

## Inbetriebnahme des Willem Programmers PCB5-E.

Die Software GQUSBprg 0.98d6 Willem Enhanced / Dual Power Programmer – PCB5.0E

von der Website <http://www.zeitech.de/index.php?s=36&> enthält die Datei

11.02.2011 16:17 3.481.221 Enhanced\_Willem\_Programmer\_CD\_PCB5.0E.zip

welche folgenden Inhalt hat:

26.06.2007 13:12 792.779 Enhanced Willem manual.pdf

25.09.2008 12:07 1.918.367 NEW Dual Powered Willem PCB5.0 User Guide.pdf

25.09.2005 17:20 971.850 Willem-setup\_98d6.exe

Die Manuals haben bewährte asiatische Qualitäten, was mich aber nicht vom Projekt zurückschrecken liess. Und das exe meldete sich mit Willem Eprom PCB35(0.98D6) was angesichts der Board-Beschriftung PCP5-E Zweifel aufkommen liess.

Erstmal stellte ich im BIOS meines PC für den Parallel Port den ECP Mode ein.

Dann fuhr ich den Rechner wieder runter und steckte die Kabel wie beschrieben an.

Beim ersten Start der Software gabs im wesentlichen nur Fehlermeldungen,

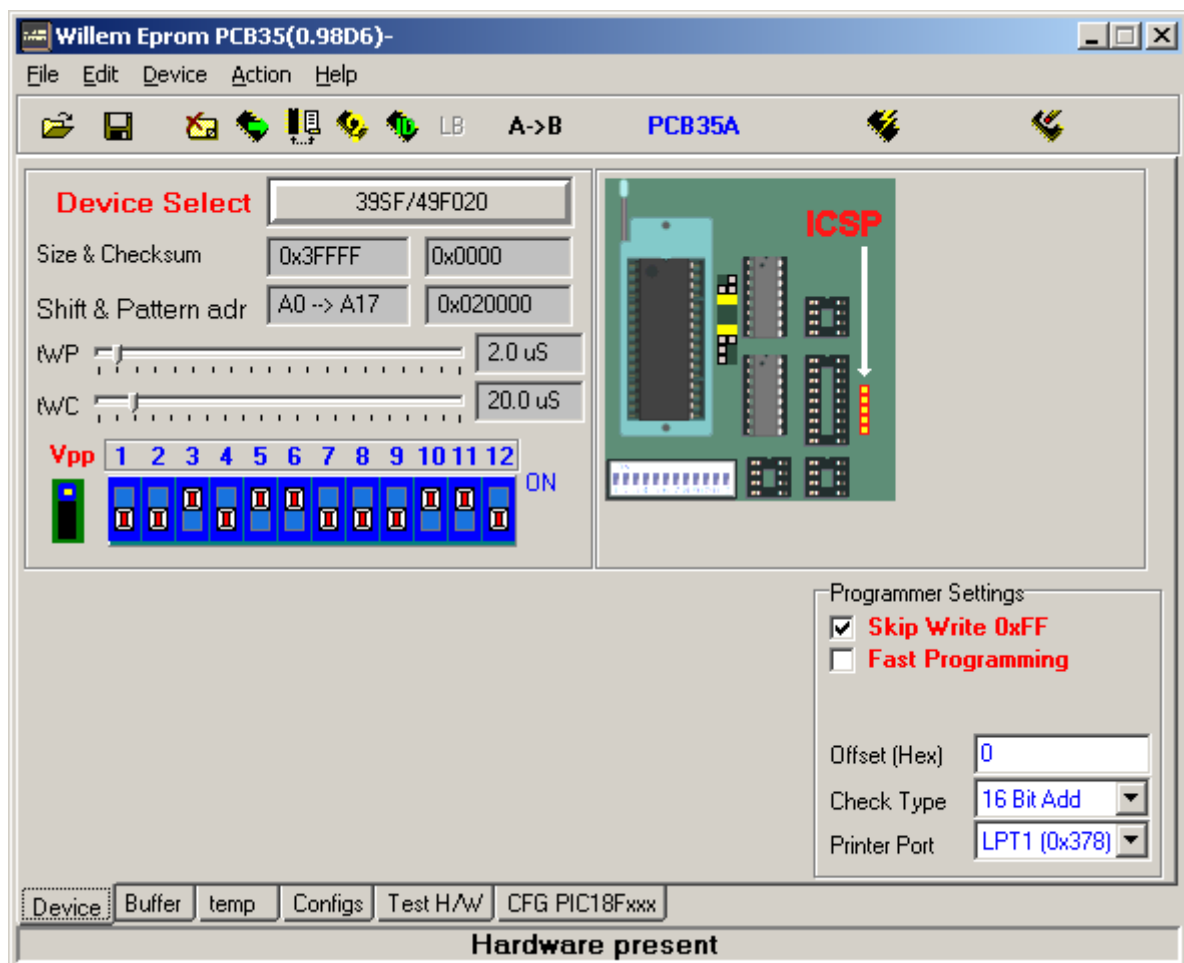
aber die grüne Power Led leuchtete und auch Program und Vpp.

