How To Patch The Xerox Phaser 3100MFP So That It Doesn't Ask For A Smart Card On Every Toner Change

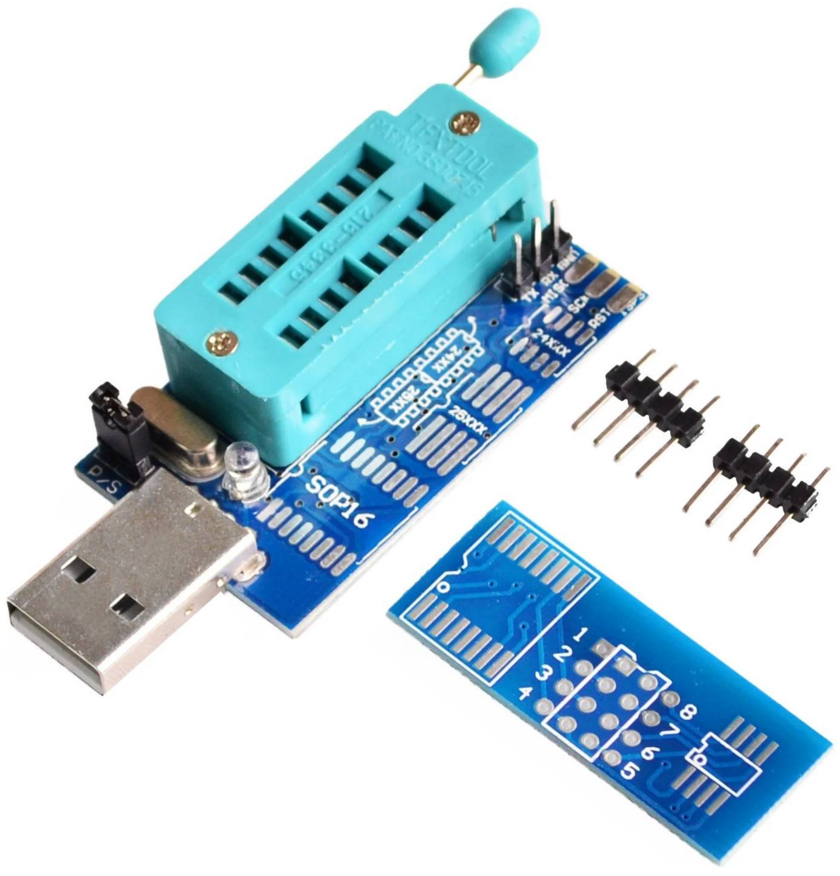
First, I have to note that this tutorial requires some knowledge in electronics (mainly soldering) and that it's not a software patch. Therefore, it's not exactly aimed at your every day user. If you're handy with a soldering iron, yes, this tutorial is for you ☺.

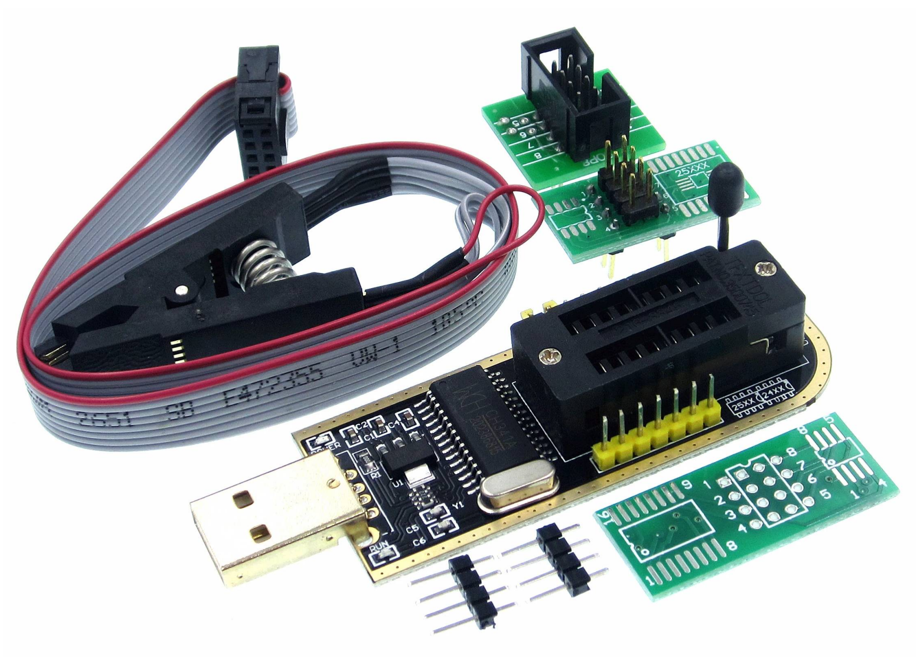
Also, I have to note that this hardware patch fools the printer into thinking that it's always got the same toner inside and that it's always 100% full. So, if the printer starts printing with a poorer print quality, then the toner's probably empty and you need to replace it. Also, as mentioned in the title, this patch doesn't require you to insert a smart card after you replace the toner (the toner's always at 100%, thus, you don't need to reset it), so you can use cheaper compatible toners that don't come with a smart card ☺.

