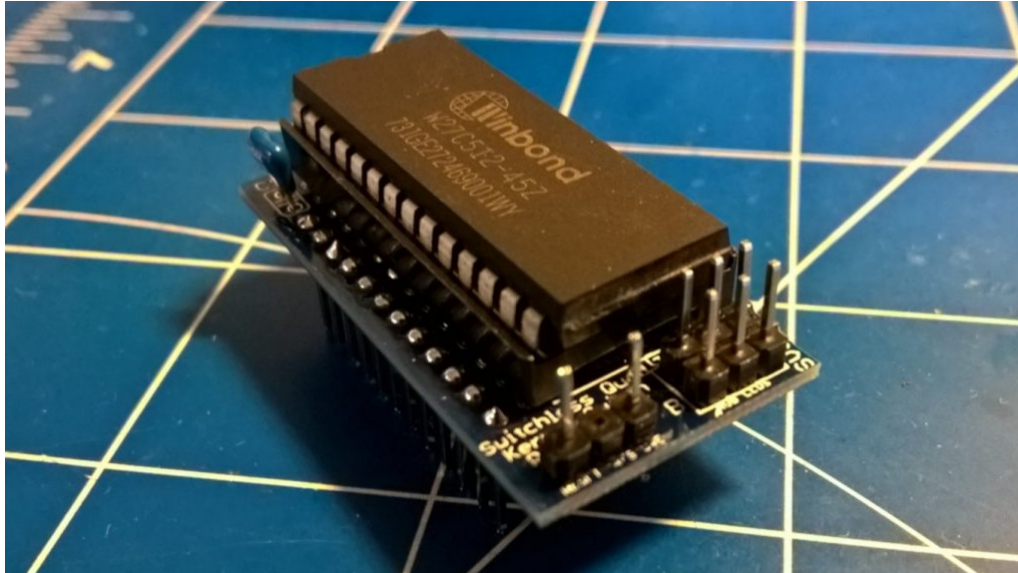


VIDEOSPIEL GESCHICHTEN

Persönliche Geschichten über Videospiele

<https://www.videospielgeschichten.de>



Anleitung: Einbau eines Ersatz-ROM in einen C64 oder VIC20

Thilo Niewöhner am Samstag, dem 6. Oktober 2018

Nach einigen Experimenten mit ROMs für den C64 schreibe ich hier eine kurze Anleitung, wie man die alten 2364 ROMs, die sich in vielen 8-bit-Rechnern finden, durch neue ersetzt.

Das Hauptproblem mit diesen ROMs ist, dass es sie praktisch nicht mehr als Ersatzteile gibt. Wenn sie ausfallen – die Frage nach dem „Falls“ stellt sich fast nicht mehr –, ist es sehr schwer, wenn nicht unmöglich, sie mit Originalen zu ersetzen. Es gibt bereits viele Anleitungen, wie man (E)EPROM für den Einsatz auf einem C64-Board anpasst. Ein (sehr gutes) Beispiel findet sich bei [Tynemouth Software](#).

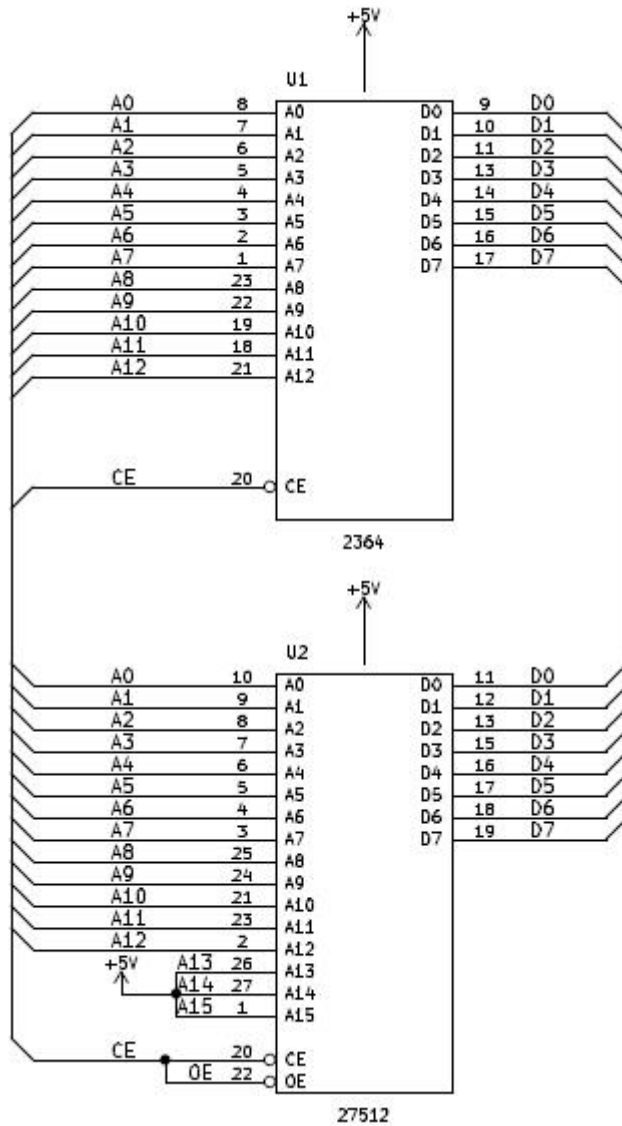
Schritt 1 – Guter Draht zur Hardware

Das erste Problem, das sich stellt: Die Anzahl der Pins und die Belegung des 2364 und des 27C128 unterscheiden sich gravierend. Der einfache direkte Austausch klappt also schon mal nicht.

