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 O.

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.

Ch341Pr	rogrammer			– a ×
File	Chip Help	Buffer	Settings	
2	Eeprom 24 xx	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	A Nerror MC To J R T
		00000020	3F 00 00 00 00 FF 35 7A 48 47 00 00 00 00 54 E3 09 41 00 00 00 00 00 84 42 00 00 80 3F 00 00	?
1	 24C01 24C02 	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	
	• 24C04	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	GTr.A€BЪ?яъTHG
2	 24C08 ✓ 24C16 	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 54 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	Tr.A€Bb?яъTHGT r.AIbBb?
	• 24C32	000000E0	00 00 00 00 00 00 8A 42 00 00 80 3F 00 00 00 FF CE 43 48 47 00 00 00 54 E3 09 41 00 00 00	
	 24C64 24C128 	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 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	
	• 24C256	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	.B?яУ, HGTr.AtBB?
	 24C512 24C1024 	00000160	00 00 00 00 00 00 00 00 00 00 00 00 00	
	 24C2048 	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	ЕХ94>Ъ?Л/яЕ
See.	• 24C4096	000001C0 000001E0	00 0A F4 09 3C AB 54 AB 4E 00 10 87 45 00 00 00 00 00 00 00 00 00 FC 00 B6 4B 47 00 00 00 00 83 C0 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	ф.<«T«N‡Еь.¶КБлА J?Ъ?рА.РОБ.
02	Datact chin	00000200	00 00 00 00 00 00 00 00 00 00 00 00 00	
3		00000220	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	<u>д</u> гнп
2	Event log	00000260		·····
and .		00000220	00 00 00 00 00 00 00 00 00 00 00 00 00	·····
		000002C0		
		00000300	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	
		00000360	00 00 00 00 00 00 00 00 00 00 00 00 00	
		00000380 000003A0	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	·····
		000003E0 00000400	00 00 00 00 00 00 00 00 00 00 00 00 00	Buffer Data
		00000420	00 00 00 00 00 00 00 00 00 00 00 00 00	Buller Data
		00000440		
		00000480	00 00 00 00 00 00 00 00 00 00 00 00 00	·····
		000004A0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
		000004E0	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	·····
		00000540		
		00000580	00 00 00 00 00 00 00 00 00 00 00 00 00	· · · · · · · · · · · · · · · · · · ·
		000005A0 000005C0	00 00 00 00 00 00 00 00 00 00 00 00 00	
The desire it		00000520		✓
	a not detected :	I TIX PROV Place	OC A HEIDARE (REEKLIDA, ALLID) DO	
110 0011001		C:tXerox Phas	er stuumer (EEPRUM 24C16).0m	
Ch341Pr	rogrammer	C:tXerox Phas	PESTOWARP (EEEROM 24010).0m	- 0 ×
Ch341Pr File	rogrammer Chip Help	Buffer	Settings	- 0 X
Ch341Pr File	rogrammer Chip Help Eeprom 24 xx	Buffer OFFSET 00000000	Settings 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 16 19 1A 1B 1C 1D 1B 1F 72	- 0 ×
Ch341Pr File	Chip Help Exprom 24 xx	Buffer OFFSET 00000000 00000020	Settings 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 7 Open File Button: You load the .bin files with this	- 0 × repmpluga0,HSTr.A,BB.