Now, let's get started. There are some prerequsites that you'll require in order to apply this patch to the printer.

- **A soldering iron.** Preferably one with a small tip since you'll need it to remove one SMD chip on the mainboard. An SMD rework station would also be nice, but it's not necessary at all, everything can be done with a simple soldering iron ☺. I've also included a few clips that shows how to unsolder SMD components with a simple soldering iron, so even though special tools can be a plus, they aren't necessary.

- **An EEPROM programmer.** If you have an "advanced" one (like Willem EPROM Programmer or the MiniPRO TL866 one), they will do the job nicely. But, if you don't own any EEPROM programmer, look for the **CH341A Programmer** on AliExpress or eBay. They are dirt cheap and do the job really well (image below).



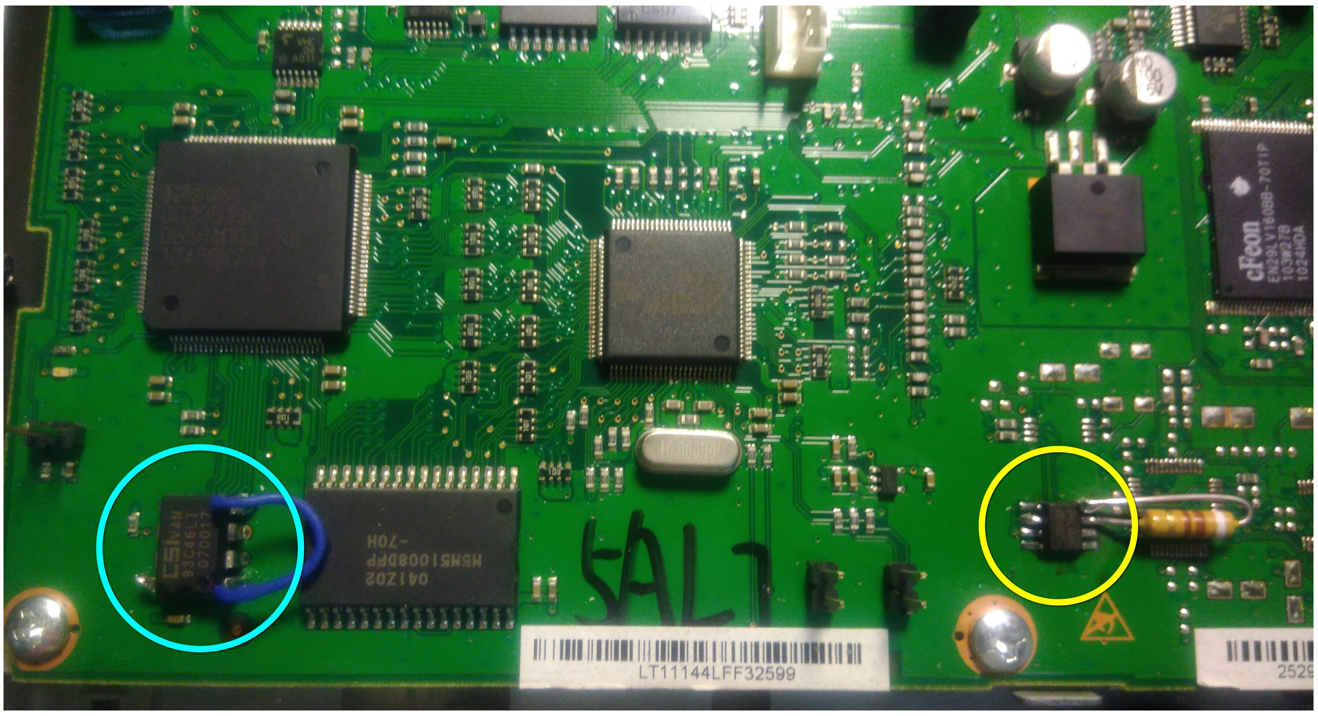


I also suggest you get a bundle with an SMD clip, as in the second image (the programmer with the black PCB). If not, you'll have to do a second job soldering the chip to the programmer, but if you don't plan on using the programmer in the future, that's fine as well.