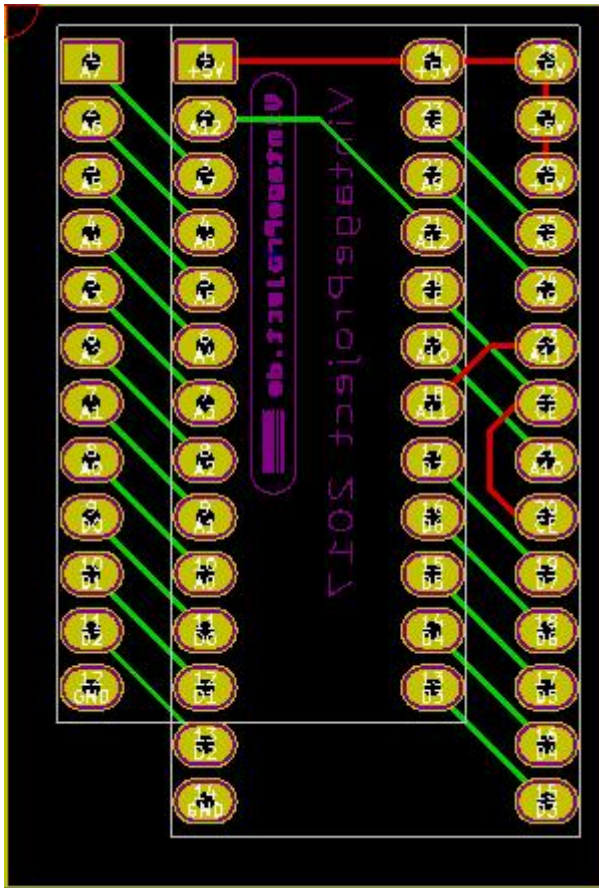
Die Hardwareanpassung ist also einer der notwendigen Schritte, für den es aber zum

Glück viele verschiedene Lösungen gibt.

Eine, die ich für mich selber erstellt habe, findet sich auf [meinem Github Repository](#). Dieses kleine Board ist die Grundlage für den Einbau eines EPROM aus der 27C-Serie (27C64, 27C128 usw.) auf den vorhandenen Sockel im C64, etwa als BASIC oder KERNAL ROM. Wer das möchte, kann übrigens auch EEPROM verwenden, also elektrisch löschbare ROMs. Diese stammen dann aus der 28C-Serie, sind ansonsten aber genauso einsetzbar.



KiCad-Schaltplan für den „2364 to 27512 adapter“, abgeleitet von Tynemouth Softwares Entwurf. (Bild: Thilo Niewöhner)



Entwurf der Platine in KiCad: „2364 to 27512 adapter“. (Bild: Thilo Niewöhner)

Wichtig hier: Die Adressleitungen A13-A15 werden fest auf +5V verbunden. Dies hat damit zu tun, dass bei EPROMs auf diesen Pins zum Teil Steuerleitungen liegen, die LOW-aktiv sind. Erst ab einer Größe von 128 kbit sind dort zusätzliche Adressleitungen zu finden.

Diese Pins auf +5V zu legen, verhindert also Fehlfunktionen. Allerdings ist damit auch nur ein 64-Kilobit-Block (8 kByte) adressierbar. Sollen mehrere ROMs über Schalter oder externe Logik ausgewählt werden, ist hier mehr Aufwand in der Beschaltung erforderlich. In diesem Beitrag geht es aber erstmal nur um einen 1:1-Ersatz bestehender EPROMs.

Nachdem nun das EPROM erfolgreich mit dem alten Sockel verbunden werden kann, ist der nächste Schritt, das EPROM mit den richtigen Daten zu füllen, also das EPROM zu „brennen“.

Hierbei gibt es eine Stolperfalle: Wird ein EPROM mit mehr als 64 kbit (wie die originalen 2364) verwendet, müssen die 64-kbit-ROMs des C64 im Adreßraum des größeren EPROM richtig platziert werden. Wie aus dem Schaltplan zu sehen ist, sind die Adreßleitungen A13-A15 auf HIGH gelegt. Dies dient zum einen der Kompatibilität mit kleineren ROMs, bei denen hier keine Adreßleitungen liegen, sondern LOW-aktive Steuerleitungen. Wären diese Leitungen LOW, würde das EPROM nicht wie erwartet funktionieren.

Zum anderen legt es den adressierbaren Bereich auf den untersten Speicherblock (16-bit-Adresse: 000x xxxx xxxx xxxxb) fest.

Um sicherzugehen, dass der Computer auf jeden Fall ein gültiges ROM-Abbild vorfindet