Ch341Pr File	Chip Help Eeprom 24 xx • 24C01 • 24C02	Buffer OFFSET 0000000 0000020 0000040 00000040	Settings 00 01 02 03 04 05 06 07 08 09 0A 08 0C 00 05 0F 10 11 12 13 14 15 16 17 18 19 1A 18 1C 1D 1F 1F 7 3 3 4 5 5 5 5 5 5 5 5 5 5 5 5 5	- 0 ×
Ch341Pr File	Ecprom 24 xx 24001 24002 24003	Buffer OFFSET 0000000 0000000 00000000 0000000 00000000 0000000 00000000 0000000	Settings 00 04 05 06 07 08 08 00 01 12 13 14 15 16 17 18 19 14 16 17 18 19 14 16 17 18 19 14 16 17 18 19 14 16 17 18 19 14 16 17 18 12 14 15 16 17 18 12 10 <t< th=""><th>- C ×</th></t<>	- C ×
Ch341Pa File	Chip Help Exprom 24 xx - - 24001 - - 24002 - - 24003 - - 24004 -	Buffer OFFSET 00000000 0000000 00000000 00000000 0000	Settings 00 01 02 03 05 06 07 08 07 08 08 10 11 12 13 14 15 16 17 18 19 14 10 10 11 12 13 14 15 16 17 18 19 14 10 <t< th=""><th>- C ×</th></t<>	- C ×
Ch341Pr File	Chip Help Exprom 24 xx - - 24001 - - 24002 - - 24003 - - 2405 - - 2405 - - 2405 - - 2405 - - 2405 - - 2405 -	Citeros Pas Buffer 052552T 00000000 00000000 00000000 00000000 0000	Settings 00 01 02 03 03 05 06 07 08 03 0A 05 0C 00 05 07 10 11 12 13 14 15 16 17 18 19 1A 15 1C 1D 18 17 Open File Button: You load the .bin files with this button. 70 00 71 00 00 72 Save Buffer Data To File Button: You save the 73 read data from the EEPROM to file using this button.	- 0 ×
E Ch341Pr File	Chip Help Exprom 24 xx - - 24001 - - 24001 - - 24001 - - 24001 - - 24005 - - 2405 - - 2405 - - 2405 - - 2405 - - 2405 - - 2405 - - 2405 - - 2405 - - 2405 - - 2405 -	Buffer OFFSET 0000000 0000000 0000000 0000000 0000000 0000000 0000000 0000000 0000000 00000000 0000000 0000000 0000000 0000000 0000000 00000000 00000000	Settings 00 01 02 03 03 04 05 06 07 08 09 0A 05 0C 00 05 07 10 11 12 13 14 15 16 17 10 19 1A 15 1C 1D 18 17 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. 00 00 04 00 00 04 00 00 00 04 00 00 00 04 00 00 00 04 00 00 00 04 00 00 00	- 0 ×
Ch341Pr File	Chip Help Exprom 24 xx - - 24001 - - 24002 - - 24003 - - 2405 - - 2405 - - 2405 - - 2405 - - 2405 - - 2405 - - 2405 - - 2405 - - 2405 - - 2405 - - 2405 - - 2405 -	Buffer OFFSET 0000000 00000040 00000040 00000040 000000	Settings 00 01 02 03 04 05 06 07 08 03 0A 05 0C 00 05 07 10 11 12 13 14 15 16 17 10 15 1A 15 1C 10 15 14 15 16 17 10 15 1A 15 1C 10 15 17 10 15 1A 15 1C 10 15 1F 17 7 7 7 7 7 7 7 7 7 7 7 7 7	- 0 ×
Ch341Pr File	Chip Help Exprom 24 xx - - 24001 - - 24002 - - 24003 - - 2405 -	Buffer 0/2/53/1 00000000 00000020 00000000 00000000 000000	Settings 00 01 02 03 04 05 06 07 08 03 0A 05 0C 0D 05 07 10 11 12 13 14 15 16 17 18 15 1A 15 1C 1D 1F 17 7 7 7 7 7 7 7 7 7 7 7 7 7	- C ×