Das Ausschalten der Software gelang nur mit dem Windows Task Manager; wenn ich das Programm von sich aus ausschalten wollte, gabs die Fehlermeldung Privileged Instruction und die Software machte lustig weiter.

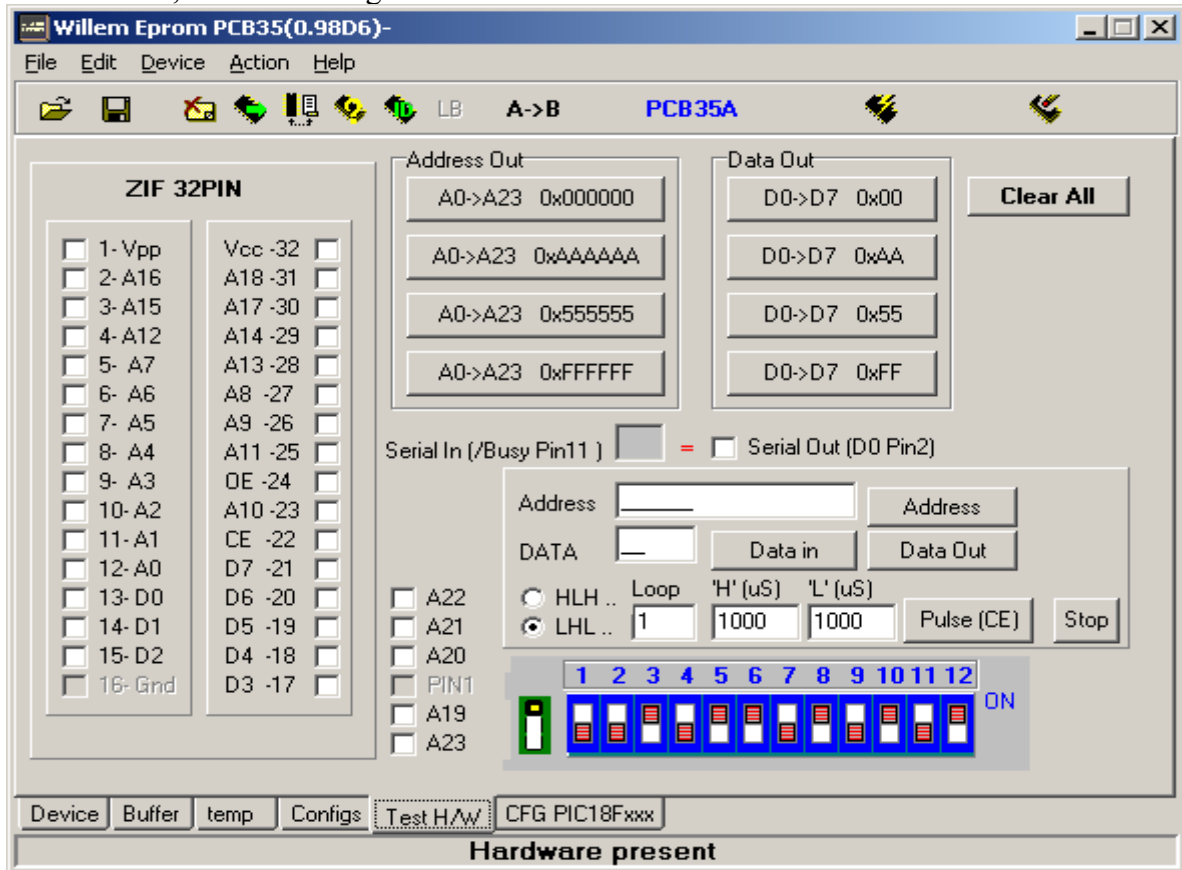
Beim zweiten Start sah es schon besser aus. Es leuchtete nur die grüne Power LED und es gab keine Fehlermeldungen.

Dann wählte ich im Menu oben Help und Test Hardware aus. Nun sagte er Hardware present. Soweit so gut.