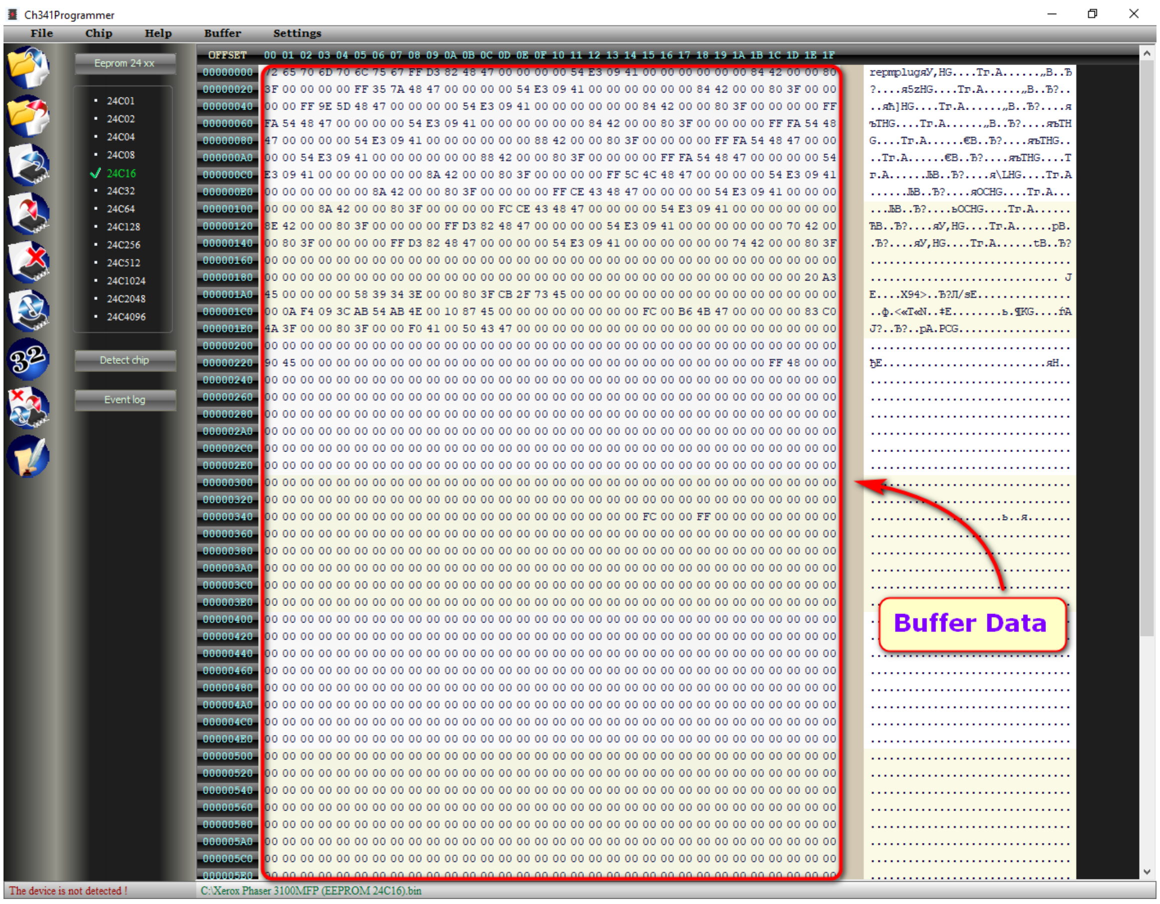
I've also included one of the applications for Windows that can be used with this type of programmer, as well as the Windows drivers for the programmer. Connect the programmer to the PC/laptop and load the drivers for the programmer via Device Manager. Use the programmer in EPP mode (there's a jumper that sets the programmer to work either in serial or parallel/EPP mode, it's probably marked as **P/S**) and load the drivers for the programmer while in EPP mode (there are different drivers for parallel and serial mode operation). If you're on Linux, I believe modern kernels (anything above Linux kernel 5.x) should include the drivers for this programmer by default. You can use an application like **flashrom** in Linux to program EEPROMs (or many other MCUs that have EEPROMs in them). Flashrom should be in your distro's default repository (I know Ubuntu has it, as well as Arch, Fedora, Suse and Void).

OK, now that we're set up, we can begin. Open up the right cover of the printer (image below). I've also included a video that covers the complete disassembly of a Xerox Laser 3100MFP, so if you don't know how, just watch the video.

Now, open up the metal covers and locate these two chips.



