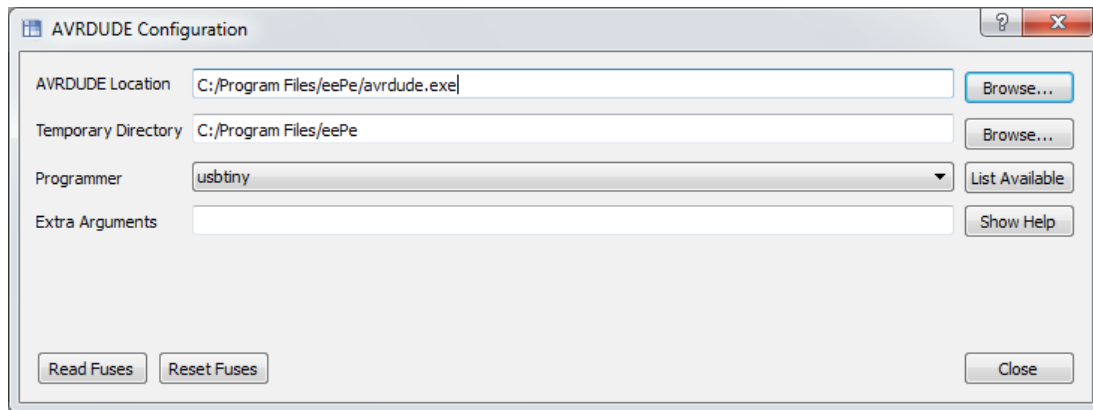
After it is finished installing, go to Start>All Programs>eePe>eePe, and start the program. When it finishes opening, you should see this:



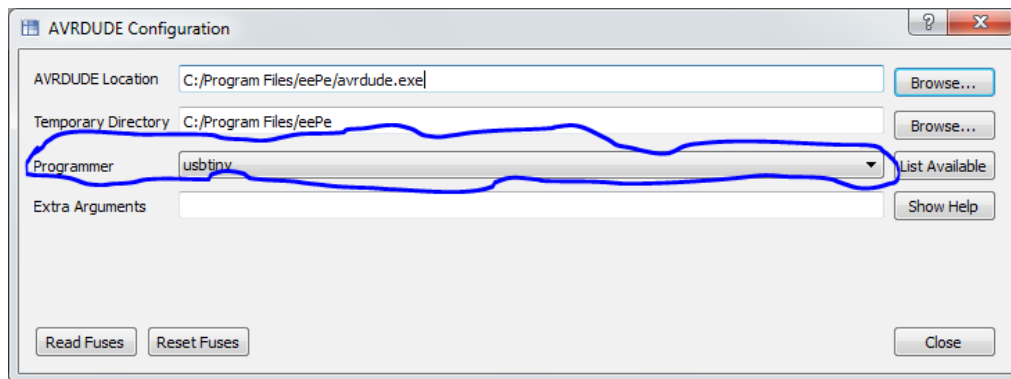
Click on the Configure Icon:



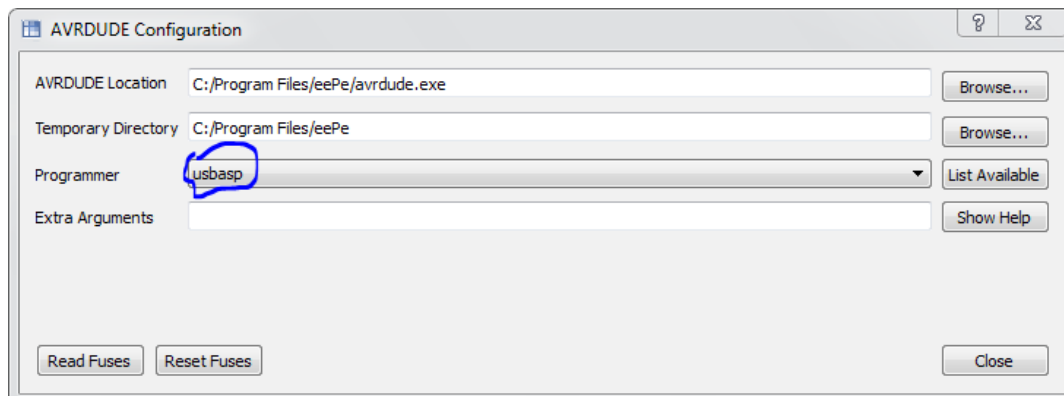
You will get a window that looks like this:



Click on the Programmer drop down box. NOTE, this is for the AVR Pocket Programmer. For the Protostack/ebay programmer, see the screenshot after this one:



And select your programmer. For the AVR Pocket Programmer, select "usbtiny", as above. For the Protostack/ebay programmer, choose "usbasp" as below:

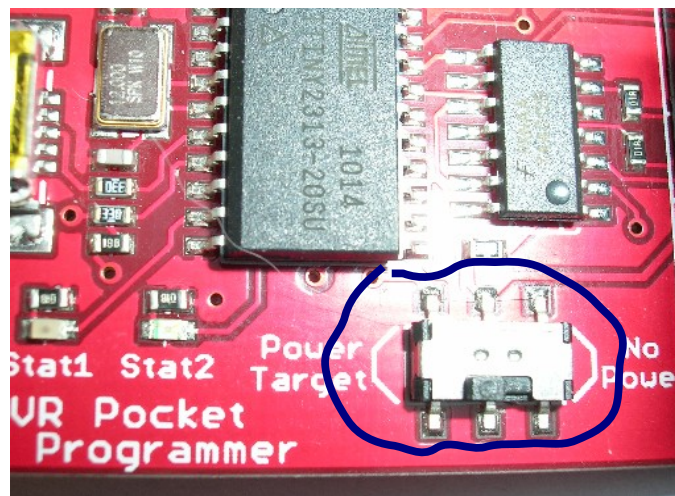


Don't touch any other buttons, except Close. If you installed eePe using the defaults, the location of AVRdude is correct. Close the window.

Backing up Flash and EEPROM

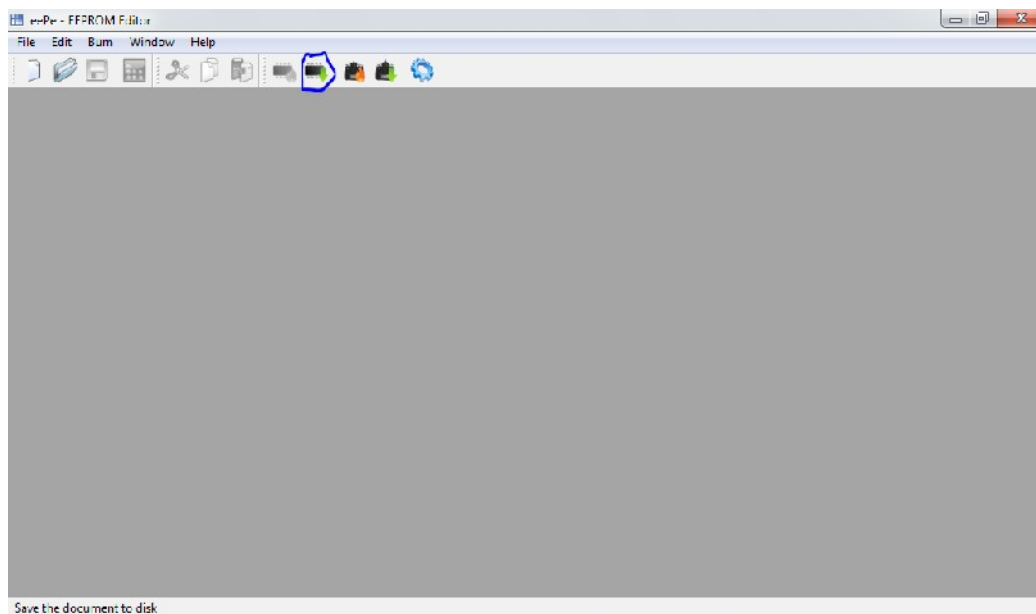
The first thing we are going to do is backup the FLASH and EEPROM in your 9x.