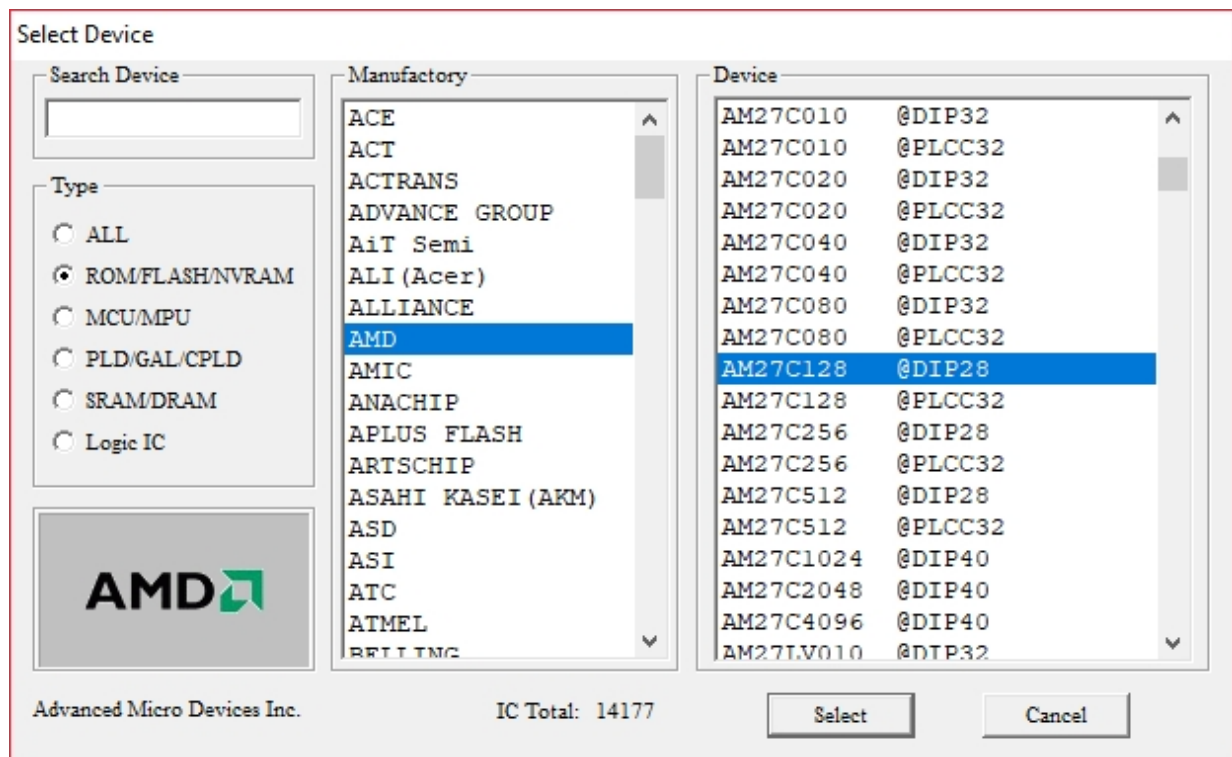
und in seinen Speicher einbinden kann, auch wenn man wegen der Adreßleitungen unsicher ist, ist der einfachste Ansatz, das ROM in mehreren Kopien im EPROM abzulegen. Für ROM-Module, die umschaltbare ROMs bekommen sollen, müssen diese dann aber wiederum anders angelegt werden.

Schritt 2 – Die richtigen Bits an die richtige Stelle

Um nun das EPROM mit Daten zu füllen, brauchen wir ein EPROM-Programmiergerät bzw. einen EPROM-Brenner.

Das Standardgerät unter Elektronikhobbyisten ist mittlerweile der TL866, hergestellt von [AutoElectric](#) in China. In der Maker-Community ist dieses Gerät weitverbreitet und stellt momentan die günstigste Lösung dar, selbst EPROM und EEPROM zu programmieren.

In der Software ist nun als erstes das richtige EPROM auszuwählen. In meinem Beispiel ein AM27C128, also ein EPROM mit 128 kbit bzw. 16 kByte.