Ch341Pi File Pile	Chip Help Exprom 24 xx 24001 2 24001 24002 2 24002 24004 2 2403 24036 2 24036 24036 2 24036 24036 2 24036 24036 2 24036 24036 2 24036 24036 2 24036 24036 2 24036 24036 2 24036 24036 2 24036 24036	E.G. Georges C.G.	Settings Op of 02 02 03 04 00 06 07 08 09 0A 08 0C 0D 05 07 10 11 12 13 14 15 16 17 18 19 1A 18 1C 10 16 17 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. 00 00 00 00 00 00 00 00 00 00 00 00 00	- C ×
Ch341P- File P	Chip Help Exprom 24 xx 24001 24001 24002 24002 24003 24008 24016 24008 24026 24008 24026 24024 24026 24025 24026 24026 24026 24026 24026 24026 24026 24026 24026 240204 240204 240204 24004 240096 24004	E.G. Actor Part Buffer 075387 000000040 00000040 000000050 000000050 000000050 000000050 000000050 000000050 000000050 00000010 0000010 0000010 0000010 0000010 0000010 0000010 0000010 0000010 0000010 0000010 0000010 0000010 0000010 0000010 0000010 0000010 0000010 0000000 0000000 0000000 0000000 000000	Settings Op of 02 05 04 00 06 07 08 05 04 00 06 07 10 11 12 13 14 15 16 17 18 15 14 15 16 17 18 15 14 15 16 17 18 15 14 15 16 17 18 15 14 15 16 17 18 15 14 15 16 17 18 15 14 15 16 17 18	- C × repuplog#/HSTr.A,BB. ?s5zHSTr.A,BP. s5zHSTr.A,BP. s5zHSTr.A,BP. sHHSTr.A,BP. sHHSTr.A,BP. sHHSTr.A,BP. sHHSTr.A,BP. sHHSTr.A,BP. sHHSTr.A,BP. sHHSTr.A,BP. sHHSTr.A,BP. sHHSTr.A,BP. sHHSTr.A,BP. sHHSTr.A,BP. sHHSTr.A,BP. sHHSTr.A,BP. sHHSTr.A,B. sHHSTR sHHSTR sHHSTR sHHSTR sHHSTR sHHSTR sHHSTR sHSTR sHHSTR sHHSTR sHSTR sHHSTR sHSTR sHSTR sHSTR sHSTR sHSTR sHSTR sHSTR sHSTR sHSTR sHSTR sHSTR sHSTR sHSTR sHSTR sHSTR sHSTR sHSTR sHSTR sHSTR
E Ch341P/ File	Chip Help Exprom 24 xx 24001 24001 24002 24002 24003 24008 24016 24028 24026 24026 24026 24026 24026 24026 24026 24026 24026 24026 24026 24026 24026 240208 240208 240208 240208 240208 24008 240096 24008 Detect chip Detect chip	E.G. 4000 PMI	Settings Op on 02 02 03 04 00 06 07 08 09 0A 08 00 00 02 07 10 11 12 13 14 15 16 17 18 19 1A 18 10 10 10 11 12 13 14 15 16 17 18 19 1A 18 10 10 10 11 12 13 14 15 16 17 18 19 1A 18 10 10 10 10 10 10 10 10 10 10 10 10 10	- C × repuplug#/HSTr.A,BB. ?s5zHSTr.A,BP. s5zHSTr.A,BP. s5zHSTr.A,BP. stHSTr.A,BP. stHSTr.A,BP. stHSTr.A,BP. stHSTr.A,BP. stHSTr.A,BP. stHSTr.A,BP. stHSTr.A,BP. stHSTr.A,BP. stHSTr.A,B. stHSTR stHSTR stHSTR stHSTR stHSTR stHSTR stHSTR stHSTR stHSTR stHSTR stHSTR stHSTR stHSTR stHSTR stHSTR stHSTR stHSTR stHSTR stHSTR
E Ch341P/ File	Chip Help Exprom 24 xx 2 44001 2 44002 2 44002 2 44003 2 4403 2 44034 2 44044 2 44044 2 44044 2 44044 2 44044 2 44044 2 44044 2 4404	E.G. 4200 PMI	Settings Op on 00 00 00 00 00 00 00 00 00 00 00 00 00	- C × reproplugat/HSTr.A,BB. ?s5zHSTr.A,BB abHSTr.A,BB abHSTr.A,BB72stHS CTr.A€B72stHS Tr.AEB72stHS Tr.AEB72stHS BB72atHSTr.A BB72atHSTr.A BB72atHSTr.A BB72atHSTr.A BB72atHSTr.A BB72atHSTr.A BB72atHSTr.A BB72atHSTr.A BB72atHSTr.A BB72atHSTr.A BB72atHSTr.A BB72atHSTr.A BB72atHSTr.A BB72atHSTr.A BB72atHSTr.A BB72atHSTr.A BB72atHSTr.A BB72atHSTr.A BB72 BB7