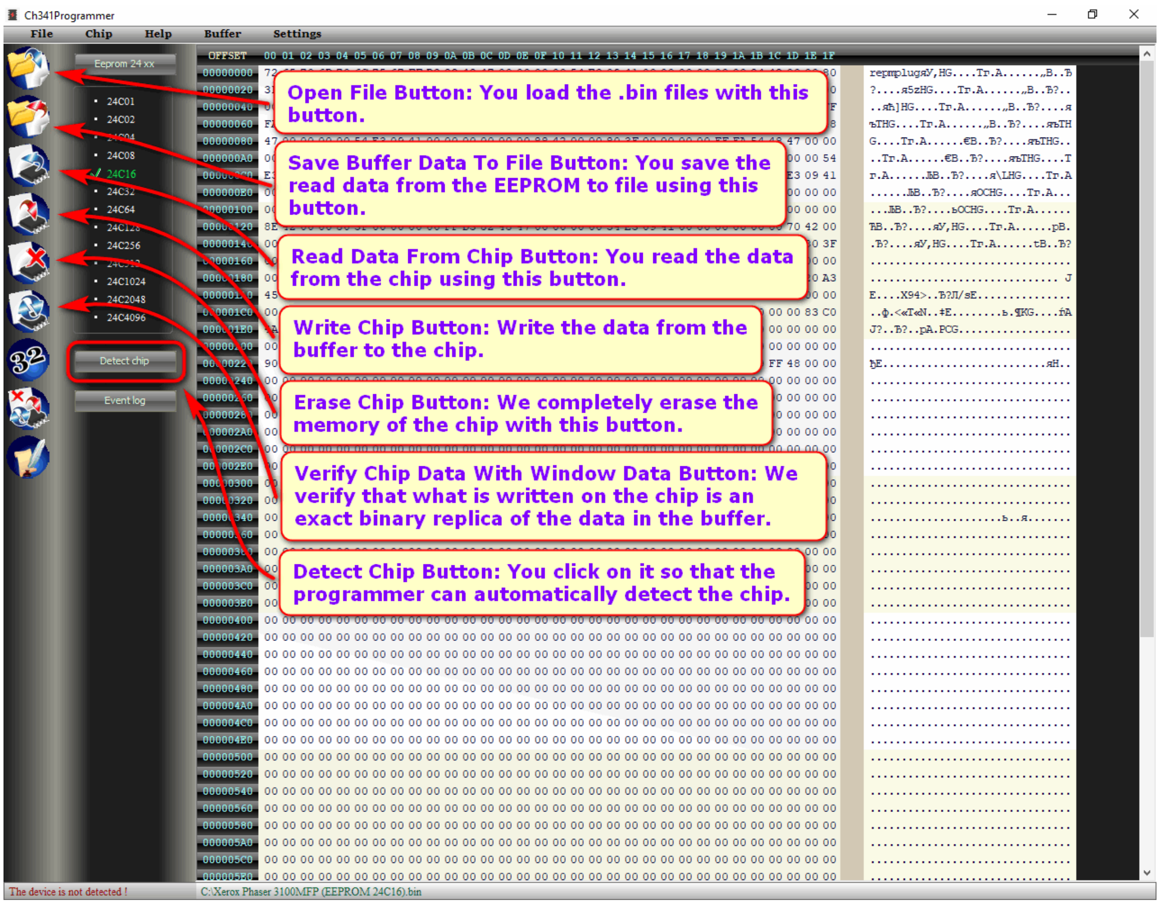
Both of them are EPROM (EEPROM) chips. The one in the yellow circle holds the data about how much toner is left in the cartridge. The one in the blue circle holds data about previously used cartridges (or at least I think so). So, what we need to do is “freeze” both of them (no new data gets written on both of them). But, before we freeze them, we need to “fill” the toner cartridge “full”, meaning we have to write data on the chip in the yellow circle (24C16) that “tells” the MCU that the toner’s full. In order to do this, we need to desolder the chip and write the data on it, then solder it back. I already mentioned that I’ve included some clips that depict how you can desolder SMD components with just a simple soldering iron, so if you’re not familiar with soldering or desoldering SMD components, take a look at the clips and practice on something (an old motherboard would do just fine ☺). After we’ve desoldered the chip, we have to connect it to the programmer in order to write the data on it that “tells” the MCU/CPU that the toner is full. I’ll be using the **CH341A Programmer** hardware with the **Ch341Programmer** application.



```
OFFSET    00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
00000000  72 65 70 6D 70 6C 75 67 FF D3 82 48 47 00 00 00 00 54 E3 09 41 00 00 00 00 00 00 84 42 00 00 80  repmplugяУ,HG....Тг.А......„B..Ђ
00000020  3F 00 00 00 00 FF 35 7A 48 47 00 00 00 00 54 E3 09 41 00 00 00 00 00 00 84 42 00 00 80 3F 00 00  ?....я5zHG....Тг.А......„B..Ђ?..
00000040  00 00 FF 9E 5D 48 47 00 00 00 00 54 E3 09 41 00 00 00 00 00 00 84 42 00 00 80 3F 00 00 00 00 FF  ..яћ]HG....Тг.А......„B..Ђ?....я
00000060  FA 54 48 47 00 00 00 00 54 E3 09 41 00 00 00 00 00 00 84 42 00 00 80 3F 00 00 00 00 FF FA 54 48  ъTHG....Тг.А......„B..Ђ?....яъTH
00000080  47 00 00 00 00 54 E3 09 41 00 00 00 00 00 00 88 42 00 00 80 3F 00 00 00 00 FF FA 54 48 47 00 00  G....Тг.А......€B..Ђ?....яъTHG..
000000A0  00 00 54 E3 09 41 00 00 00 00 00 00 88 42 00 00 80 3F 00 00 00 00 FF FA 54 48 47 00 00 00 00 54  ..Тг.А......€B..Ђ?....яъTHG....T
000000C0  E3 09 41 00 00 00 00 00 00 8A 42 00 00 80 3F 00 00 00 00 FF 5C 4C 48 47 00 00 00 00 54 E3 09 41  г.А......ЉB..Ђ?....я\LHG....Тг.А
000000E0  00 00 00 00 00 00 8A 42 00 00 80 3F 00 00 00 00 FF CE 43 48 47 00 00 00 00 54 E3 09 41 00 00 00  ......ЉB..Ђ?....яОCHG....Тг.А...
00000100  00 00 00 8A 42 00 00 80 3F 00 00 00 00 FC CE 43 48 47 00 00 00 00 54 E3 09 41 00 00 00 00 00 00  ...ЉB..Ђ?....ьОCHG....Тг.А......
00000120  8E 42 00 00 80 3F 00 00 00 00 FF D3 82 48 47 00 00 00 00 54 E3 09 41 00 00 00 00 00 00 70 42 00  ЋB..Ђ?....яУ,HG....Тг.А......pB.
00000140  00 80 3F 00 00 00 00 FF D3 82 48 47 00 00 00 00 54 E3 09 41 00 00 00 00 00 00 74 42 00 00 80 3F  .Ђ?....яУ,HG....Тг.А......tB..Ђ?
00000160  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
00000180  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 20 A3  .............................. J
000001A0  45 00 00 00 00 58 39 34 3E 00 00 80 3F CB 2F 73 45 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  E....X94>..Ђ?Л/sE...............
000001C0  00 0A F4 09 3C AB 54 AB 4E 00 10 87 45 00 00 00 00 00 00 00 00 FC 00 B6 4B 47 00 00 00 00 83 C0  ..ф.<«T«N..‡E........ь.¶KG....ѓA
000001E0  4A 3F 00 00 80 3F 00 00 F0 41 00 50 43 47 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  J?..Ђ?..pA.PCG..................
00000200  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
00000220  90 45 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF 48 00 00  ђE..........................яH..
00000240  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
00000260  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
00000280  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
000002A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
000002C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
000002E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
00000300  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
00000320  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
00000340  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FC 00 00 FF 00 00 00 00 00 00 00  .....................ь..я.......
00000360  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
00000380  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
000003A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
000003C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
000003E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
00000400  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
00000420  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
00000440  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
00000460  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
00000480  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
000004A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
000004C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
000004E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
00000500  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
00000520  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
00000540  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
00000560  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
00000580  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
000005A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
000005C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
000005E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................................
```