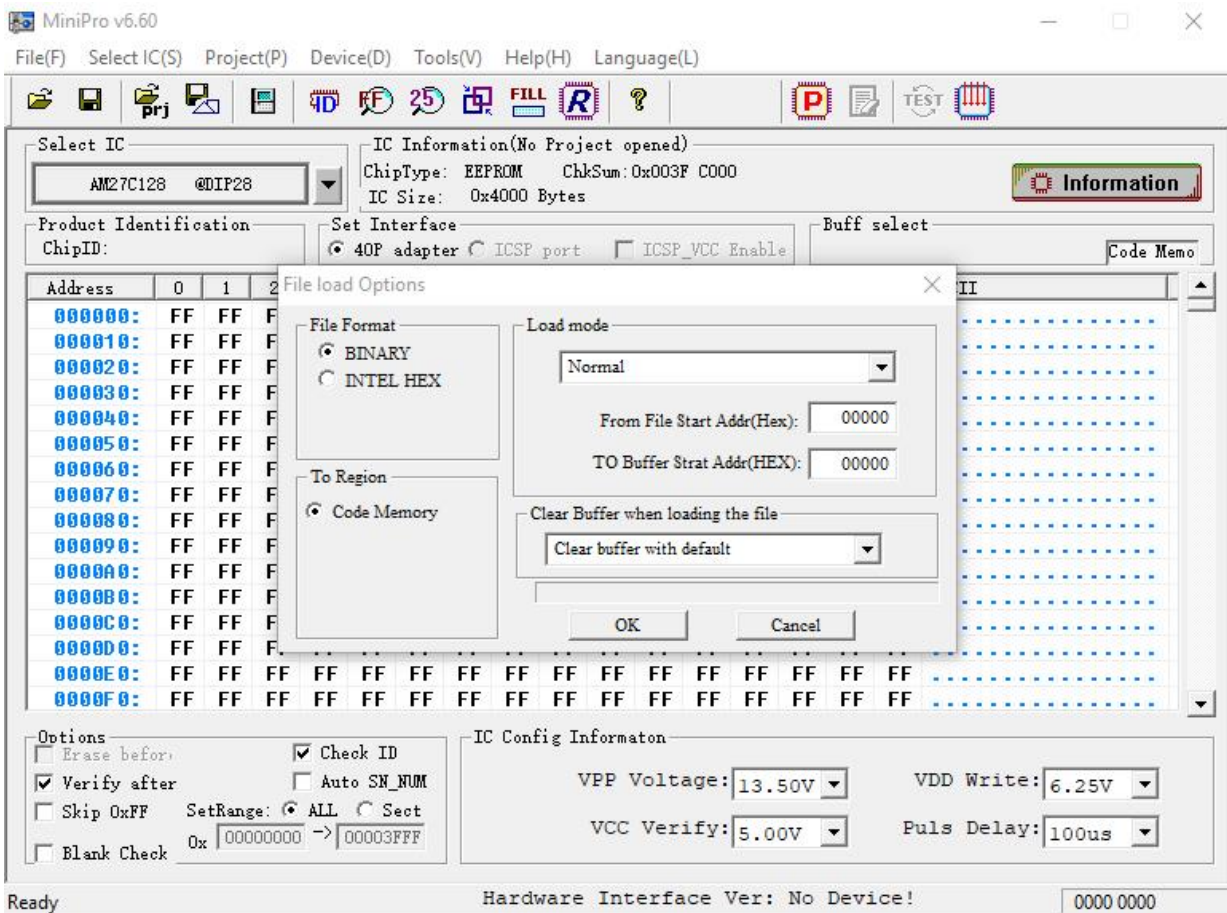


Auswahl EPROM AMD AM27C128. (Bild: Thilo Niewöhner)

Danach zeigt das Fenster, dass der Puffer zunächst mit „FF“ gefüllt wird. Das repräsentiert leeren Speicher, also den Urzustand des EPROM, gibt aber keine Information über den tatsächlichen Zustand des EPROMs. Dieses muss natürlich ebenfalls leer sein, was über die „Blank Check“-Funktion des TL866 überprüft werden kann.

Jetzt laden wir die .bin-Datei für das ROM, das wir brennen wollen. Hier ist es das Original Kernal-ROM.

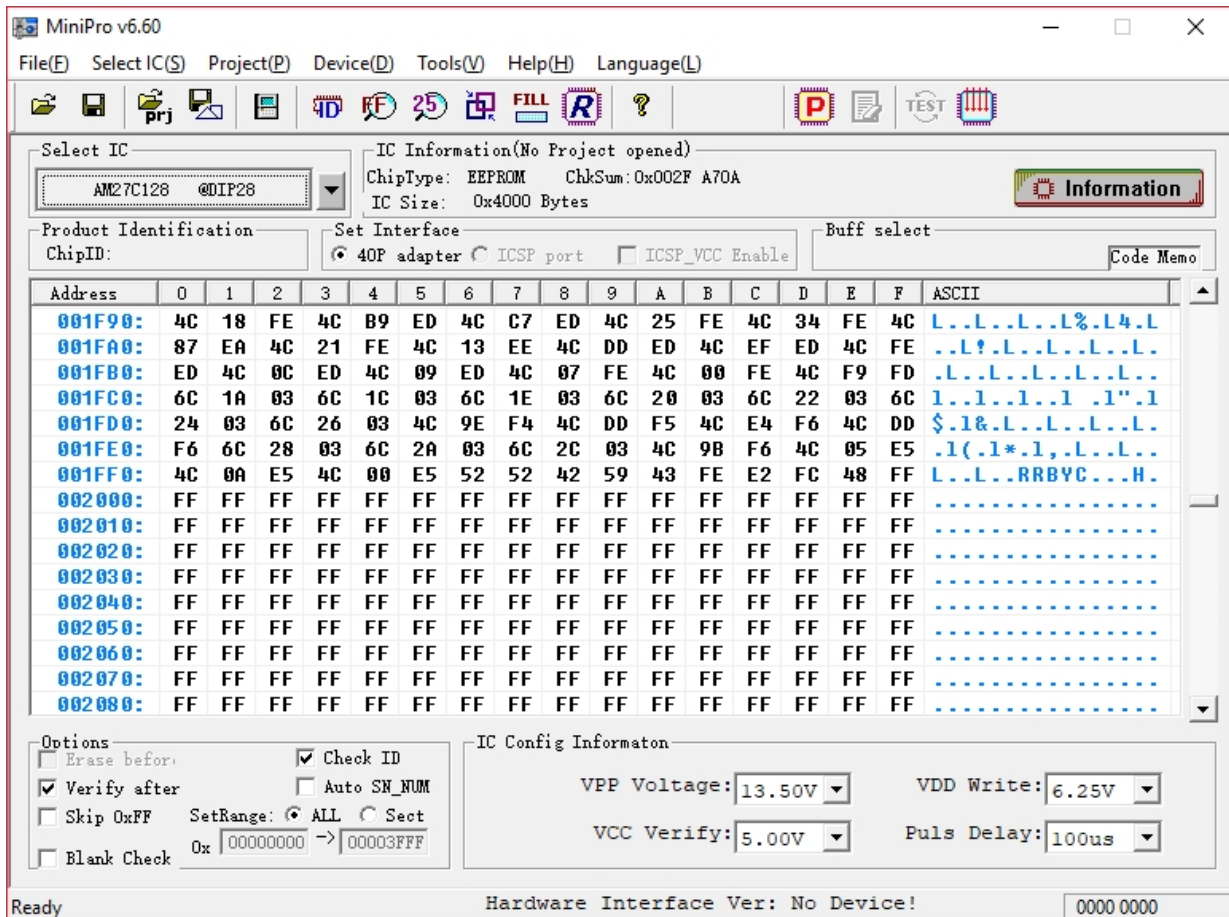
Nach der Auswahl der bin-Datei und einem Klick auf „Öffnen“ erscheint folgendes Bild:



MiniPro ROM laden. (Bild: Thilo Niewöhner)

Lasst alle Parameter unverändert und klickt „OK“. Damit wird die .bin-Datei ab der Adresse 00000h in den Puffer geladen und der Rest des Puffers mit FFh gefüllt. (FFh entspricht leeren Speicherplätzen)

Da die normale ROM-Größe des C64 8 kByte ist, könnt Ihr nachschauen und feststellen, dass ab 02000h der Puffer nur noch „FF“ zeigt.



MiniPro ROM geladen ab Adress \$2000. (Bild: Thilo Niewöhner)

Das ist sehr gut, da wir ab dieser Adresse dasselbe ROM noch einmal einfügen wollen. Das machen wir um sicherzugehen, dass wir mit der voreingestellten Adresse (siehe „A13-A15 auf 5V“) auf jeden Fall ein gültiges ROM-Abbild erwischen. Im späteren Ausbau lassen sich verschiedene ROMs anlegen, die dann von außen umschaltbar sein können.

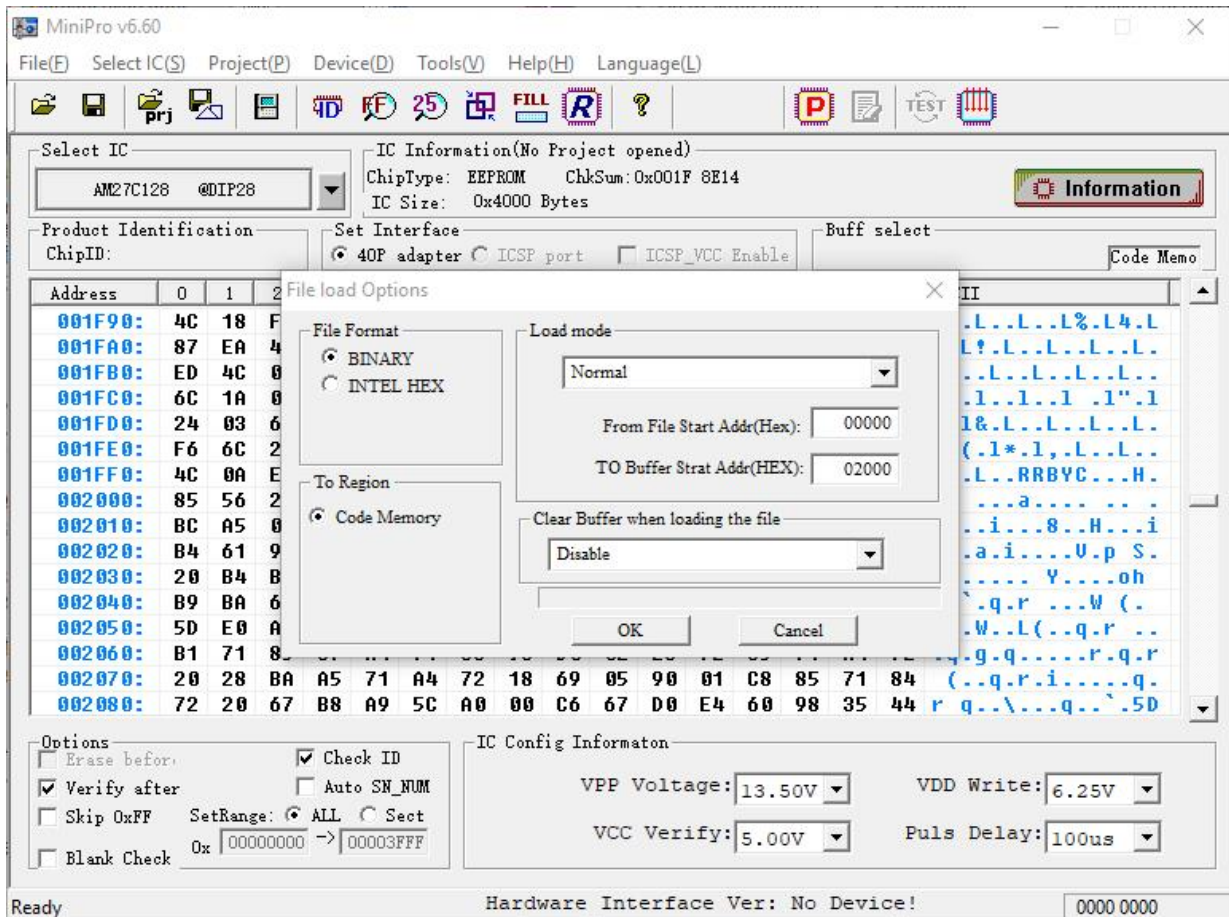
Sollten hier aber Daten vorhanden sein, prüft bitte, ob Ihr nicht die falsche Datei erwischt habt. Die richtigen .bin-Dateien haben genau 8 kByte.

Dazu wiederholen wir den oben beschriebenen Prozeß, verändern aber beim zweiten Dialog die Parameter.

Lasst den Parameter „From File Start Address“ auf 0000h. Das bedeutet, dass die .bin-Datei von Anfang an geladen und nichts ausgelassen wird, ganz wie vorhin.