E Ch341P/ File	Chip Help Egron 24 xx 2 4401 2 4402 2 4402 2 4403 2 4404 2 4404	E.G. 4200 PMI	Settings Op on 00 00 00 00 00 00 00 00 00 00 00 00 00	repmplug#/HSTr.A,BB. 7s5zHSTr.A,BB. 7s5zHSTr.A,BB. 8HBSTr.A,BB. 8HBSTr.A,BB. 8HBSTr.A,BB. 8HBSTr.A,BB. 8HS
E Ch341P/ File	Chip Help Egron 24 xx 2 4401 2 4402 2 4402 2 4403 4 2403 4 2403 2 4403 2 4403	E.G. Actor Past Buffer 017.537 000000040 00000040 00000040 000000050 000000050 000000050 000000170 00000170 00000140 00000000 00000000 00000000 0000000	Settings Open File Button: You load the .bin files with this button. Save Buffer Data To File Button: You save the pread 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. Buffer to the chip. Erase Chip Button: We completely erase the memory of the chip with this button.	repmplug#/HSTr.A,BB. ?s5zHSTr.A,BBP? s5zHSTr.A,BP? s5zHSTr.A,BP? s5zHSTr.A,BP? s5.HBSTr.A,BP? sb.HSTr.A,BP? sb.HSTr.A,BP? sb.P sb.P sb.P?stHSTr.A sb.P sb.P sb.P sb.P sb.P sb.P
E Ch341P/ File	Chip Help Egron 24 xx 2 4401 2 4402 2 4402 2 4403 4 2403 2 4403 2 4403	E.G. Actor Part Buffer 017.537 000000040 00000040 00000040 00000040 00000040 00000040 00000040 00000040 00000140 00000000 00000000 00000000 0000000	Settings Open File Button: You load the .bin files with this button. Save Buffer Data To File Button: You save the pread 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. Output Verify Chip Data With Window Data Button: We	repmplug#/HSTr.A,BB. 7s5zHSTr.A,BB. 8HSTr.A,BB. 8HSTr.A,BB. 8HSTr.A,BB. 8HSTr.A,BB. 8HSTr.A,BB. 8HSTr.A,BB. 9HSTr.A,BB. 9HSTr.A,BB. 9HSTr.A,BB. 9HSTr.A,BF?stHST 10
E Ch341P/ File File S S S S S S S S S S S S S S S S S S	Chip Help Egron 24 xx 2 44001 2 44002 2 44002 2 4403 2 4404 2 44	E.G. Georges Buffer D.F. 5.87 00000000 00000000 00000000 00000000 0000	Settings Op on 00 00 00 00 00 00 00 00 00 00 00 00 00	repmplugat/HSTr.A,BB. 7s5zHSTr.A,BBP s5zHSTr.A,BP s5zHSTr.A,BP s5zHSTr.A,BP s5.HBSTr.A,BP s5.HBSTr.A,BP
E Ch341P/ File Fil	Chip Help Exprom 24 xx 2 44001 2 44002 2 44002 2 4403 2 4404 2 4	E.G. Georges Buffer D.F. 5.87 00000000 00000000 00000000 00000000 0000	Settings Op on 00 00 00 00 00 00 00 00 00 00 00 00 00	repmpluga0/,H5Tr.A,BB.,B 745±H5Tr.A,BB.,B2,B 8HB5Tr.A,BB.,B2,B 8HB5Tr.A,BB.,B2,B 9HB5Tr.A,BB.,B2,B 9HB5Tr.A,BB.,B2,B 9HB5Tr.A,BB.,B2,B 9HB5Tr.A,BB2,B 9HB5Tr.A,BB2,B 9HB5Tr.A,BB2,B 10,180B2, 00CH5Tr.A,B 10,180B2, 00CH5Tr.A,B 10,180B2, 00CH5Tr.A,B 10,180B2, 00CH5Tr.A,B 10,180B2, 00CH5Tr.A,B 10,180B2, 00CH5Tr.A,B 11,180B2, 00CH5Tr.A,B 12,180,170,180Tr.A,180B2 13,180,170,180
