

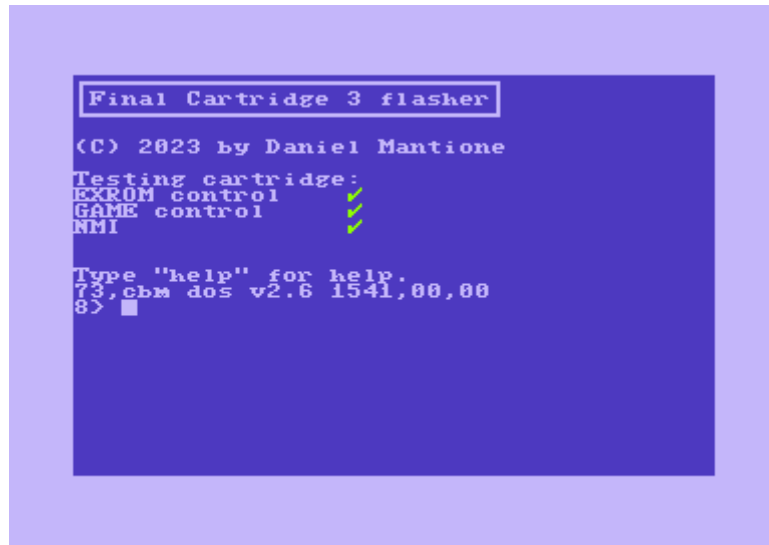
Final Cartridge III Flash-Utility

Benutzerhandbuch

Ein Utility zum Auslesen und Flashen von Final Cartridge III Modulen

von Daniël Mantione

März 2026



Einleitung

Dieses Handbuch beschreibt das Final Cartridge III Flash-Utility. Das Programm wurde als Begleitung zu meinem Final Cartridge III 101% Design geschrieben, einem Nachbau der klassischen Final Cartridge III Hardware, jedoch mit Flash-ROM anstelle von Festwertspeicher. Dies ermöglicht dem Endbenutzer, seine bevorzugte Firmware auf dem Modul zu installieren und zu aktualisieren.

Das Programm kann das ROM von jedem Final Cartridge III kompatiblen Modul auslesen, kann aber nur in den Flash-Speicher eines Final Cartridge III 101% schreiben. Flashen und Auslesen erfolgt von/auf Dateien auf Speichermedien, die an Ihren Commodore Computer angeschlossen sind.

Der Commodore 64 und Commodore 128 werden in diesem Handbuch fortan als C64 und C128 bezeichnet. Final Cartridge III wird fortan als FC3 und Final Cartridge III 101% als FC3101% bezeichnet.

Programmierkompatibilität

Die Schaltung des FC3101% ist gut durchdacht und ähnelt eher der des EasyFlash-Moduls als der minimalistischen Schaltung von beispielsweise Gmod2-Modulen. Daher sollte das Flashen des FC3101% auf jedem C64-Mainboard funktionieren, einschließlich C128, Ultimate64 und Mega65 mit C64-Core. Wenn Sie EasyFlash-Module flashen können, sollten Sie auch den FC3101% flashen können.

Start und Beenden

Das Flash-Utility kann von Diskette geladen werden oder sich selbst in das Modul flashen. Wenn das Tool in das Modul geflasht ist, startet es bei einem Reset automatisch.

Das Flash-Utility führt einen kleinen Test des Bank-Switch-Registers durch, um zu prüfen, ob ein Modul vorhanden ist,

und testet dessen wesentliche Funktionen.

Wenn dieser Test fehlschlägt, wird dies nicht als Fehlerzustand betrachtet, aber das Tool wird versuchen fortzufahren. Sie können dann nichts flashen, aber vielleicht noch das ROM eines FC3-Moduls auslesen.