Plug the programmer cable into the transmitter noting the key of the connector. Don't force it in the wrong way. For the Pocket Programmer, on the end next to the cable connection to the 9x, there is a very small switch which allows the computer to power the 9x thru the programmer. Make sure the switch is set to the "Power Target" side. The Protostack/ebay programmer comes preconfigured to supply power to the 9x.



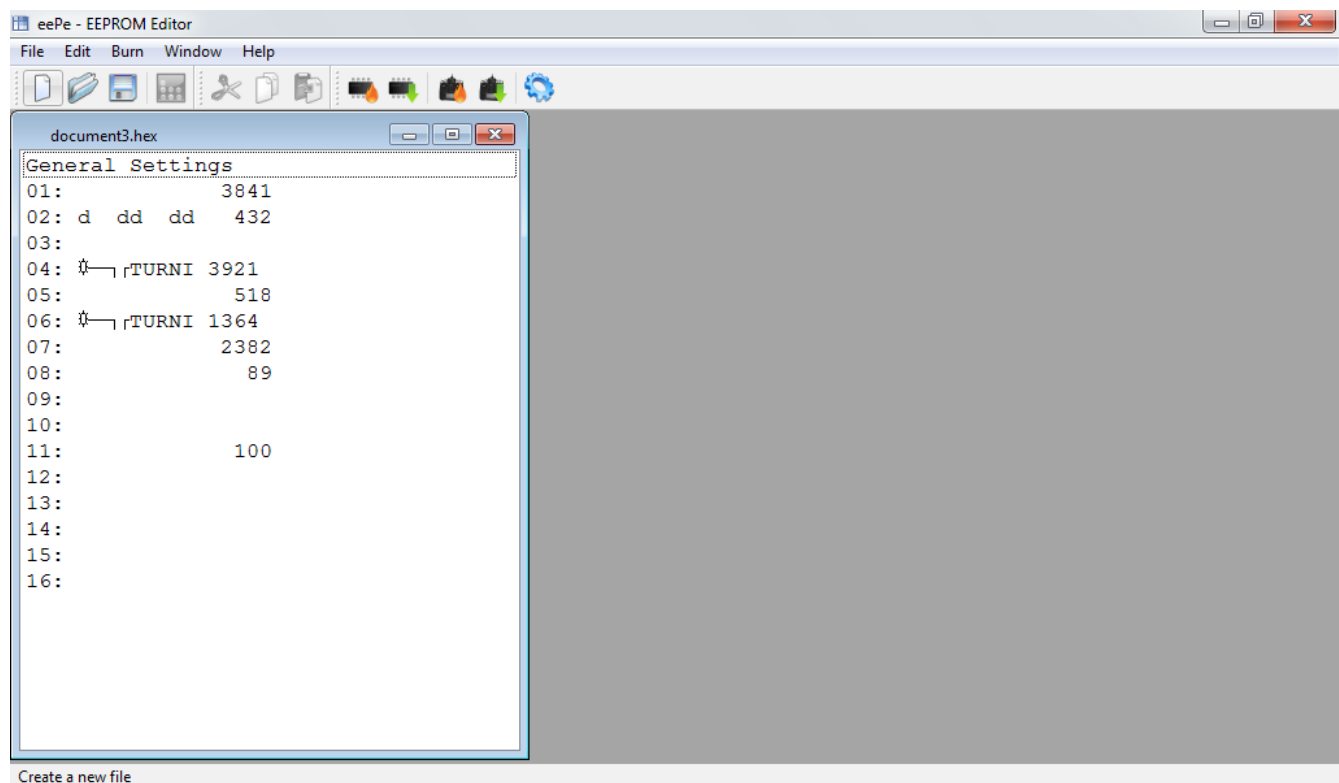
Now plug the usb cable from the programmer into the computer. If everything is connected correctly, the main screen of the transmitter should power up. Make sure to clear any Switch Error screens by flipping the switches to the up/back position. If it doesn't power up, disconnect everything, and double check the connections inside the 9x. Plug just the programmer into the computer, and make sure the leds on it light up. On the Pocket Programmer, there should be two blue leds and one red led lit. On the Protostack/ebay programmer, there should be one green led lit. If everything looks correct, plug everything up again.

Assuming you get the main screen on the 9x, hit the Read eeprom button on eePe. This will read your current eeprom/models from the 9x and save it to your computer. More leds will light up on the programmer during the process.