The device is not detected !

C:\Xerox Phaser 3100MFP (EEPROM 24C16).bin



Open File Button: You load the .bin files with this button.

Save Buffer Data To File Button: You save the read data from the EEPROM to file using this button.

Read Data From Chip Button: You read the data from the chip using this button.

Write Chip Button: Write the data from the buffer to the chip.

Erase Chip Button: We completely erase the memory of the chip with this button.

Verify Chip Data With Window Data Button: We verify that what is written on the chip is an exact binary replica of the data in the buffer.

Detect Chip Button: You click on it so that the programmer can automatically detect the chip.

The device is not detected !

C:\Xerox Phaser 3100MFP (EEPROM 24C16).bin

A general overview of the programmer's user interface window is given in the images above.

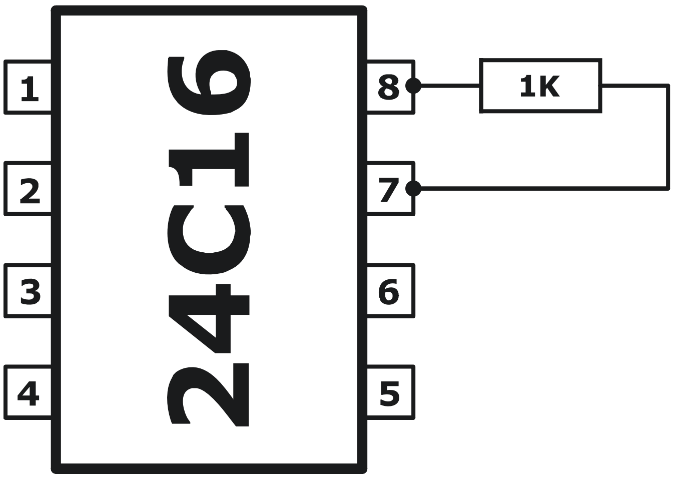
First, we need to select the chip that we've got connected to the programmer, so that the programmer knows how to read/write data from/to the chip. The **Detect Chip** button can do this automatically, but in case it doesn't detect our chip, hit the chip selector button till the button reads **EEPROM 24 XX**. From the menu below the chip selector button select **24C16** (this is the EEPROM chip we've got loaded in the programmer).

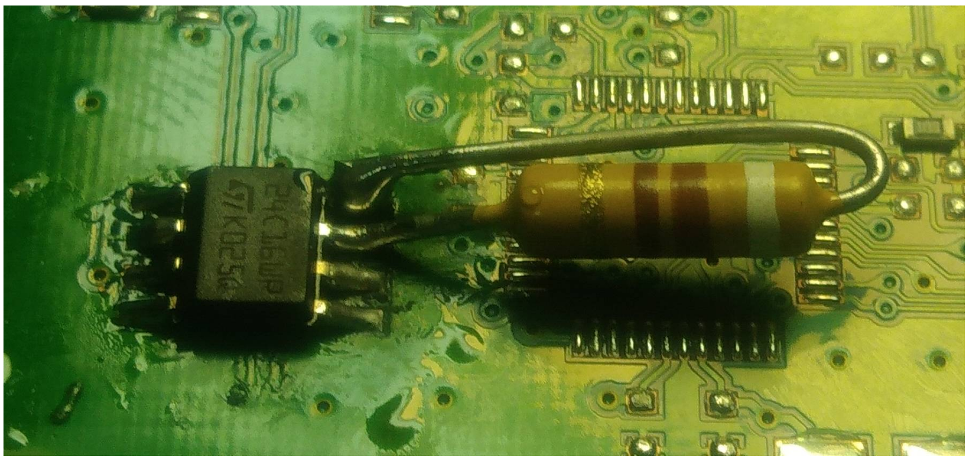
Next, we need to save the data from the chip, in case something goes wrong or just doesn't work. So, we hit the **Read Data From Chip** button so that the data that's located in the chip can be read to the application's buffer. After the programmer finishes reading the data (there's a progress window), we'll save the buffer data to file using the **Save Buffer Data To File** button. A window will pop up asking where to save the file. The location and name don't matter as long as you know where you saved them and the filename under which the data is saved. I usually use **backup.bin** for the filename.