E Ch341P/ File	Chip Help Expron 24 xx 2 44001 2 44002 2 44002 2 4403 2 44	E.G. Actor Part Buffer 017.537 000000040 00000040 00000040 00000040 00000040 00000040 00000040 00000040 00000140 00000000 00000000 00000000 0000000	Settings 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.	repmplugat/HSTr.A,BB. 7s5zHSTr.A,BBP mbHSTr.A,BP mb mbH mbH mbH mbH mbH mbH mbH
E Ch341P/ File	Chip Help Expron 24 xx 2 44001 2 44002 2 44002 2 44036 2 44056 2 44066 2 440666 2 44066666666666666666666666666666666666	E.G. Actor Past Buffer 013.537 000000040 00000040 00000040 00000040 00000040 00000040 00000040 00000040 00000040 00000040 00000140 00000000 00000000 00000000 0000000	Settings 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.	repmplugady,HSTr.A,BB.,B. 7s5zHSTr.A,BB.,P2s sHHSTr.A,BP2s sHHSTr.A,BP2s sHHSTr.A,BP2s sHHSTr.A,BP2s sHSTr.A,BP2s sHSTr.A,BP2s sHSTr.A,BP2s sHSTr.A,BP2s sHSTr.A,BP2s sHSTr.A,BP2s sHSTr.A,SBP2s sHSTr.A,SBP2s sHSTr.A,SBP2s sHSTr.A,SBP2s sHSTr.A,SBP2s sHSSs str.Ss str.S
E Ch341P/ File	Chip Help Expron 24 xx 2 44001 2 44002 2 44002 2 44034 2 44036 2 44066 2 440666 2 44066666666666666666666666666666666666	E.G. Actor Past Buffer 013-337 000000040 00000040 00000040 00000040 00000040 00000040 00000040 00000040 00000040 00000140 00000000 00000000 00000000 0000000	Settings 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.	repmplugady,HSTr.A,BB.,B. 7s5zHSTr.A,BB.,P2s sHHSTr.A,BB.,P2s sTHSTr.A,BP2s s
E Ch341P/ File	Chip Help Expron 24 xx 2 44001 2 44002 2 44002 2 44034 2 44036 2 44066 2 440666 2 44066666666666666666666666666666666666	E.G. Construction of the second secon	Settings Op of 02 05 04 05 06 07 08 09 0A 05 0C 0D 05 07 10 11 12 13 14 15 16 17 10 19 1A 15 1C 1D 1F 17 Op on 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.	repmplugady, HSTr.A,BB.,B 7s5zHSTr.A,BB.,P, s5zHSTr.A,BB.,P, s1HSTr.A,BB.,P, s1HSTr.A,BP, B
E Ch341P/ File	Chip Help Expron 24 xx 2 44001 2 44002 2 44002 2 44034 2 44036 2 44066 2 440666 2 44066666666666666666666666666666666666	E.G. Construction of the second secon	Settings 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.	repmplugs#/H5Tr.A,BB. 7s5zH5Tr.A,BB. 8
Ch341P/ File P	Chip Help Exprom 24 xx 2 44001 2 44002 2 44003 2 44034 2 44034 Detect dip Event log	E.C.G.Cod Past Buffer D.C.S.ST CODE0020 CODE000000 CODE000000	Settings Op oblic 02 05 06 07 06 09 05 00 00 00 00 00 00 00 00 00 00 10 11 12 13 14 15 16 17 10 19 14 15 16 10 11 12 13 14 15 16 17 10 19 14 15 10 10 11 12 13 14 15 16 17 10 19 14 15 10 10 10 10 10 10 10 10 10 10 10 10 10	repmplugs0/H5Tr.A,BB. 7stsH5Tr.A,BBP. stsH5Tr.A,BP. stsH5Tr.A,BP. stsH5Tr.A,BP. stsH5Tr.A,BP. stsH5Tr.A,BP. stsH5Tr.A,BP. stsH5Tr.A,BP. stsH5Tr.A,BP. sts.Tr.A,BP.