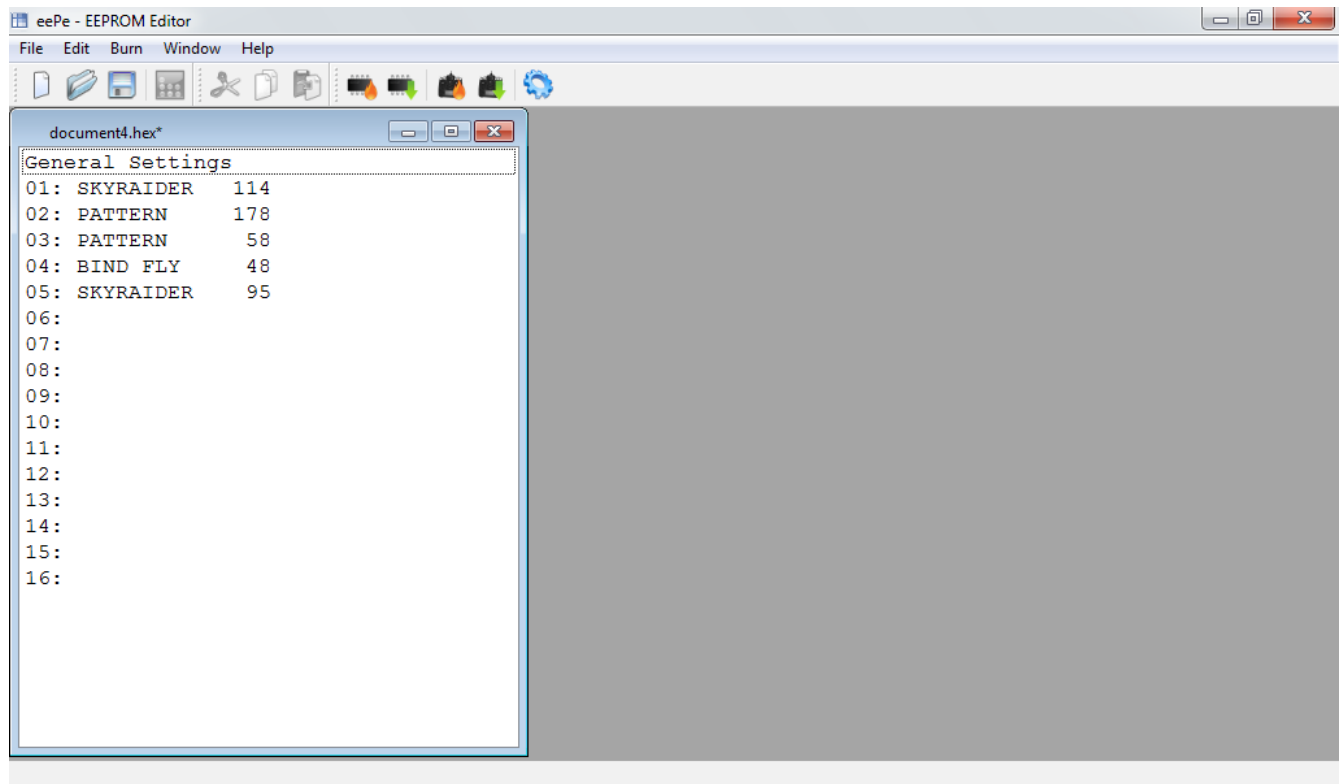


You should see the status in the eePe window as it reads the file. After a few seconds, it should complete and show the models list in the main window of eePe. (NOTE: If you are reading the eeprom from stock Turnigy/FlySky,etc firmware, the entries in the window won't make any sense. Don't worry.) Save the file by going to File>Save as..., and choose a file name and location to save it.

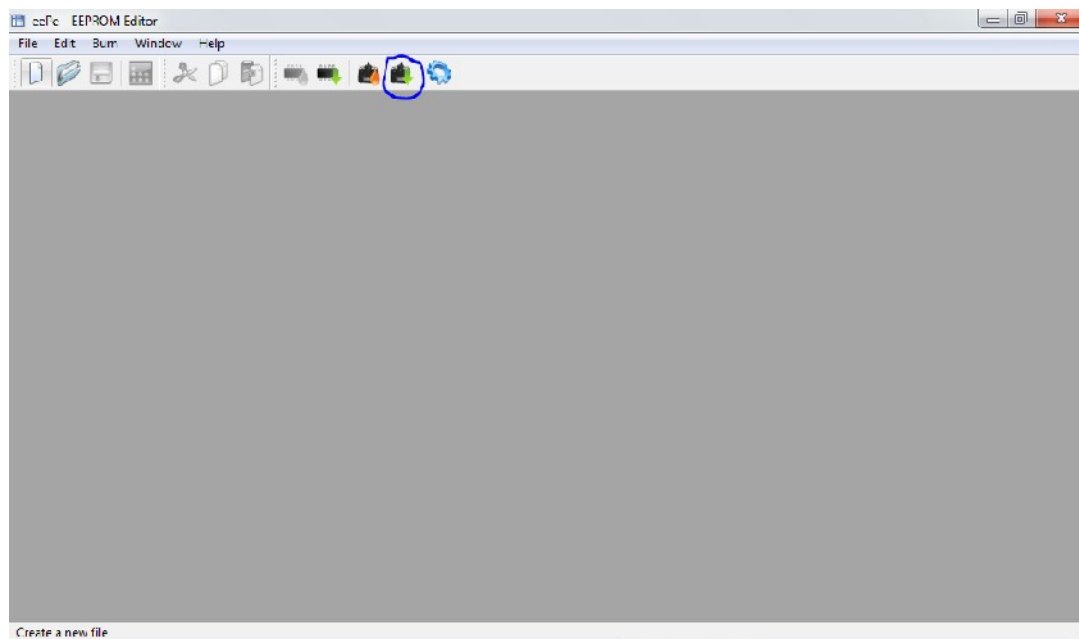
Here is how the window looks after downloading a stock firmware eeprom file:

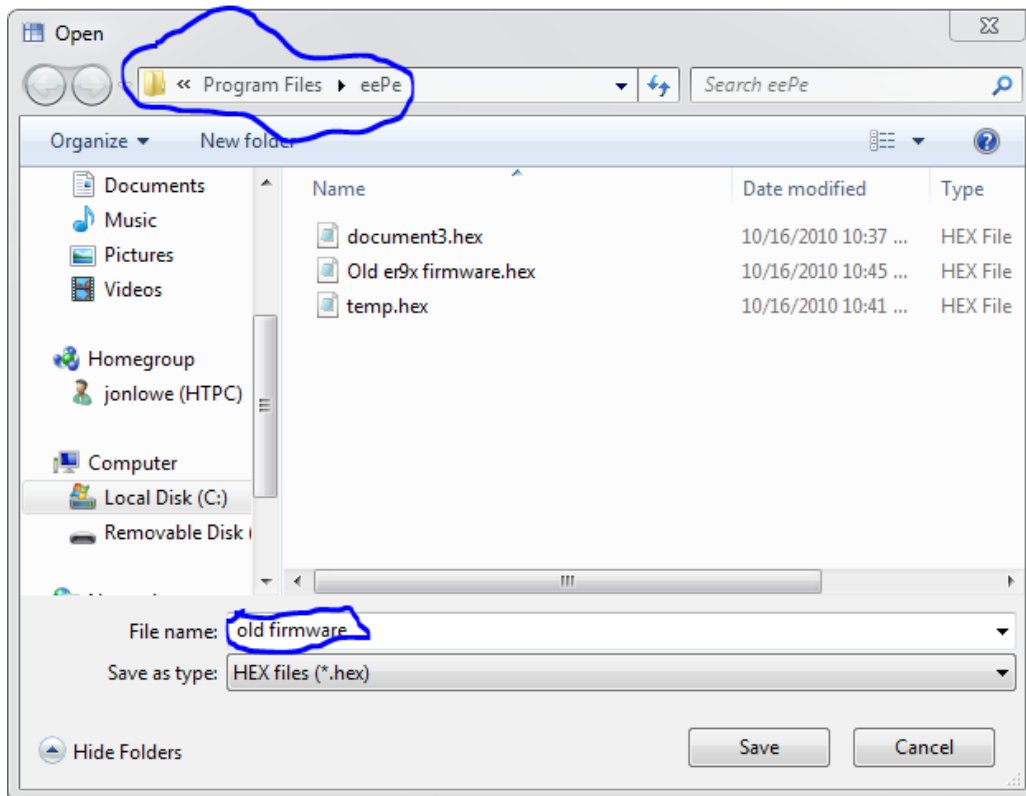


Here is how it looks for an er9x firmware:



Now do the same thing for the Flash memory (firmware). You will get a file dialog to name and save your firmware to a specific location. This may take 30 seconds or so to complete as compared to the couple of seconds for your eeprom file.