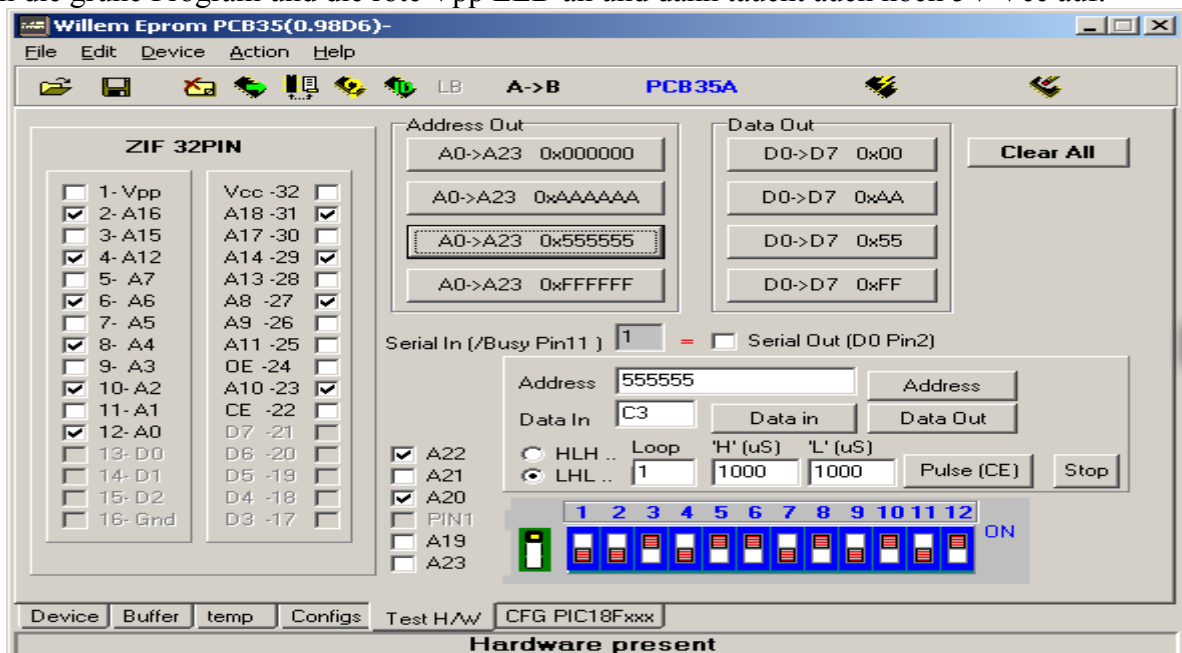
Da ich einen SST 39SF020 Bios Flash-Speicher bearbeiten wollte, wählte ich nun bei Device Select unter Flash 28/39 den passenden Typ aus und stellte das Mäuseklavier SW12 Pin 1-12 entsprechend dem nun angezeigten Bild ein:



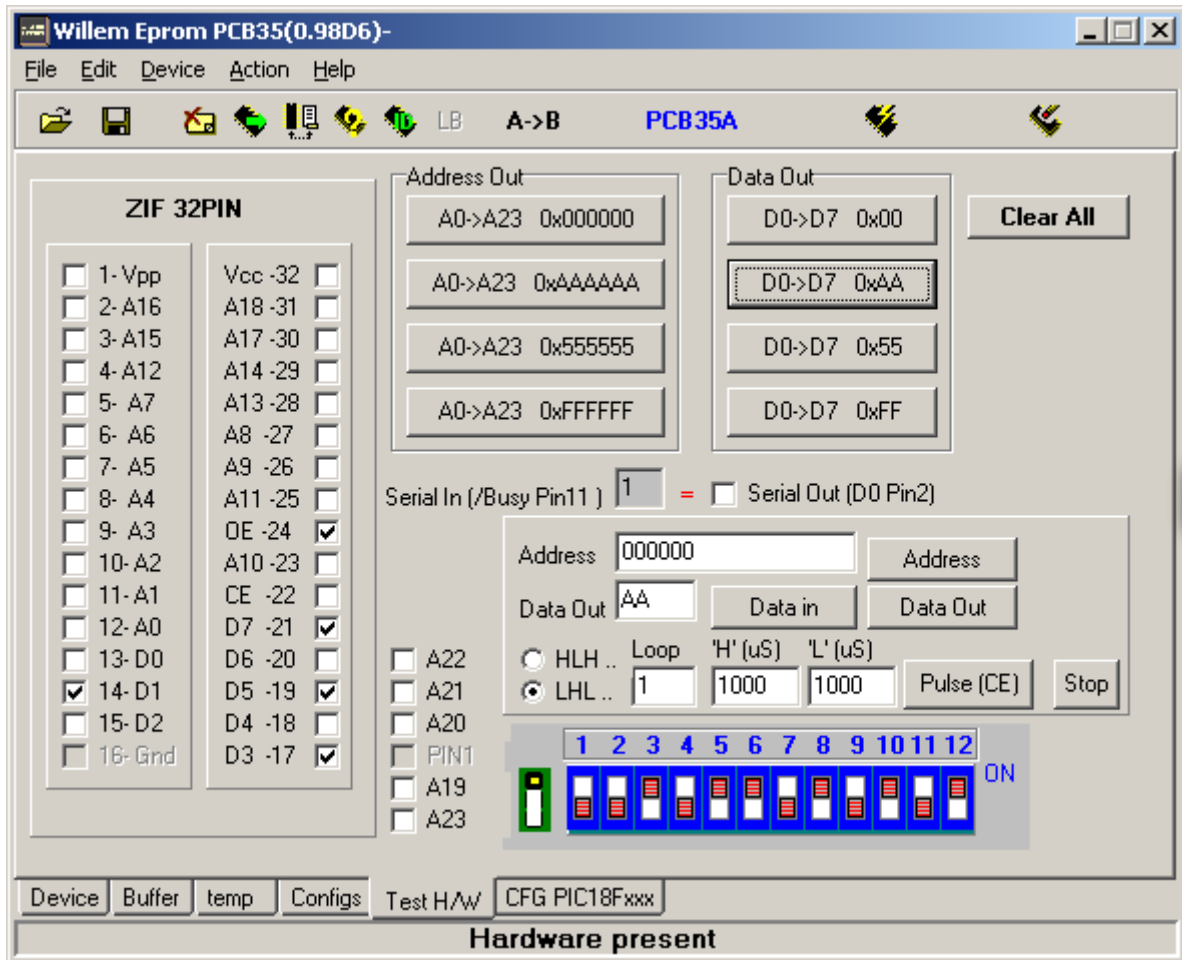
Mit gesundem Misstrauen bewaffnet wollte ich nun erstmal die Spannungen checken, bevor ich das Chip brate. (Obwohl wir mit dem Braten schon einige zehntausend Jahre Erfahrung haben). Da bietet sich der auf dem Bild mit einem roten Pfeil bezeichnete ICSP-Pfostensteckverbinder an. Auf dem Board steht rechts daneben von oben nach unten Vpp, Vcc, Ground, Dat, Clock. Vpp liefert erstmal 0V und Vcc auch. Wenn ich nochmal Help und Test Hardware anwähle, brennt kurz die grüne Programm LED. Wenn ich nun unten auf dem Bildschirm Test H/W drücke, erhalte ich folgendes Bild:



Jetzt geht's zum erweiterten Hardware-Test. Jetzt muss ich schon wieder das Mäuseklavier umstellen. Wenn ich nun J3 (Bild: grüner Rahmen) oben jumpere wie auf dem Bild und Address Out 0x555555 drücke, erhalte ich sogar auf pin 1 des ZIF 32 Pin 4,8V Vpp(x1). Manchmal gehen auch die grüne Program und die rote Vpp LED an und dann taucht auch noch 5V Vcc auf.



Die Häkchen auf dem ZIF 32Pin Sockel sagen +4,8V, die man mit dem Voltmeter nachmessen kann.  
 Hier habe ich auch noch Data out 0xAA gedrückt und kann die Pegel auf den Datenpins nachmessen.



Eigentlich sollte Pin 1 (VPP) 5V zeigen, wenn man auf dem Bild auf Pin 1 drückt, aber hier geht es nur auf 5V wenn man Address Out auf 0x555555 oder 0xFFFFF drückt. (x1). Das ist aber merkwürdig.

Bei Pin 32 (Vcc) geht es richtig, wenn man auf Pin 32 im Bild klickt. Aber Vpp (Pin 1) geht auch auf 4,8V, wenn man Clear All drückt. ???!

Nächster Tag.

Nun das Kabel mal fest an beiden Seiten angeschraubt.

Es hat sich nichts geändert bis darauf, dass Vpp (Pin1) nicht mehr auf H geht wenn man Clear All drückt.

Nun mal Pulse CE (Pin 22) probiert. H und L auf 99000 µs eingestellt und Loop auf 999.

Mit dem Oszi maß ich nun eine Periodendauer von 166 ms. Wenn man Loop auf 1 lässt, kommt wirklich bloss ein Impuls.

Auf Pin 22 (CE) des PLCC32-Sockels kommt er auch an.

WE (PLCC32 Pin 31) geht auf 5V wenn man Address Out auf 0x555555 oder 0xFFFFF drückt, aber nicht wenn man Vpp oder Vcc drückt.