It's also preferable to completely erase the chip afterwards. We'll use the **Erase Chip** button for this. After we hit the button, a progress window will appear indicating the progress of the operation. When the operation is over, just close the window.

Now, we'll load the data to buffer that "tells" the MCU/CPU of the printer that the toner cartridge is full. The file that we need to load is **Xerox Phaser 3100MFP (EEPROM 24C16, 100% Toner Level).bin** (you can find the file in the **Dumps.7z** archive). The content of the buffer data will change when we load this file in the application. Next, we need to write this data to the chip. We use the **Write Chip** button for this. A progress window will pop up indicating when the operation is over. It's also preferable to verify that the data in the chip an exact binary replica of the data in the buffer. We'll use the **Verify Chip Data With Window Data** for this. If the application reports that the data on the chip is identical to the data in the buffer, that means that the write operation completed successfully. Just close the pop up window with the progress bar, that's it ☺.

Next, we need to do some soldering work ☺. Remove the 24C16 EEPROM from the programmer (desolder it, or if you were using clips, remove the chip from the clip) and solder it back on the main board of the printer. But, we'll also add a resistor that prevents new data being written to the chip, so that the MCU/CPU "thinks" that the toner is always full. This *hack* basically freezes the chip, the toner is always at 100% ☺. The value of the resistor is 1K and it's added between pins 7 and 8 of the chip, as depicted in the images below.