If all of that went well, you have now backed up everything on your 9x, and are ready to flash new firmware.

Flashing New Firmware

Since you are here, you must be wanting to flash different firmware to your 9x. The latest er9x firmware is here:

<http://code.google.com/p/er9x/>

The latest Thus firmware is here:

<http://code.google.com/p/th9x/>

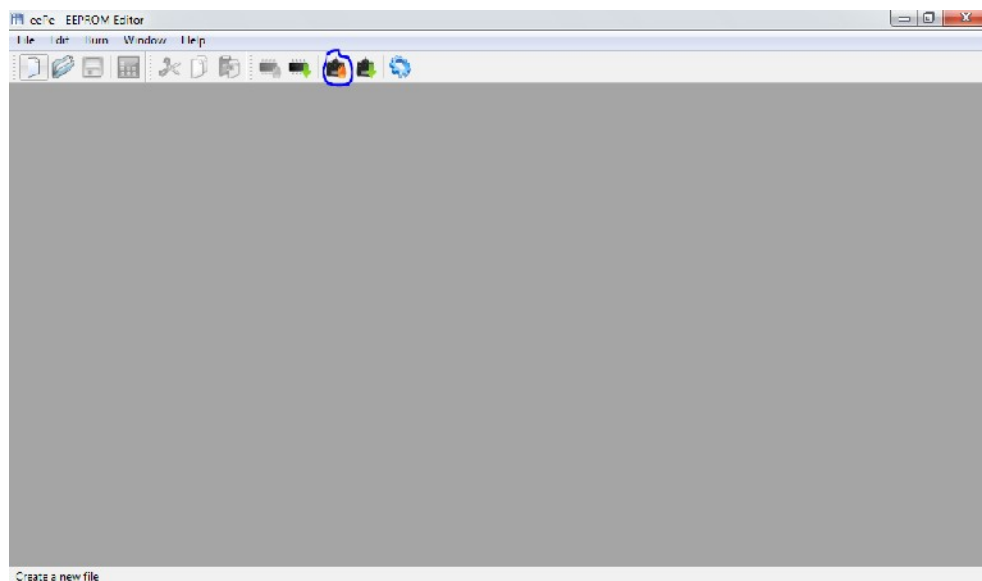
And Turnigy V2 firmware can be found in this thread:

<http://www.rcgroups.com/forums/showthread.php?t=1035575>

With the Turnigy firmware, you also need an eeprom file, and flash it too. It can usually be found with the V2 firmware. You can PROBABLY use your backed up eeprom file, but either check any saved settings carefully after flashing, or reset the entire eeprom to defaults after flashing it. You do this by pressing and holding the EXIT button on the 9x as you turn it on. It will emit a continuous beep. After 10 or 15 seconds, the beep will stop, and you will be given the default screen on your 9x.

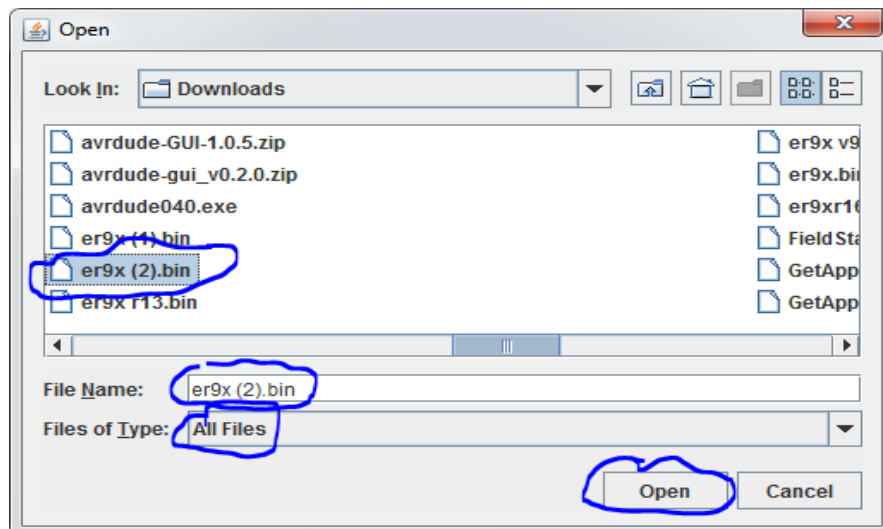
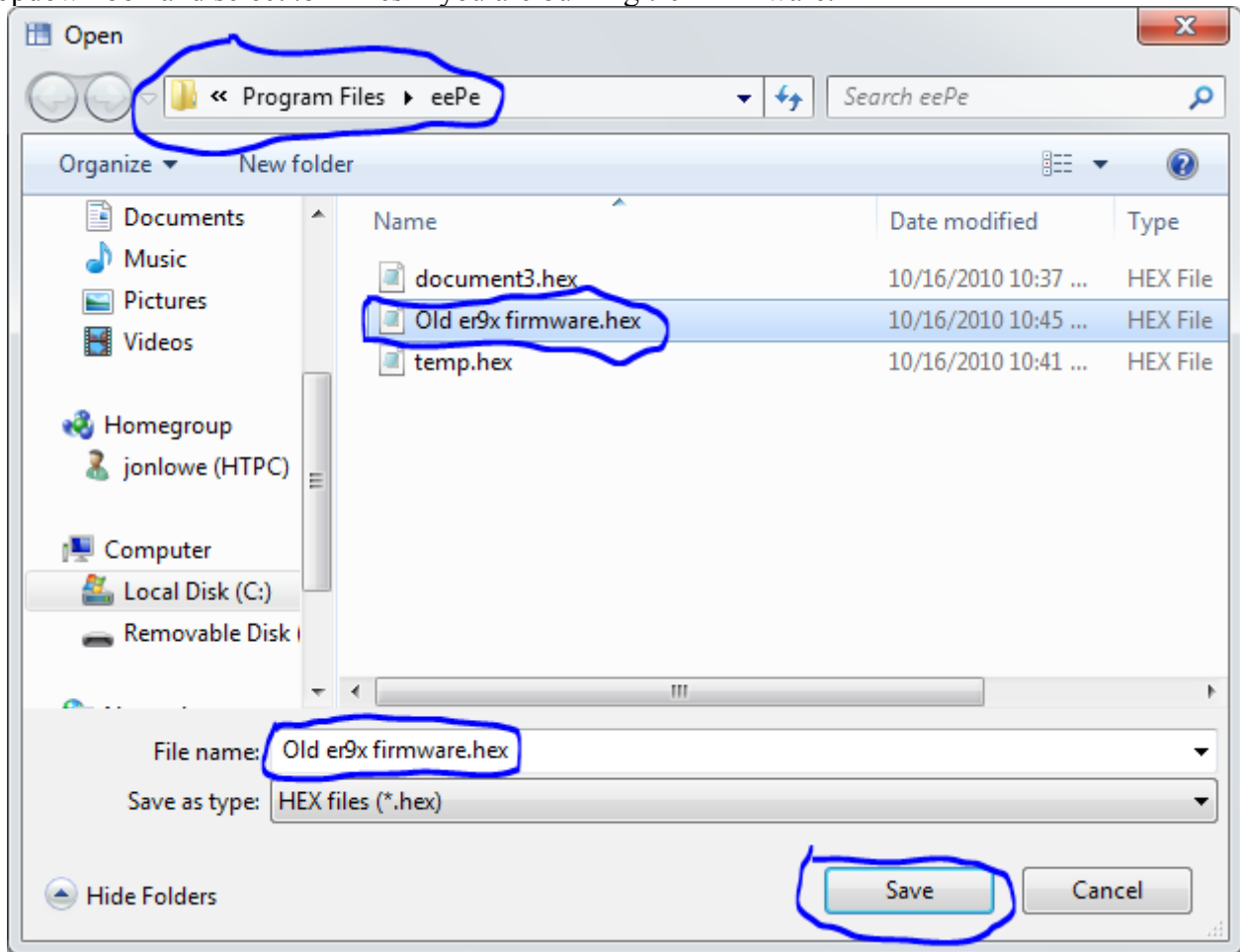
With either the er9x or the th9x firmware, you do not need to flash the eeprom if this is your first time flashing either firmware. The firmware will reset the eeprom once you have flashed the firmware.