Aber den anderen Parameter „To Buffer Start Address“ müssen wir auf 2000h einstellen und „Clear buffer when loading the file“ auf „Disabled“.

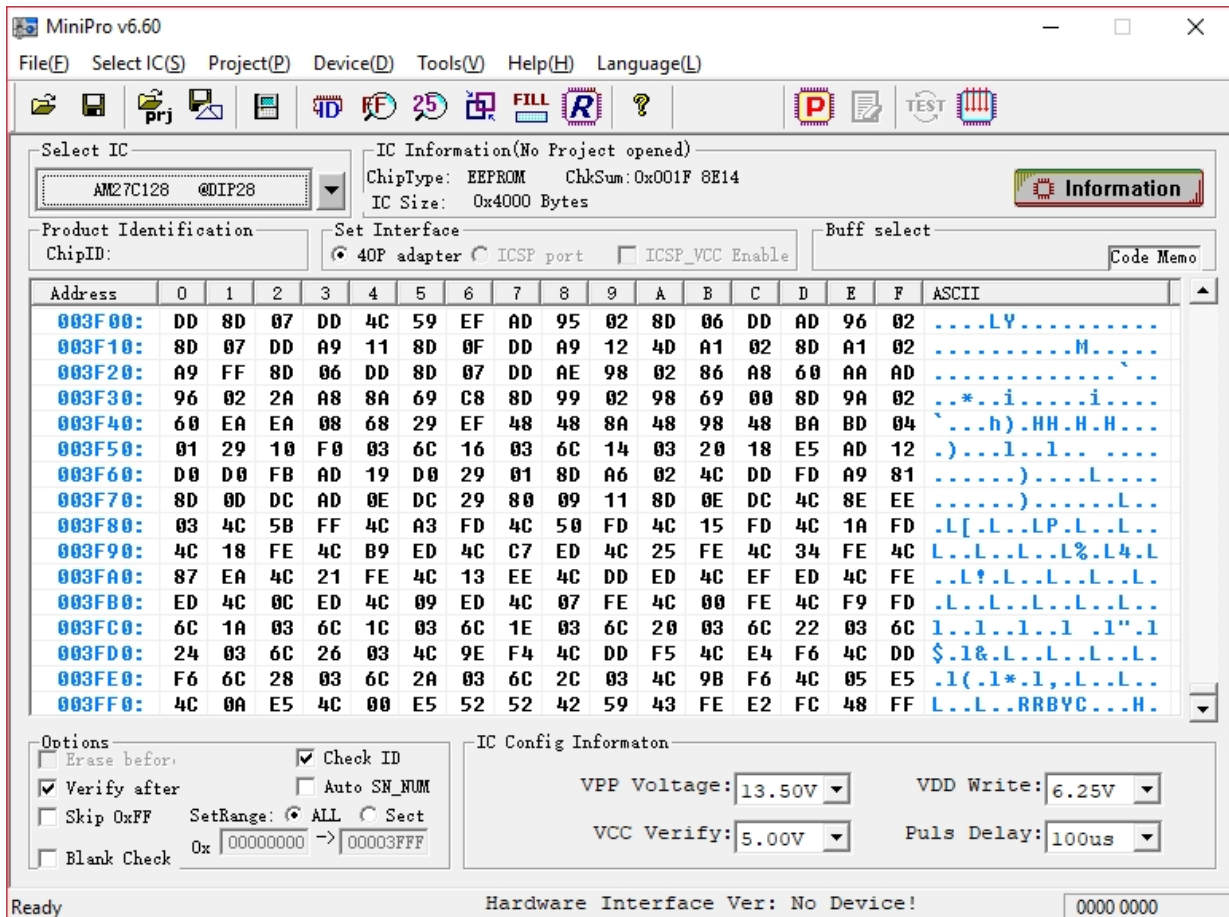
Damit wird die zweite Kopie überschneidungsfrei nach der Ersten in den Puffer eingelesen, ohne den Rest des Puffers wieder zu leeren (und damit unsere erste Kopie zu löschen).



MiniPro ROM Kopie laden ab 2000h. (Bild: Thilo Niewöhner)

Schauen wir in den Puffer, sehen wir, dass jetzt ab 2000h statt „FF“ ebenfalls Daten vorhanden sind. Ihr werdet auch sehen, dass sich hier die Daten von ganz oben wiederholen.

Ihr solltet auch überprüfen, dass der Puffer bis zum Ende beschrieben wurde.