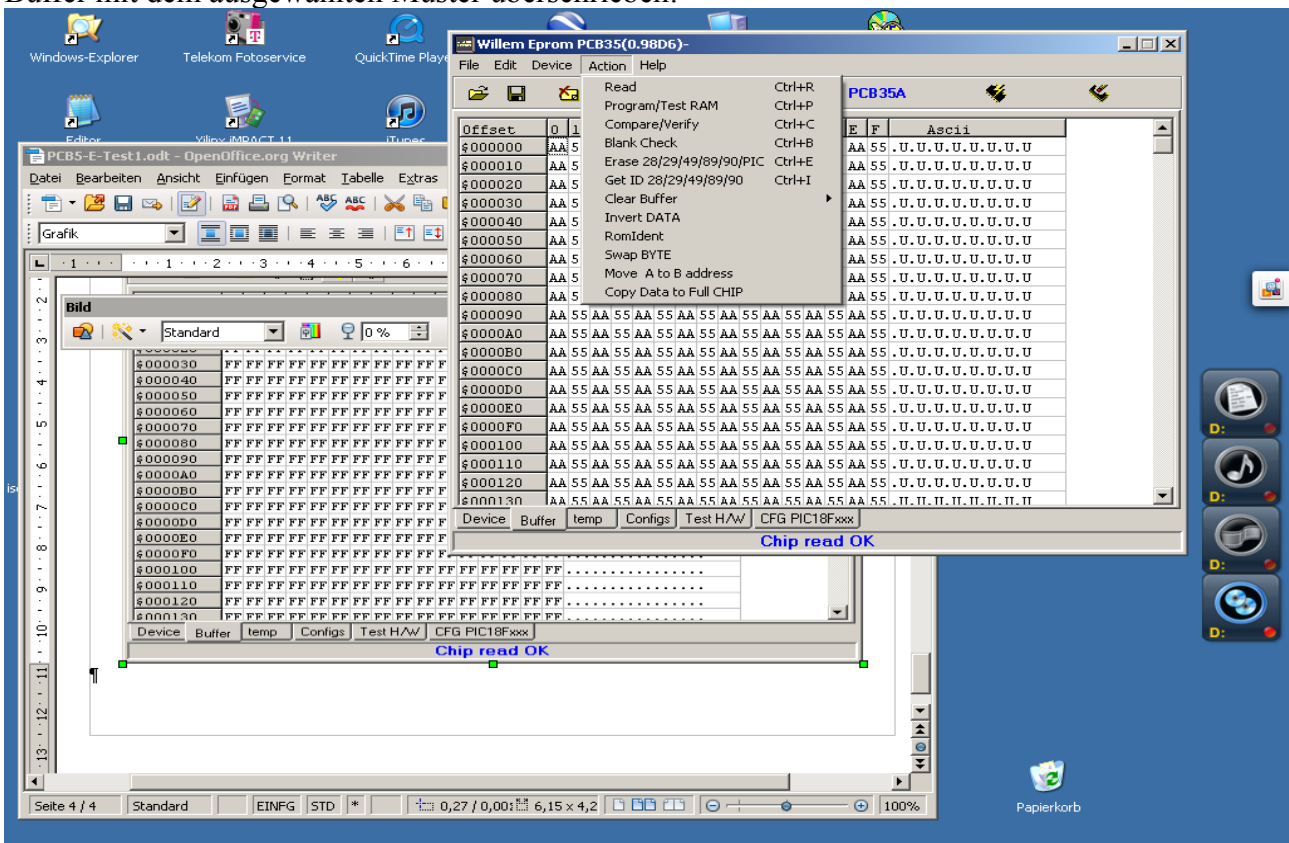
Wenn man OE drückt, tauchen im ZIF32-Bild die Pins 13-15 und 17-21 auf und lassen sich auch setzen.

Nun habe ich mal einen 39SF020A in den PLCC32-Sockel gesteckt und die Read Data from Chip to Data buffer-Taste gedrückt. Dann passiert folgendes:

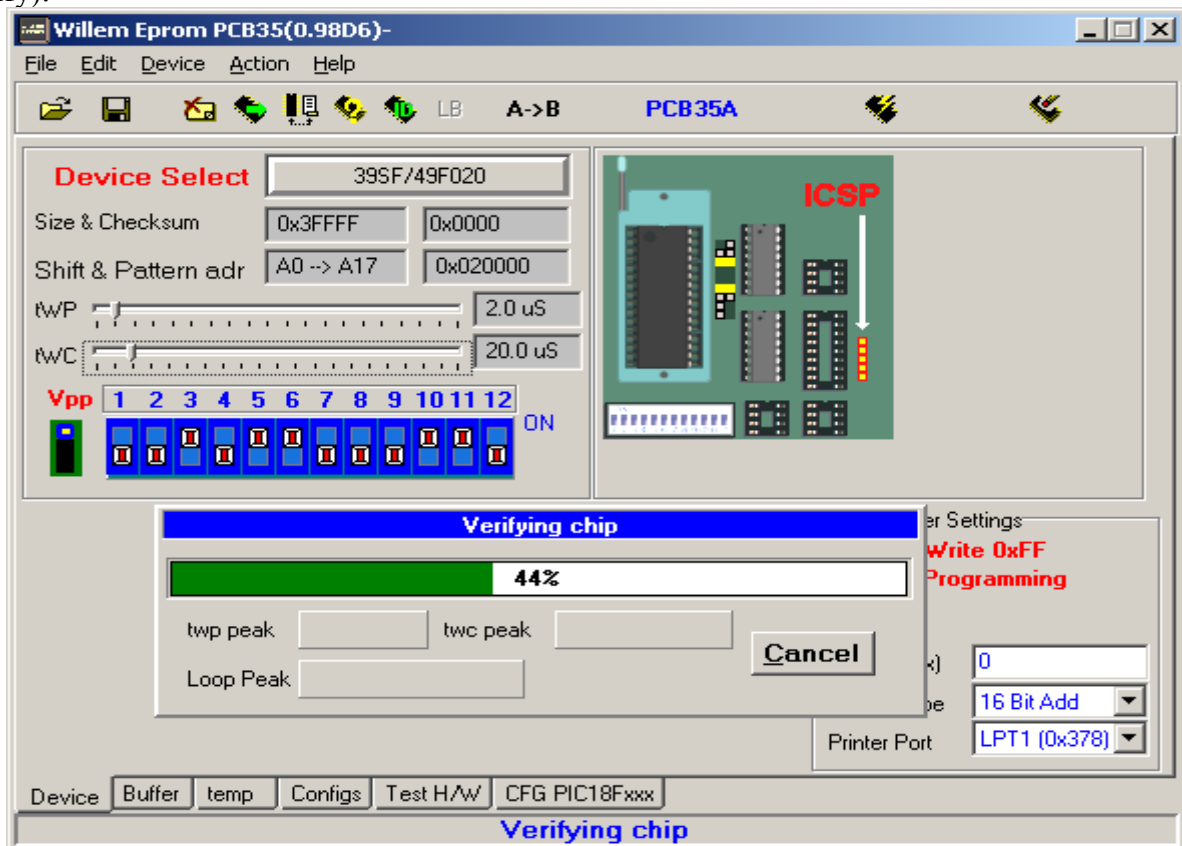


Na klar, das war ein unprogrammierter Flash.

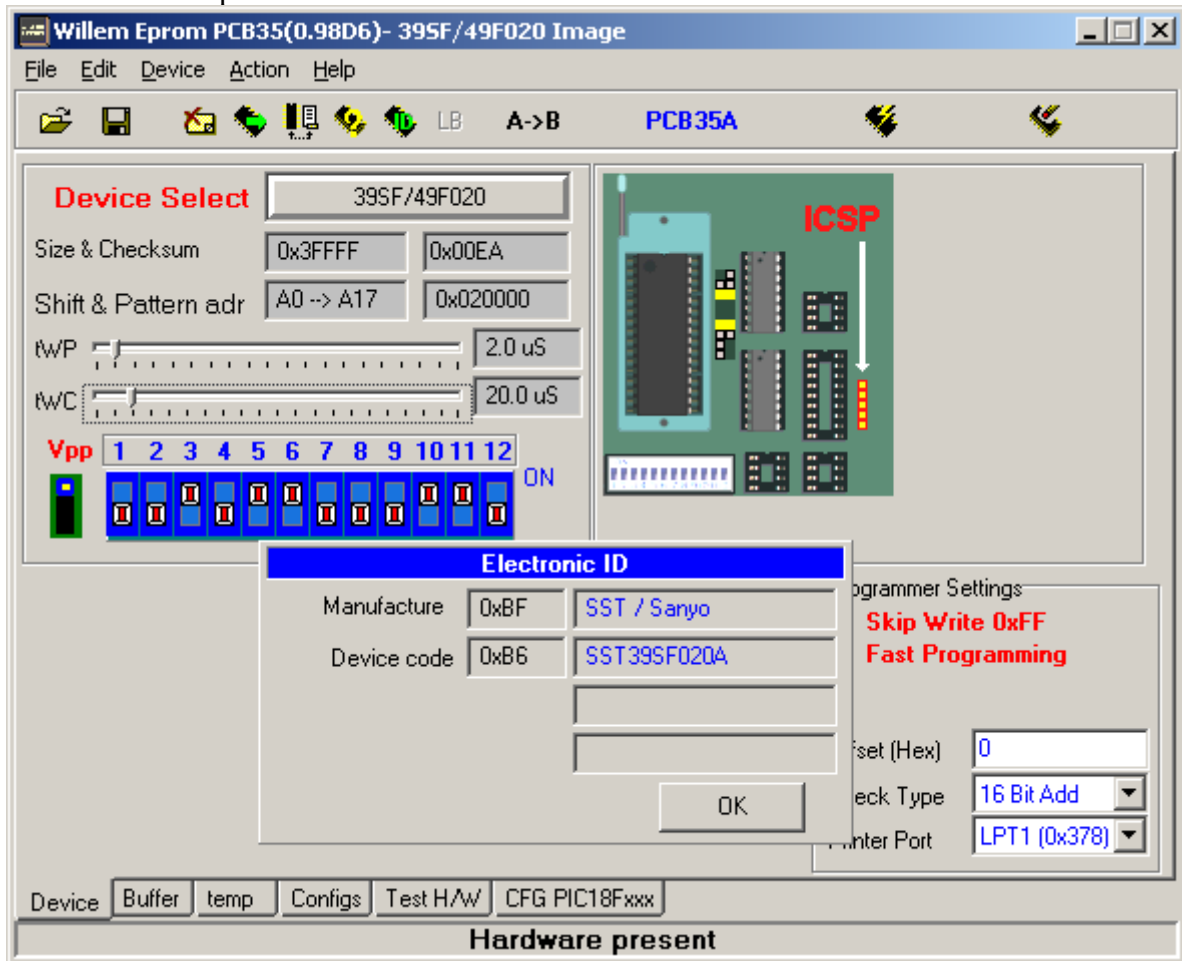
Wenn ich oben die Action-Taste drücke und Clear Buffer und Auswahl AA55 drücke, wird der Buffer mit dem ausgewählten Muster überschrieben:



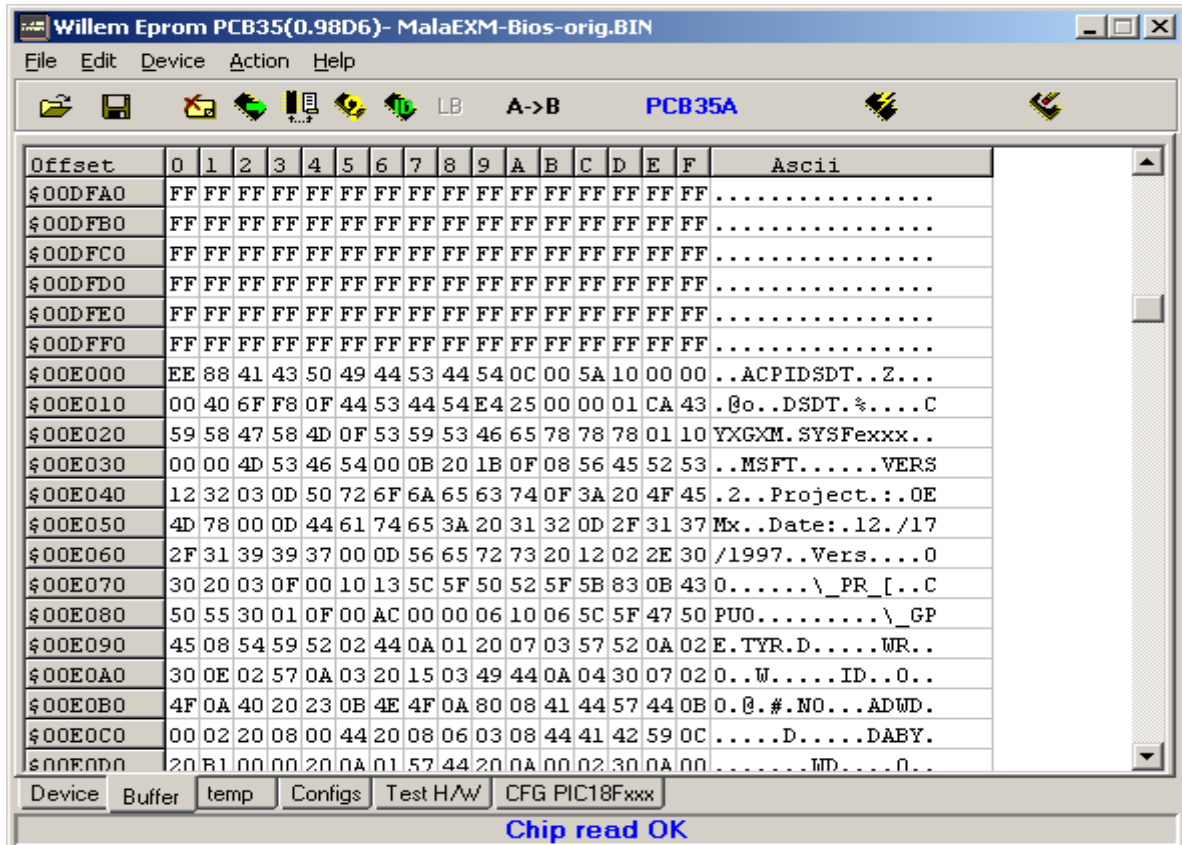
Und nun kann ich den Buffer ins Chip schreiben mit dem Symbol Chip+Blitz (rechts oben) oder Action + Program)Test RAM. Das sieht dann so aus (hier hat er gerade programmiert und macht verify):