Ok. Fire up eePe. To flash the firmware, hit the Flash firmware button on eePe:



You will get a dialog box to locate the firmware you want to flash. Find it, and click Open. Note that eePe assumes you are looking for a .hex file since that is how er9x firmware comes nowadays; hit the

dropdown box and select .bin files if you are burning th9x firmware.

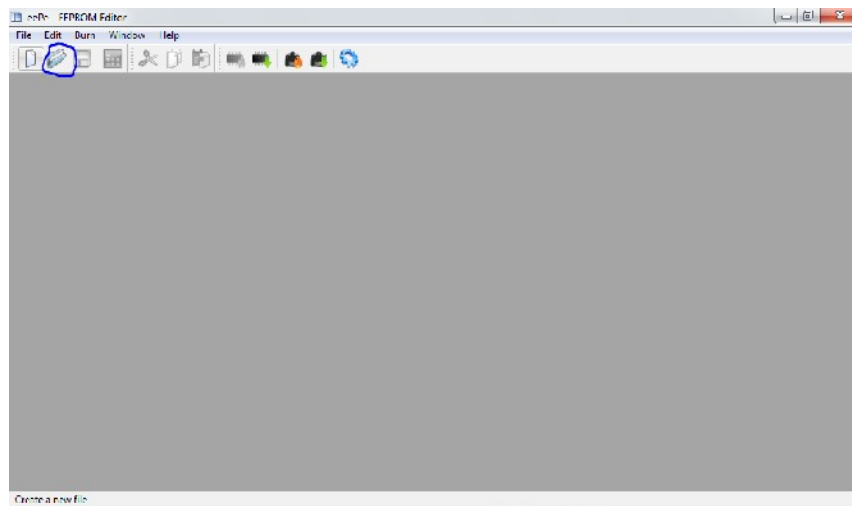


The th9x firmware is always provided in a .bin format. The er9x firmware is now in the .hex format
The Turnigy firmware can be found in either .bin or .hex format.

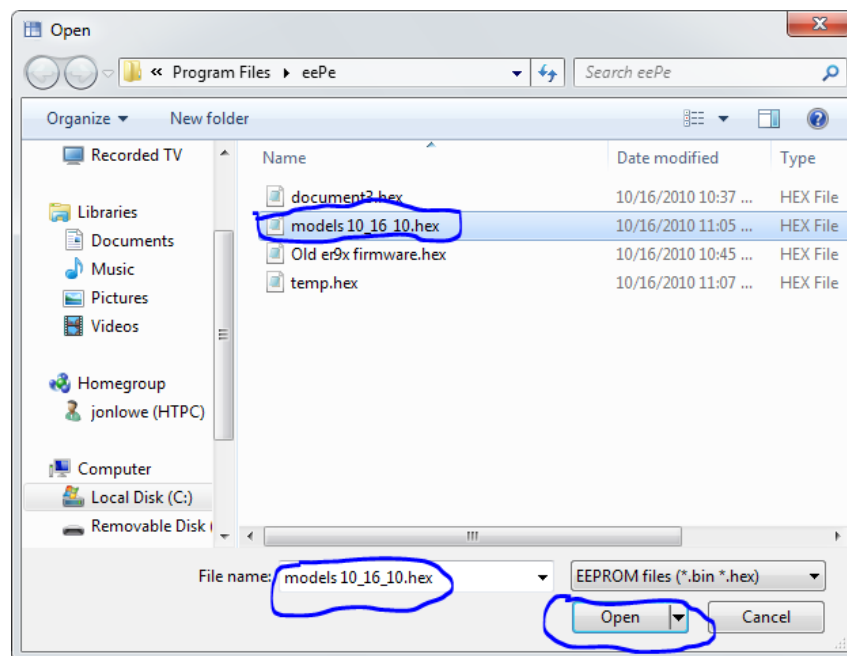
You will see a status of the flashing process. It will take 30 seconds or so, plus several more to verify the flash.

If everything went well, you will get an error screen on your 9x about the eeprom; that's a GOOD thing! For the er9x and th9x firmware, press any key, and the eeprom will be formatted. For the Turnigy V2 firmware, you will need to load an eeprom file. Just follow the instructions above for flashing the firmware, only use the EEPROM area of eePe and be sure to select the right file.

If you want to reload a previous saved eeprom file after burning a newer version of firmware, there is an added step. First select the eeprom you want to reload into your transmitter by hitting the File button on eePe.



Select the file you want to reload, in the resulting dialog box:



If all worked well, you should now have your new firmware installed, and have either a fresh eprom, or a reflashed old one installed. Disconnect everything, turn yioir 9x on, and calibrate your sticks as described in the next section. Note: if you reinstalled saved eeprom from YOUR 9x, you should not have to recalibrate your sticks. If you used someone elses eeprom, or yu have changed to a different firmware, you MUST recalibrate your sticks.`

After Flashing Your Firmware

After flashing your new firmware (and eeprom, if applicable), you will need to calibrate the sticks on your 9x for centers and endpoints. You calibrate the sticks in the er9x and th9x firmware by following the instructions in the er9x or th9x manuals, respectively, here:

<http://er9x.googlecode.com/svn/trunk/ER9x%20Users%20Guide.pdf>

<http://th9x.googlecode.com/svn/trunk/doc/th9x-en.pdf>

The instructions to calibrate the ORIGINAL (**NOT** th9x or er9x) Turnigy firmware are below. They are a direct lift from my post on RC Groups here, based on instructions on a German site:

<http://www.rcgroups.com/forums/showpost.php?p=15328341&postcount=2415>

With the 9x turned off, push the right lower trim tab to the left (aileron on a mode 2 TX), and the left side trim tab up (Throttle trim on a mode 2 tx), and while holding them, turn the Tx on. You will get a screen that shows the version number of the firmware, with four zeroes (0000) at the top of the screen. Center both sticks in their range in all axis, and press the menu button until it beeps and release. The screen number will change to 0001. Move the right stick to the upper right corner, and press menu, and you should get a beep, and the number on the screen will change to 0002. Put the right stick in the lower left corner, and press menu, and get a beep, and the number will change to 0003. Release the right stick, and put the left stick in the upper right corner, and press menu, get the beep, and the number will change to 0004. Put the left stick in the lower left corner, and press menu again, get the beep and the number will change to 0005. Center both sticks, and press menu, get the beep. You are done calibrating the sticks.