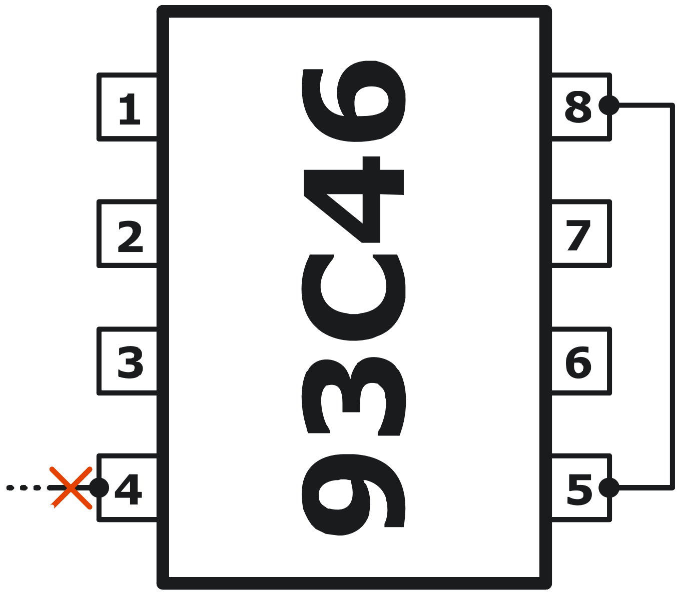


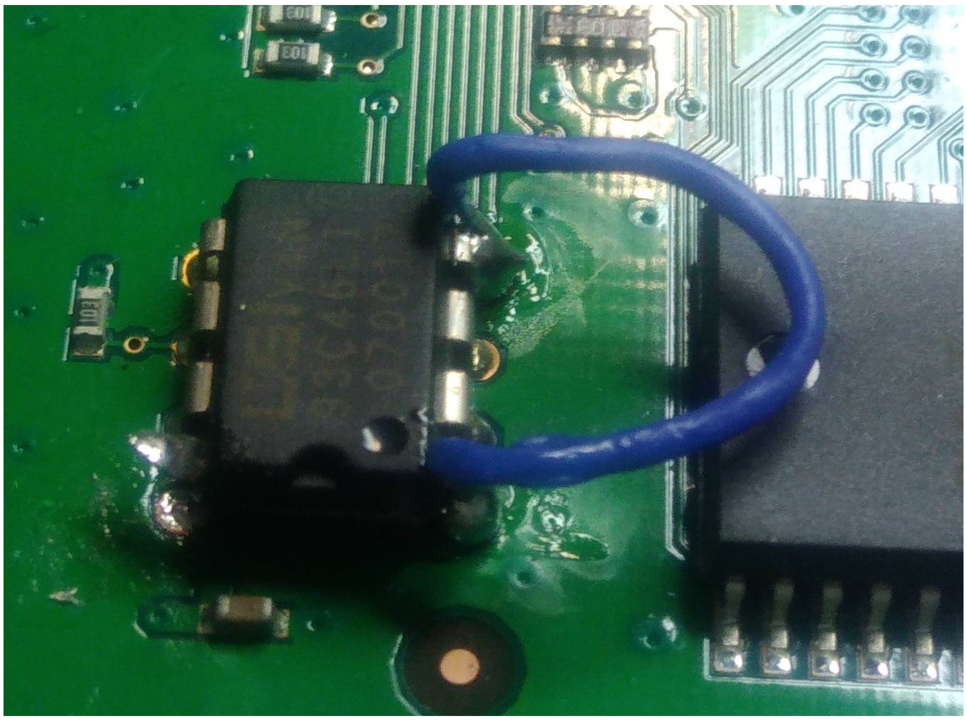


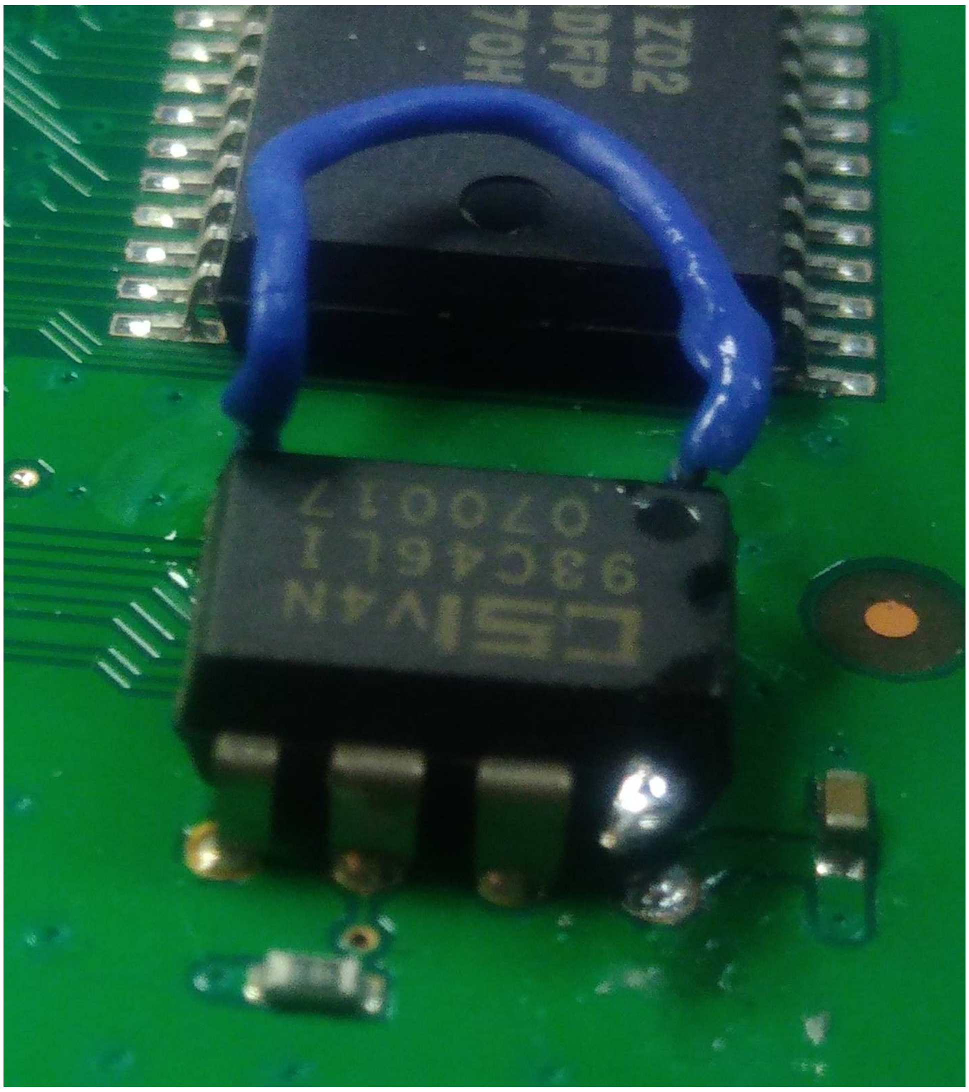
I used a 910Ω resistor, I didn't have a 1K one at my disposal, LOL ☺. The value is close enough though and the value is not really crucial, 1K, 1.1K, 1.2K, 910Ω, doesn't really matter as long as the value is close to 1K as possible ☺. The power rating of the resistor should be 0.25W. A higher value can be used, but not a lower one (haven't tested this with a 0.125W resistor… my guess is, it should work fine as well, but just in case, keep the power rating of the resistor to 0.25W).