Beim Beenden kehrt der Computer zu BASIC zurück, oder falls vom Modul gestartet, setzt der Computer den normalen Bootvorgang fort. Das Tool ist eine sehr gutartige Anwendung, die nicht mit Variablen im Speicher des Computers interferiert, die vom Betriebssystem verwendet werden. Nach dem Beenden befindet sich der Computer in einem Zustand, in dem BASIC genauso funktionsfähig ist wie vor dem Start. Das Tool überschreibt auch nicht seinen eigenen Speicherinhalt: Sie können es beenden und erneut ausführen, ohne es von Diskette neu laden zu müssen.

Das Flash-Utility verwendet die High-Level KERNAL API für die Benutzeroberfläche und die Low-Level KERNAL API für Device-operationen. Dies bedeutet, dass das Tool vollen Nutzen aus benutzerdefinierten KERNALs ziehen kann – JiffyDOS, SpeedDOS oder DolphinDOS sind kein Problem.

Da ein 1541-Laufwerk standardmäßig langsam ist, wird ein KERNAL-basierter Schnelllader sogar empfohlen, aber das Tool funktioniert natürlich auch mit einem Standard-C64 und 1541, es wird nur etwas langsam sein... aber Sie können den UI- Befehl des Laufwerks verwenden, um es etwas zu beschleunigen. Das Tool hat speziellen Code zur Unterstützung und deaktiviert den Bildschirm während TALK-Operationen.

FC3-Deaktivierung

Wenn Sie Ihren Computer mit eingestecktem FC3 starten, hängt sich dieser in die BASIC- und KERNAL-Vektoren ein, die in den ROM-Speicher des Moduls zeigen. Das Flash-Utility schreibt in Ihren Final Cartridge III ROM-Speicher. Daher ist es keine gute Idee, dass das FC3 aktiv ist, während das Programm läuft; der Computer könnte abstürzen.

Daher deaktiviert das Flash-Programm das FC3, wenn Sie es starten. Das verwendete Verfahren zur Deaktivierung des FC3 ist dasselbe wie beim KILL-Befehl, mit dem Unterschied, dass der KILL-Befehl sowohl die FC3-Firmware deinstalliert als auch die FC3-Hardware deaktiviert. Das Flash-Programm deinstalliert die FC3-Firmware, lässt aber die Modul-Hardware aktiviert (da diese zum Flashen benötigt wird).

DOS-Befehlszeile

Nach dem Start des Programms gelangen Sie in eine DOS-ähnliche Befehlszeilenoberfläche. Es zeigt einen Prompt mit der aktuellen Device-nummer an:

8>

... und Sie können Commodore DOS-Befehle eingeben, die an das Gerät gesendet werden, z.B.

8> n:newdisk,aa

... um eine Diskette zu formatieren.

Um zu einem anderen Device zu wechseln, geben Sie die Device-nummer ein:

8> 10

10>

Um ein Directory anzuzeigen, geben Sie \$ ein:

8> \$

```

Final Cartridge 3 flasher

(C) 2023 by Daniel Mantione

Testing cartridge:
EXROM control    ✓✓
GAME control    ✓✓
NMI              ✓✓

Type "help" for help.
73,cbm dos v2.6 1541,00,00
8> n:testdisk,aa
00, ok,00,00
8> $
0 "testdisk" "aa 2a"
664 blocks free.
00, ok,00,00
8>

```

Die DOS-Befehlszeilenoberfläche wurde gewählt, um Ihnen die volle Kontrolle über Ihre Speichergeräte zu geben, während Sie mit dem Tool arbeiten. Ein Menüsystem wäre immer unvollständig gewesen.