While at the same screen, press the + key, the screen will change, and you will get the position of the controls in hex code. You can see the numbers change as you move the sticks or knobs. Press - to go back to the firmware version screen. Exit the firmware screen, and go to the Display menu item. As long as you haven't turned the TX off, you will see the hex codes for stick positions for the 4 primary controls. Once you turn the tx off and back on, and go back to the Display menu item, the numbers will disappear.

That's it.

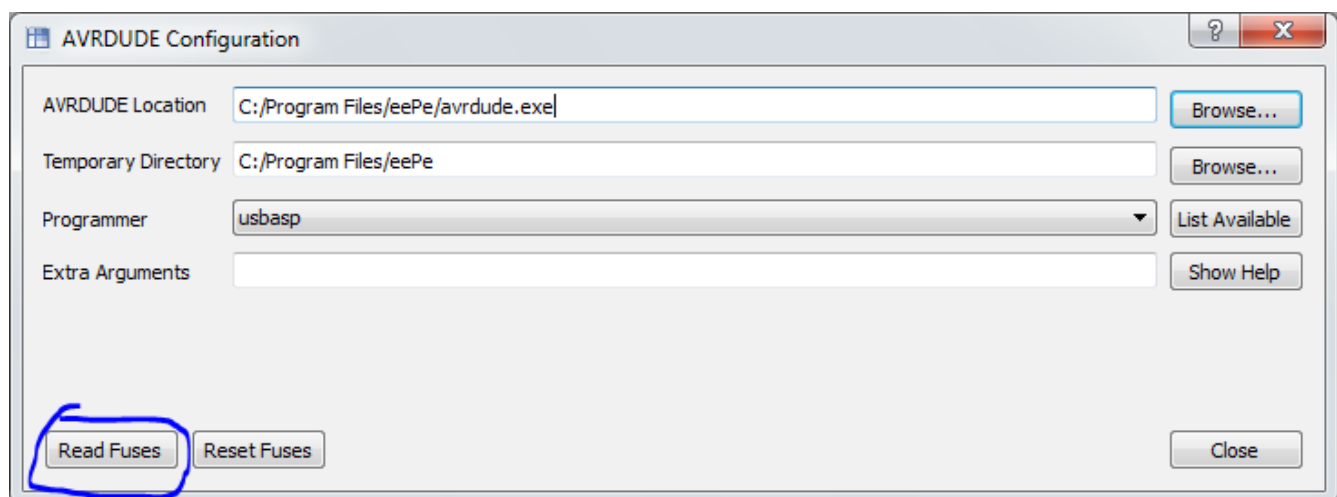
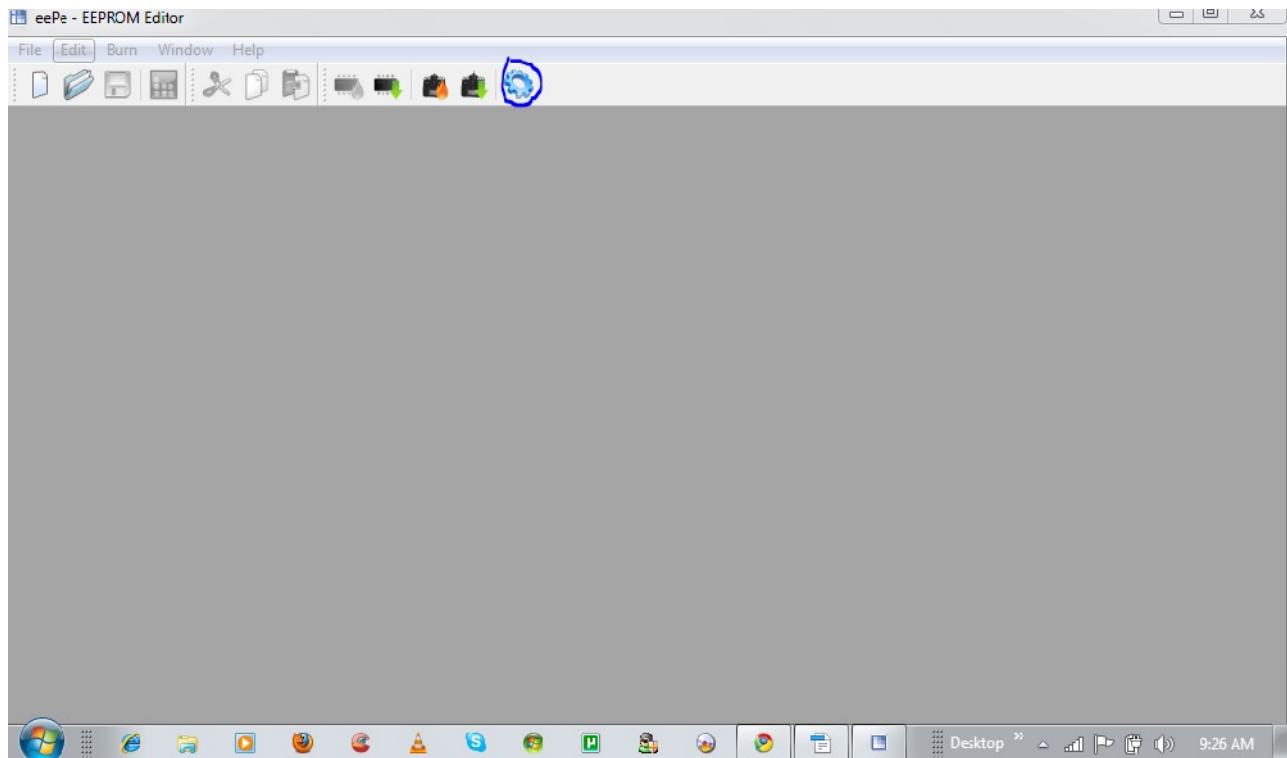
Trouble shooting