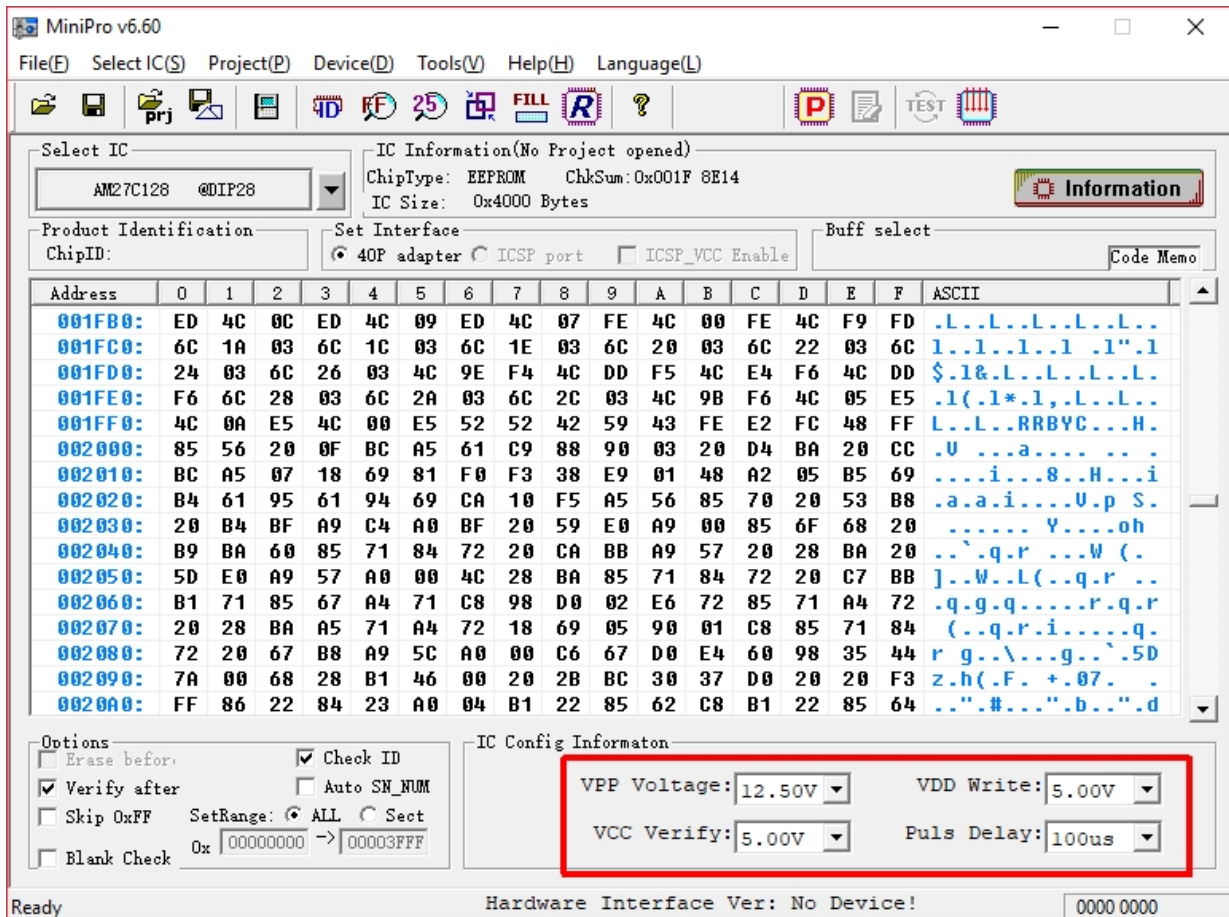
MiniPro ROM geladen Puffer gefüllt bis 3FF0h. (Bild: Thilo Niewöhner)

Schaut Euch die Zeilen 001FF0h und 003FF0h (und gerne auch weitere solche Pärchen) an. Diese Zeilen sind die jeweils letzten Bytes der ROM-Datei und müssen identisch sein.

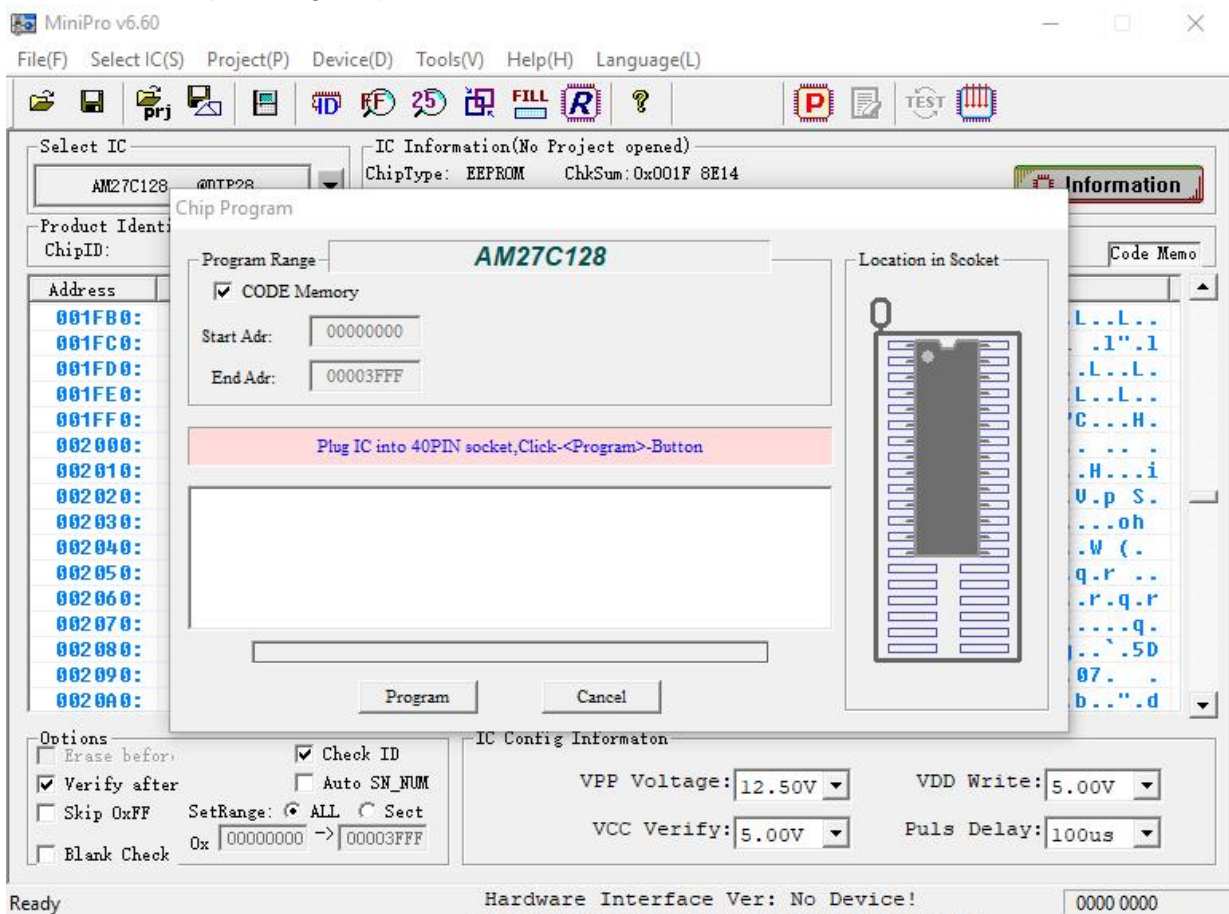
Sind sie das, brennen wir jetzt das EPROM.