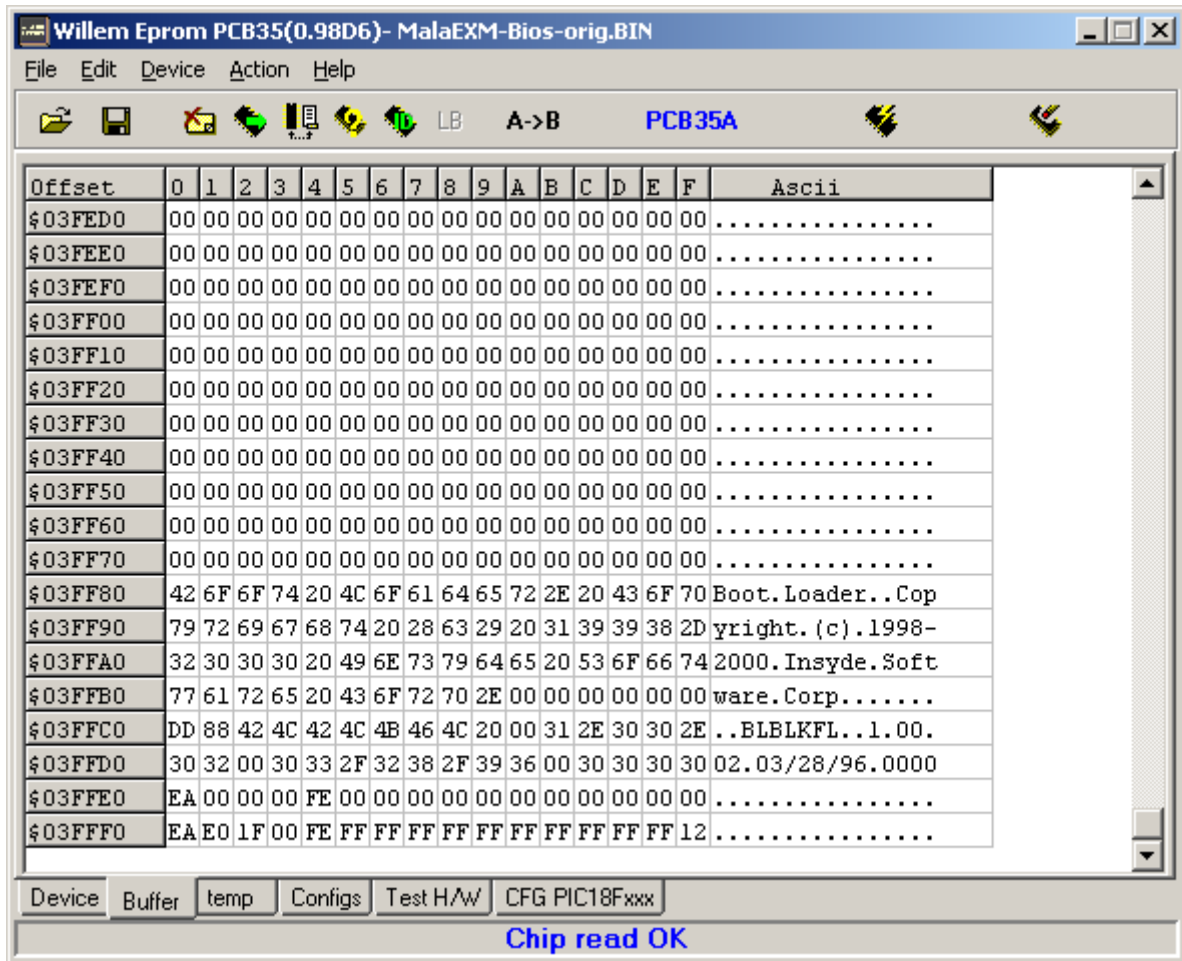
Und siehe da, beim Rücklesen steht es tatsächlich noch drin.  
 Und so sieht das Chip ID aus:



Und so sieht der Inhalt des BIOS aus:



Und hier ist das Ende des BIOS:



Wunderbar. Alles was ich brauchte hat funktioniert.