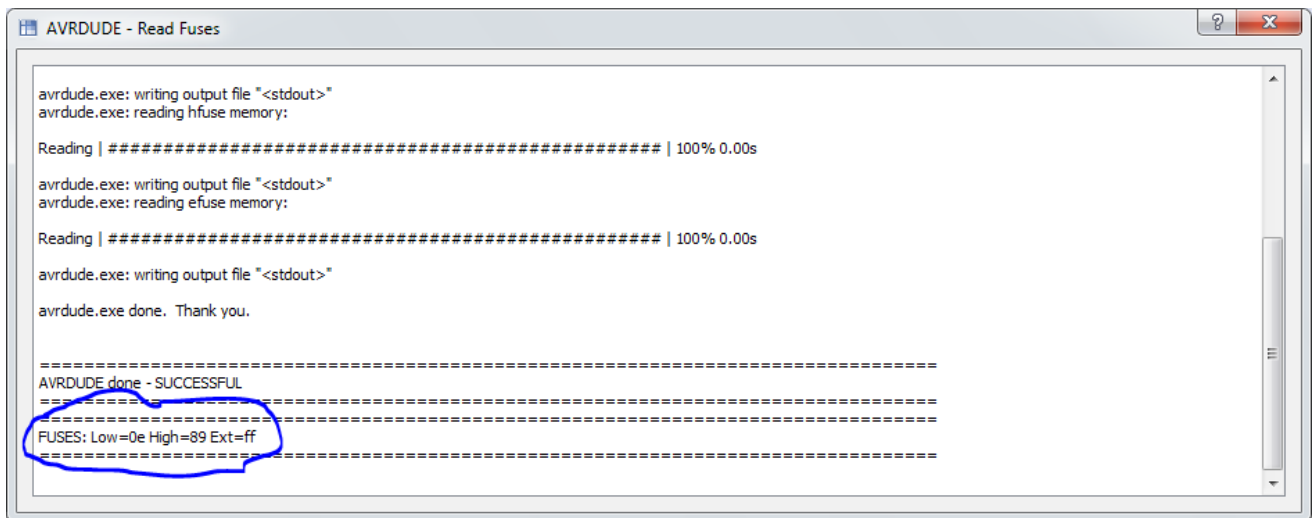
This is a tough area to write about. Problems can fall into four main areas:

1. The programmer is not recognized by eePe/AVRdude.
If this happens, check the following.
 - a. Is the driver for your programmer installed correctly? Check in the Devices section of the control panel in Vista and Win7, or System/Hardware for XP and 2000. There should not be a question mark or exclamation next to the programmer entry. If there is, reinstall the driver.
 - b. Did you select the correct type of programmer in eePe? It should be “usbtiny” for the AVR Pocket Programmer, and “usbasp” for the Protostack/ebay programmer.
 - c. Unplug and replug the programmer from the usb port. If there are only two leds lit on your AVR Pocket Programmer, you definitely have to do this. Three leds must be lit, or it won't program.
2. The transmitter does not turn on when plugged into the programmer and the programmer is plugged into the computer.
 - a. Your wiring is incorrect. Double check all connections carefully. Make sure, using an ohmmeter, that none of the wires are shorted to each other. Make sure you didn't induce any shorts when you soldered the wires to the mainboard on your 9x. The connectors on the end of the 5x2 cable between your programmer and 9x, and the connector inside the 9x can get stray strands of wire shorting between the wires where they are clamped into the connectors.
 - b. The power switch on the AVR Pocket Programmer is not set to the left/”Power Target” setting.
 - c. Your usb port does not provide enough power to power up the programmer and 9x together. Try another port or another computer. Try a powered usb hub, and plug in thru that.
 - d. There have been some reports that early AVR Pocket Programmers might not have provided the correct power to the transmitter. Using an unpowered (!) usb hub generally corrected these problems.
 - e. The “fuses” in the microprocessor in the 9x got set to the wrong values. See section 4.
 - f. The circuit boards on the Pocket Programmer and the Protostack/ebay programmer are not insulated. Make sure you don't have them on a metallic surface shorting them out, or have them contaminated on them shorting between a couple of components or traces on the board.
3. You get errors in the status box that pops up while eePe is trying to program:
 - a. While using the Protostack/ebay programmer, you see a warning about the sck couldn't be set. This is normal, don't worry about it!!
 - b. You get a “Yikes!” message and it says it didn't get the correct signature. This means your programmer was unable to communicate properly with your 9x. Check your wiring; unplug your programmer from the usb port and plug it back in and try again. Your “fuse” settings may also be incorrect.
 - c. You get a verify error after a write to either the flash or eeprom. Try flashing again. Unplug/replug your programmer from the usb port.
4. Fuse settings:
 - a. This is the most difficult area to write about. We are just beginning to understand how to correct for this kind of error. Depending on what happens, this can lead to the 9x only working with certain firmware, working at about 1/8th its normal speed, to not responding at all. In extreme cases, the 9x can become what is known as a “fuse brick”, and may require

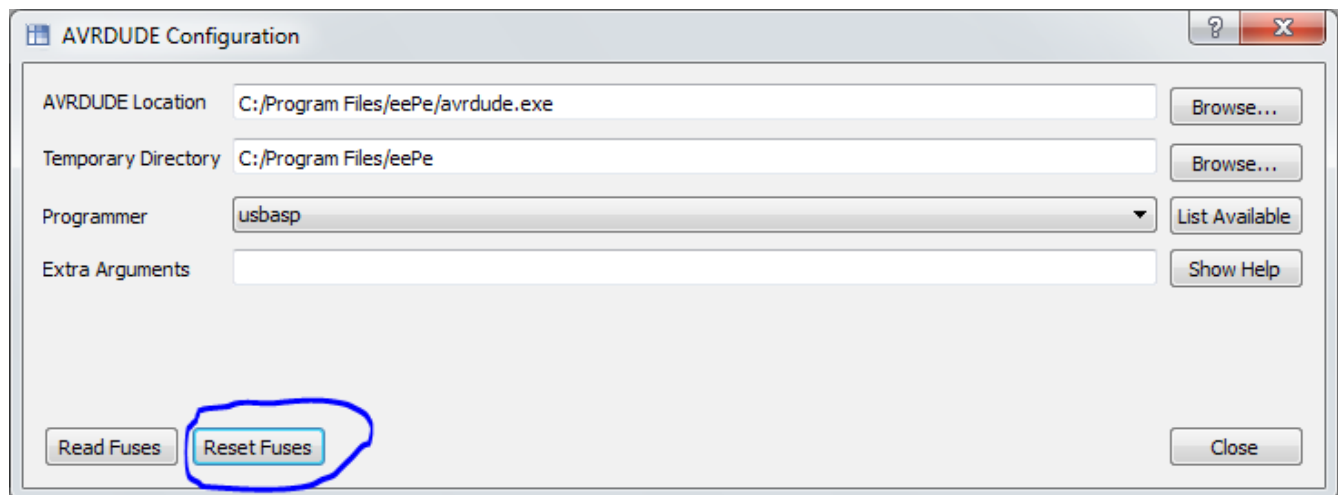
extraordinary means to revive it. The 9x uses an AVR microprocessor, and they are known to exhibit these problems occasionally while flashing.

b. Check your fuse settings. Click on the “configure” icon in eePe. Hook up your programmer to your 9x and hit the “Read Fuses” button.