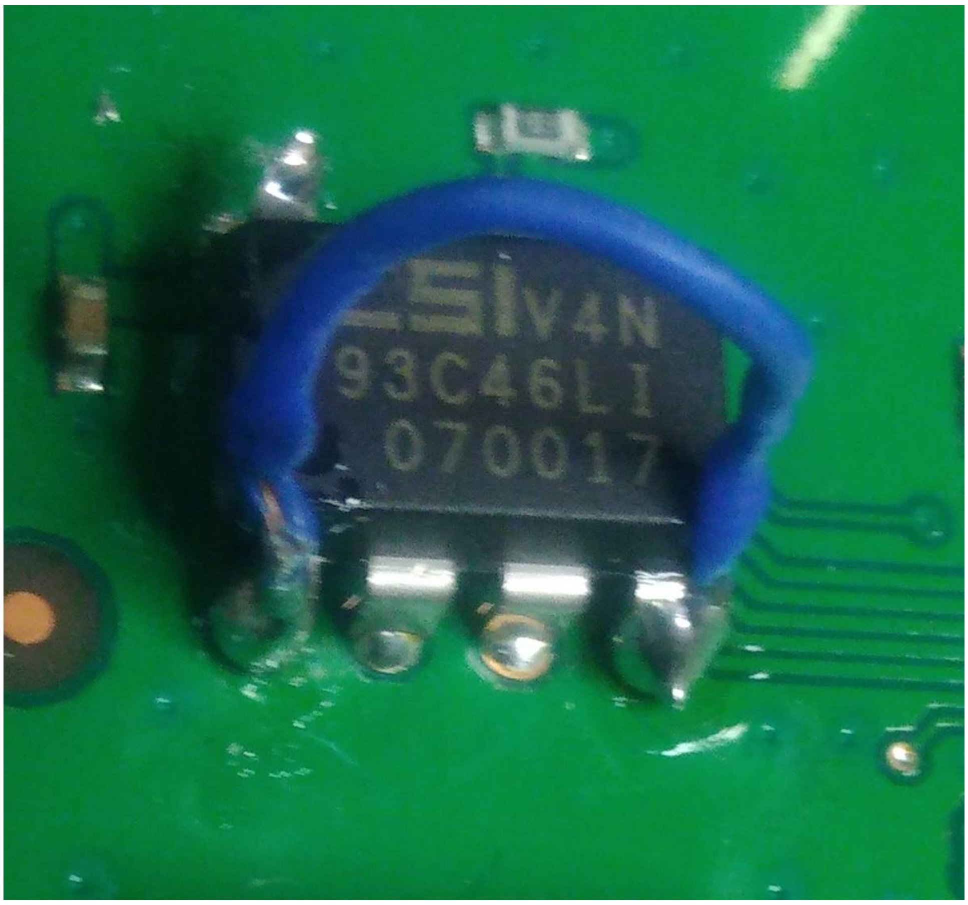
Next, we need to *patch* the other EEPROM, the **93C46** one. As noted previously this *patch* will lock down the memory of the 93C46, freezing it in time, thus every time the data about the toner's level and the unique toner's ID is read from the 24C16, this will be a "new toner" for the MCU, which basically means that we can reload the same toner over and over again (fill it manually if you'd like doesn't matter, LOL ☺) and the toner will always be a "brand new one" for the MCU/CPU of the printer. Neat, huh ☺.

The next few images depict what needs to be soldered/desoldered/patched on the 93C46 EEPROM in order to freeze it in time (no new data gets written on it).




CSI
93C46


CSI
V4N
93C46LI
070017



Basically, we just need to short circuit pins 8 and 5 of the IC and cut off pin 4 from the main board, that's it ☺. You can cut pin 4 using small cutting pliers or anything that comes at hand. Just be careful not to damage the IC's package ☺. If you've got a hot air soldering station at your disposal, you can desolder the IC, raise the 4th leg and solder it back again without soldering the 4th leg on the board. Makes no difference how you do it, as long as pin 4 is not touching the board ☺.

I also forgot to mention that this *hack* works on any firmware version of the printer. I read online about patched versions of the firmware, but you have to downgrade the firmware via FTP (older versions were patched, newer versions of the firmware weren't, so you downgrade the printer to a lower firmware version). This does work, but the downside is that you have to have a Phaser 3100MFP with a NIC (network card). My version of the 3100MFP didn't have a NIC, so I couldn't downgrade the firmware via FTP, and for some reason, the firmware upgrade tools included in the Xerox Companion Suite didn't let me downgrade the firmware version in the printer, even though (as far as I know) there are no firmware downgrade restrictions in this MFP device ☹. My guess is, it was the OS that blocked the firmware downgrade process (the Xerox Companion Suite is old and built back in the Windows XP/Vista/7 days and was meant to run on Windows 7 max, from what I read, while I installed it and ran it on Windows 10). Maybe I should've tried downgrading from a VM (virtual machine), but, oh well, things worked out even better this way, since this hack works on any firmware version ☺.

Comments or suggestions, write me at 0x4e4f@gmail.com or on Reddit (u/PCChipsM922U).

Download links for the full archive:

https://cloud.mail.ru/public/uFAz/tvcKgSvAV
https://drive.google.com/file/d/1_p_pi48ZiH5uIwqO96dc0xn_ebhoJXuZ
https://mega.nz/file/XsVVlRIJ#ptpU9R8cKOaSLhPhxK9nVkF7ReagDK6c1gqZ42hVZV8