Anmerkung: Nicht immer sind die Parameter für das Programmieren korrekt.

In der Regel ist die Programmierspannung bei 12,5 V, die Lesespannung 5 V. Prüft das Datenblatt Eurer (E)EPROMs und korrigiert die Werte. Ansonsten kann es zu Fehlern kommen.



MiniPro ROM Spannungen. (Bild: Thilo Niewöhner)



MiniPro ROM programmieren. (Bild: Thilo Niewöhner)

MiniPro Programmer wird in der Standardconfiguration eine abschließende Überprüfung

des EPROM-Inhaltes vornehmen und Euch eine Rückmeldung geben.

Schritt 3 – Test am lebenden Objekt

Hat alles geklappt, könnt Ihr das EPROM jetzt im C64 ausprobieren.

Achtung bei der Einbaulage!

Achtet auf die kleine Kerbe nahe Pin 1, damit Ihr es nicht verkehrtherum einbaut.

Hat etwas nicht geklappt, fangt von vorne an. Bei EPROMs (das sind die mit dem kleinen Fenster) ist es wahrscheinlich notwendig sie zu löschen. Dafür braucht Ihr aber ein UV-Löschgerät. EEPROMs dagegen werden vom TL866 automatisch gelöscht.

Viel Erfolg bei Euren Reparaturen und haltet die C64, VIC20 und wie sie alle heißen in Betrieb.

Es wäre schade drum!

Schaut auch auf Twitter vorbei und lest die Twitter-Threads und die verlinkten Blogbeiträge der Retro Community.

Es wird immer wieder Lesenswertes und Interessantes gepostet.

Ach ja: Falls Ihr noch einen Fehler findet, lasst es mich wissen.

Ergänzungen/Anmerkungen

EPROM/EEPROM/OTP-ROM

Ich schreibe hier zur Übersichtlichkeit immer von EPROM. Allerdings habe ich mehrere dieser Umbauten auch mit EEPROMs durchgeführt.

„EPROM“ schließt also sowohl die UV-löschbaren EPROMs als auch die elektrisch löschbaren EEPROMs ein. Auch sog. OTP-ROM („One-Time Programmable“) sind hier prinzipiell einsetzbar, wenn auch in vielen Fällen unwirtschaftlich, da sie nur ein einziges Mal beschrieben werden können. Für Dinge wie den Ersatz des PLA kann es aber notwendig sein, auf schnellere OTP-ROM mit <45ns zurückzugreifen.

TL866

Die Varianten TL866A und TL866CS unterscheiden sich in einer zusätzlichen 6-poligen Schnittstelle und geänderter Software, die das einfachere Programmieren von Mikrocontrollern erlaubt. Dabei hat der TL866A diese Schnittstelle bereits eingebaut, beim CS ist ein späterer Umbau möglich, erfordert aber einige Arbeit.

Wer noch kein Programmiergerät hat, sollte die A-Version bestellen, da die Preisunterschiede meist marginal, die zusätzlichen Möglichkeiten aber wenigstens interessant sind.

Zum aktuellen TL866 II kann ich noch nichts sagen. Wenn Ihr da mehr wisst, sagt Bescheid! ?

No Device!

Ja, das ist korrekt. Als ich die Screenshots erstellt habe, hatte ich den TL866 nicht zur Hand. ?

Für diesen Zweck ist das nicht kritisch. Wenn Ihr aber selbst ROMs brennen wollt, achtet darauf, dass Euer Brenner korrekt erkannt wird.

Weiterführende Links

- [GitHub profile von VintageProject \(Thilo Niewöhner\)](#)
- [Commodore VIC20 Repair bei Tynemouth Software](#)

Dieser Beitrag wurde publiziert am Samstag, dem 6. Oktober 2018 um 10:00 Uhr in der Kategorie: [Hardware](#). Kommentare können über den [Kommentar \(RSS\)](#) Feed verfolgt werden. Du kannst zum Ende springen und ein Kommentar abgeben. Pingen ist momentan nicht erlaubt.



Über Videospiegelgeschichten

Videospiegelgeschichten ist eine offene Plattform für Hobbyautoren und Journalisten. Die Webseite wurde 2009 gegründet, um es jedem Menschen, unabhängig von seiner Profession, zu ermöglichen, persönlich, authentisch und unabhängig über Videospiele zu schreiben

<https://www.videospielgeschichten.de>