Um die eigentliche Arbeit zu erledigen, unterstützt das Tool mehrere spezielle Befehle, von denen Sie mit dem Befehl "help" eine Kurzbeschreibung erhalten können:

```

73,cbm dos v2.6 1541,00,00
8> n:testdisk,aa
00, ok,00,00
8> $
0 "testdisk" "aa 2a"
664 blocks free.
00, ok,00,00
8> help
Commands you enter are passed to the de-
vice as DOS commands. To switch between
devices, enter their device number.

Special commands:
* = wildcard supported
help - This help text.
$ - directory*
file:fn - Set "fn" as file to flash*
seca:n - "n" = first sector to flash
secz:n - "n" = last sector to flash
fdump - Dump flash to file
tdump - Dump this tool to file
won - Enable write commands
exit - Boot to BASIC
00, ok,00,00
8>

```

Lesen oder Programmieren des gesamten Flash-Speichers

Das originale Final Cartridge III hatte 64KB Firmware. Das Modul kann einfach auf 256KB erweitert werden und der FC3101% verfügt daher über 256KB Flash-ROM. Je nach Größe der Firmware, die Sie flashen möchten, könnte die Firmware zu groß für klassische Disketten sein. Beispielsweise kann ein 1541-Diskettenlaufwerk nur 170KB Daten auf einer Diskettenseite speichern. Um den gesamten Flash-Speicher in einem Durchgang lesen oder programmieren zu können, benötigen Sie ein C64-Speichermedium, das Dateien mit einer Größe von 256KB oder 1033 Blöcken in der C64-Terminologie aufnehmen kann.

Diese Speichermedien gibt es. Beispielsweise kann das Commodore 1571 340KB auf einer 5,25"-Diskette speichern und das Commodore 1581 kann 720KB auf einer 3,5"-Diskette speichern. Das beliebte SD2IEC hat ebenfalls keine Probleme mit 256KB-Dateien und der 1541 Ultimate sowie Ultimate64 unterstützen ein IEC-Gerät, das direkten Zugriff auf USB-Sticks bietet, die ebenfalls 256KB-Dateien enthalten können. Wenn Sie also eines dieser Speichermedien haben, können Sie den Flash-Speicher in einem Durchgang lesen oder programmieren.

Wenn Sie ein Modul in die Datei mydump vom Typ SEQ auf Device 10 auslesen möchten, können Sie dies mit den folgenden Befehlen tun.

Wechseln Sie zu Device 10:

10

Legen Sie den Dateinamen fest:

file:mydump

Modul auslesen:

fdump

Wenn der Vorgang abgeschlossen ist, gibt es eine Datei namens "mydump.seq" auf Device 10, die 256KB groß ist. Beachten Sie, dass Commodore-Speichergeräte normalerweise das Überschreiben von Dateien verweigern; die Datei sollte vor dem Start noch nicht existieren.

Wenn Sie eine Datei mycartridge vom Typ SEQ auf Device 10 haben und diese flashen möchten, können Sie dies mit den folgenden Befehlen tun.

Wechseln Sie zu Device 10:

10

Legen Sie den Dateinamen fest:

file:mycartridge

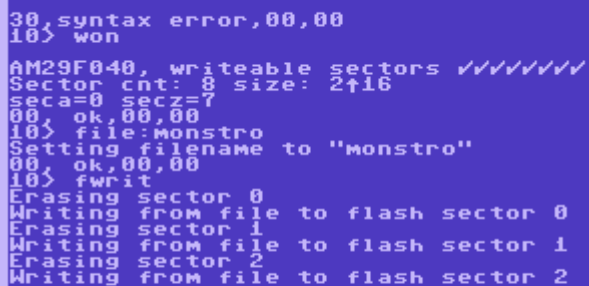
Aktivieren Sie die Schreibbefehle:

won

Modul flashen:

fwrit

Die Programmierung beginnt:



```
30 syntax error,00,00
10> won
AM29F040, writeable sectors ///////////////
Sector cnt: 8 size: 2↑16
seca=0 secz=7
00 ok,00,00
10> file:monstro
Setting filename to "monstro"
00 ok,00,00
10> fwrit
Erasing sector 0
Writing from file to flash sector 0
Erasing sector 1
Writing from file to flash sector 1
Erasing sector 2
Writing from file to flash sector 2
```

Nach einer Weile wurde der Flash-Speicher mit Ihrer Datei programmiert.