Ch341P/ File File S	Chip Help Ecoron 24 xx 2 4001 2 4002 4 2402 4 2403 4 240 4 2403 4 2403	E.C.G.Cod Past Buffer D.C.S.ST 00000020 00000020 00000020 00000020 00000020 00000020 00000020 00000020 00000100 0000110 0000110 0000110 0000110 0000110 0000110 0000010 0000010 0000010 0000010 0000010 0000010 0000010 0000020 00000000	Settings Op on 02 02 03 04 05 06 07 08 09 0A 08 0C 0D 02 07 10 11 12 13 14 15 16 17 10 19 1A 18 1C 10 1F 17 Op on 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 thip using this button. Write Chip Button: Write the data from the buffer to the chip. Frase 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.	repmplugs#, H5Tr.A,B7. s5zH5Tr.A,B72stH S.T. S. S.T. S.
Ch341P/ File File S	Chip Help Econo 24 xx 2 4001 2 4002 4 2401 2 4003 4 2403 4 2404 4 2404	E. C.	Settings Op oblic 02 05 06 07 08 09 05 00 00 00 00 00 00 00 00 00 10 11 12 13 14 15 16 17 10 19 14 15 16 10 11 12 13 14 15 16 17 10 19 14 15 16 10 11 12 13 14 15 16 17 10 19 14 15 16 10 10 11 12 13 14 15 16 17 10 19 14 15 16 10 10 10 10 10 10 10 10 10 10 10 10 10	reproplugady, HSTr.A,B7.
■ Ch341P File でう でう でう でう でう でう でう でう でう でう	Chip Help Econo 24 xx 2 4400 2 4400 2 4400 2 4400 4 2401 2 4402 4 2403 2 4403 2 4403	E. C.	Settings Op dit 02 05 06 07 08 09 05 00 00 00 05 07 10 11 12 13 14 15 16 17 10 19 14 15 16 10 11 12 13 14 15 16 17 10 19 14 15 16 10 11 12 13 14 15 16 17 10 19 14 15 16 10 11 12 13 14 15 16 17 10 19 14 15 16 10 11 12 13 14 15 16 17 10 19 14 15 16 10 11 12 13 14 15 16 17 10 19 14 15 16 10 11 12 13 14 15 16 17 10 19 14 15 16 10 11 12 13 14 15 16 17 10 19 14 15 16 10 11 12 13 14 15 16 17 10 19 14 15 16 10 11 12 13 14 15 16 17 10 19 14 15 16 10 11 12 13 14 15 16 17 10 19 14 15 16 10 11 12 13 14 15 16 17 10 19 14 15 16 10 11 12 13 14 15 16 17 10 19 14 15 16 10 11 12 13 14 15 16 17 10 19 14 15 16 10 11 12 13 14 15 16 17 10 19 14 15 16 10 11 12 13 14 15 16 17 10 19 14 15 16 10 11 12 13 14 15 16 10 11 12 13 14 15 16 10 11 12 13 14 15 16 10 11 12 13 14 15 16 10 11 12 13 14 15 16 10 11 12 13 14 15 16 10 11 12 13 14 15 16 10 11 12 13 14 15 16 10 11 12 13 14 15 16 10 11 12 13 14 15 16 10 11 12 13 14 15 16 10 11 12 13 14 15 16 10 11 12 13 14 15 16 10 11 12 13 14 15 16 10 11 12 13 14 15 16 10 11 12 13 14 15 16 10 11 12 13 14 15 16 116 111 12 13 14 15 16 116 11 12 13 14 15 16 116	reproduged, H5Tr.A,B7.