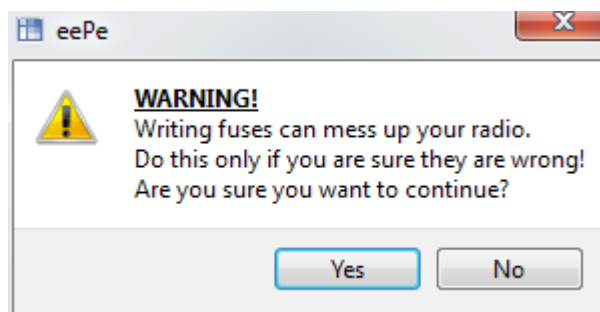




Your fuses should read EXACTLY as the file above. If they don't, do the following:



You will get the following warning, but since your fuses are messed up, hit "Yes".



Hopefully, that will bring your 9x back to normal and you can use or reprogram it.

5. If that doesn't work, you cannot communicate with your 9x at all, you have tried a different programmer, a different computer, and things were working, then suddenly didn't when you tried to flash firmware or an eeprom, you are getting into somewhat uncharted territory. Telemachus and others on RC Groups are working on solutions. Telemachus has written a guide, here:

<http://er9x.googlecode.com/svn/trunk/doc/Fuse%20Brick.pdf>

He and others have had some success using an external oscillator. He has had some further issues, so I suggest doing a search in the firmware forum on his name for the latest information.

This isn't meant to scare you. I have flashed four 9x's, two of my own, one for my Dad, and one for a friend. In the "early days" (spring of 2010!), I had a fuse issue that made my 9x only work with certain firmware. I was able to bring it back by resetting the fuses to the correct settings. I believe I induced the problem by accidentally working on faulty programmer cable while it was attached to my 9x and the transmitter was powered up. I've not had a problem since, and I've flashed one of my 9x's dozens of times since. What is causing issues for other people is unknown. You've been warned!!

Conclusion

That's all folks! If you have questions about this or other firmware questions, visit the thread on RC Groups, here:

<http://www.rcgroups.com/forums/showthread.php?t=1266162&page=28#post15803337>

Version History

Version 2.1, 17 Oct 2010; Changed some photos, typo corrections, more cleanup.

Version 2.0, 17 Oct 2010; Complete revision. Added Protostack/ebay programmer to guide, used eePe as programming software of choice, added trouble shooting section. Incorporated 64bit/x64 Windows appendix into main body of this guide.

Version 1.2.1, 20 Sep 2010; Fixed broken Protostack link.

Version 1.2, 20 Sep 2010; revised to add different instructions for calibrating er9x, change from .bin to .hex for er9x, and abbreviated instructions for x64 bit systems.

Version 1.1, 17 Aug 2010; original release

Written on Open Office 3.2.1, using Windows 7 Snipping Tool for screenshots.

Appendix

Installing the AVR Pocket Programmer Drivers on a Vista/Windows 7 x64 System

Unfortunately, Microsoft has seen fit to make installation of so-called unsigned drivers on x64 systems a real pain. Unless a driver is signed, a process which can cost a developer several hundred dollars to let Microsoft “bless” it, it can be put on the system, but Windows refuses to use it because it may be a security risk. Why this is different than a x86 system is beyond me. Anyway, the Pocket Programmer drivers are unsigned as are most others for programmers, EXCEPT for the Protostack/ebay programmer discussed in the main guide. Several people have asked Sparkfun to make a signed driver, but they haven't done so yet, and I wouldn't hold your breath.

My recommendation is to skip this headache if you have a 32bit system you can use with the Sparkfun programmer, or get the Protostack/ebay programmer. If the only thing you've got is a x64 system, and you already have the Sparkfun programmer, then you will have to go thru this.

I have not included all of the dialog boxes and graphics in this guide. If there is enough demand, I will do so. But if you are here, you are willing to wade in pretty deep into the bowels of Windows anyway, so following these instructions shouldn't phase you.

You will need two more programs to fool Windows into thinking your driver is signed, and get rid of a “watermark” that Windows puts on your desktop as a result of this fix. Get the following programs:

Driver Signature Enforcement Overrider 1.3b, available here:

<http://www.softpedia.com/get/Tweak/Video-Tweak/Driver-Signature-Enforcement-Overrider.shtml>

RemoveWatermark

http://soft3.wmzhe.com/download/deepxw/RemoveWatermark_20090509.zip

I successfully did this on an x64 computer I have. There have been no ill effects of doing this, but your experiences may be very different from mine.

Instructions:

Follow my guide for 32 bit systems for installing the driver for the Pocket Programmer. Windows will still complain about unsigned drivers, and will install the the x64 version, and you will think all is good. But, it won't actually use them. They will be blocked from working, if you look in the Device Properties for the Pocket AVR.

First, go to the Start menu, and select items in the following order:

Start/Control Panel/User Accounts/Change User Account Control settings

In the resulting dialog box, set the control to “Never Notify”. You can change this back later if you wish.

Right click on the Driver Signature Enforcement Overrider (dseo) Program that you downloaded, titled “dseo13b.exe”. Select “Run as Administrator” in the menu that comes up. You will get the usual Windows warning, click thru it, and then you will get a couple of windows from dseo itself. Good reading if you are so inclined. Anyway, click thru them, and you will get a dialog box with a bunch of buttons.

Select “Enable TestMode”, and click Next.

You will get a dialog box, saying that TestMode has been enabled, and restart the computer. Go ahead and do so now.

After restarting the computer, restart dseo as an administrator, and select the button “Sign a System File”, and click Next. You will now get a dialog box where you have to type in the complete path to the file you want to sign. So what files do you sign? Well, go to the Device Manager in the Control Panel, and click on “LibUSB-Win32 Devices”. “AVR Pocket Programmer” should now appear. Double click on it, and select the “Driver” tab in the resulting window. Click on the “Driver Details” button in the resulting window. The three driver files there, with their paths (e.g. c:\Windows\System32\drivers\libusb0.sys) are the ones you have to type in the the dseo window. I haven't been able to figure out how to cut and paste them from the window, so just type them into dseo EXACTLY as they appear. Type in the first one, and hit “Ok”. You will get a window saying that it has been fixed, and that if you restart, it should load. Don't restart yet. Click ok, and do the same thing for the other two files in the Driver File Details window. After you have done all three, restart your computer. Now if you go to the Device manager and look at the AVR Pocket Programmer info, it should be loaded.

HOWEVER, now you will probably have a watermark in the lower right hand portion of your screen with “Test Mode” and some other info in it. It won't harm anything, but if you want to get rid of it, do the following:

Open the folder for RemoveWatermark_20090509. I'd advise reading the ReadMeEn.txt file. Right click on the “RemoveWatermarkx64.exe” program you downloaded. It will start up a command line window. Type “y” . It will run a few seconds, and complete. Close the window, and restart your computer. If the watermark is still there, restart RemoveWatermark64, and hit “R” . It will take a few seconds and rebuild a cache. When it is done, either log off and back on, or restart. Now your watermark should be gone.

By the way, dseo has a button called “Remove Watermark” but all it does is take you to webpage to download RemoveWatermark.

The rest of my regular guide should now work ok.

Now wasn't that a pain in the butt? Why Microsoft made x64 different than x86, I don't know.

Version 2, 17 Oct 2010, minor revisions and incorporated into main guide.

Version 1, 20 Sep 2010