■ Ch341P File でう でう でう でう でう でう でう でう でう でう	Chip Help Ecron 24 xx 2 4401 2 4402 4 2401 2 4403 2 4403 2 4403 2 4403 2 4403 2 4404 2 4403 2 4404 2 4404	E. C.	Settings 00 01 02 03 00 01 02 03 04 05 06 01 02 03 04 05 04 05 04 05 04 05 04 05 04 05 04 05 04 05 04 05 04 05 04 05 04 05 04 05 04 05 04 05 04 05 <t< th=""><th>reproduged, H5Tr.A,B7. </th></t<>	reproduged, H5Tr.A,B7.
2 Ch341P File (2) (2) (2) (2) (2) (2) (2) (2) (2) (2)	Chip Help Ecron 24 xx 2 4401 2 4402 2 4402 2 4403 2 4403 2 4403 2 4403 2 4403 2 4404 2 4402 2 4402	E. C.	Settings 00 01 02 03 04 05 04 05 10	reprplugø/, H5Tr.A,B7. , s5zH5Tr.A,B7. , s5zH5Tr.A,B7. , s5zH5Tr.A,B7. , S57, szH8Tr
2 Ch341P File (2) (2) (2) (2) (2) (2) (2) (2) (2) (2)	Chip Help Ecrom 24 xx 2 4401 2 4402 2 4402 2 4403 2 4403 2 4403 2 4403 2 4403 2 4404 2 4402 2 4402	E. C.	Settings 00 01 02 03 00 01 02 03 04 05 06 01 02 03 04 05 04 05 04 05 04 05 04 05 04 05 04 05 04 05 04 05 04 05 04 05 04 05 04 05 04 05 <t< th=""><th>reprplugø, H5Tr.A,B7. ,sizH5Tr.A,B7. ,sizH5Tr.A,B7. ,sizH5Tr.A,B7. </th></t<>	reprplugø, H5Tr.A,B7. ,sizH5Tr.A,B7. ,sizH5Tr.A,B7. ,sizH5Tr.A,B7.
■ Ch341P/ File マシ マシ マシ マシ マシ マシ マシ マシ マシ マシ	Chip Help Ecron 24 xx 2 4401 2 4402 3 4402 4 2403 4 2403	E. C.	Bettings 00 0.0 02 00 00 00 05 07 00 00 00 00 00 00 00 00 00 00 00 00	- C ×

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 910 Ω resistor, I didn't have a 1K one at my disposal, LOL O. The value is close enough though and the value is not really crucial, 1K, 1.1K, 1.2K, 910 Ω , doesn't really matter as long as the value is close to 1K as possible O. 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 ⁽ⁱⁱⁱ⁾) and the toner will always be a "brand new one" for the MCU/CPU of the printer. Neat, huh ⁽ⁱⁱⁱ⁾.

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).









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 O. 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 O. 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 O.

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 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 <u>0x4e4f@gmail.com</u> or on Reddit (<u>u/PCChipsM922U</u>).

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/XsVVIRIJ#ptpU9R8cK0aSLhPhxK9nVkF7ReagDK6c1gqZ42hVZV8