Dateien auf Commodore-Speichermedien sind sequentiell, sie müssen von Anfang bis Ende gelesen werden und Sie kennen die genaue Größe der Datei erst, wenn Sie sie vollständig gelesen haben. Daher kann das Tool nicht im Voraus wissen, ob die Datei, die Sie flashen möchten, die richtige Größe hat, und kann daher nicht auf diese Art von Fehlern prüfen. Wenn die Datei nicht genau 256KB groß ist, beginnt das Flashen normal, aber wenn das Tool ein „End of file“ empfängt, schlägt das Flashen mit einem „End of file“-Fehler fehl und ein Teil des Flash-Speichers wurde geschrieben. Wenn die Datei größer als 256KB ist, wird das Flashen erfolgreich abgeschlossen und die ersten 256KB der Datei werden in den Flash-Speicher geschrieben.

Lesen oder Programmieren des Flash-Speichers in Teilen

Falls Sie kein CBM-Speichermedium zur Verfügung haben, das 256KB-Dateien aufnehmen kann, können Sie den Flash-Speicher dennoch von Ihrem Commodore 64/128 aus programmieren, müssen dies jedoch in mehreren Durchgängen tun. Das Flash-Programm ermöglicht es Ihnen, nur einen Teil des Speichers des FC3101%-Moduls zu flashen – Sie können einzelne Sektoren des Flash-ROMs flashen. Die Sektorgröße kann je nach Flash-ROM-Typ unterschiedlich sein. Beispielsweise hat der AM29F002B 64KB-Sektoren, während der SST39SF020 nur 4KB-Sektoren hat. Das bedeutet, wenn Ihr Modul einen AM29F002B enthält, können Sie Vielfache von 64KB programmieren, während Sie mit einem SST39SF020 Vielfache von 4KB programmieren können.

Das Flash-Programm identifiziert Sektoren anhand ihrer Nummer. Bei einem AM29F002B sind die Sektoren also 0..3 nummeriert, während sie bei einem SST39F040 0..63 nummeriert sind.

Sie können völlig frei entscheiden, wie viele Sektoren Sie in einem Durchgang programmieren möchten. Wenn Sie beispielsweise ein 1541-Diskettenlaufwerk haben, könnten Sie das 256KB ROM-Image in 2 Dateien mit je 128KB Größe aufteilen. Jede dieser Dateien passt auf eine 1541-Diskettenseite, und Sie können mit 2 Programmiervorgängen auskommen.

(Eine gute Möglichkeit, Dateien auf einem PC aufzuteilen, ist der Befehl "split", der auf jedem Linux-System verfügbar ist.)

Das Handbuch erklärt nun das Verfahren zum Flashen anhand des ersten Beispiels, d.h. Sie haben das 256KB ROM-Image in 2 Dateien mit je 128KB aufgeteilt, die auf 2 1541-Diskettenseiten gespeichert sind. Das 1541-Laufwerk ist an Device 8 angeschlossen. Die Dateien auf den Disketten haben die Dateinamen "part1", "part2".

Unser Modul enthält angenommenermaßen einen AM29F002B.

Wechseln Sie zu Device 8:

8

Legen Sie die erste Diskette ein. Legen Sie den Dateinamen fest:

file:part1

Aktivieren Sie die Schreibbefehle:

won

Der erste zu flashende Sektor ist Sektor 0:

seca:0

Der letzte zu flashende Sektor ist Sektor 1:

secz:1

Starten Sie den Flash-Vorgang.

fwrit

Legen Sie die nächste Diskette ein (oder drehen Sie sie um). Der erste zu flashende Sektor ist Sektor 2:

seca:2

Der letzte zu flashende Sektor ist Sektor 3:

secz:3

Starten Sie den Flash-Vorgang erneut.

fwrit

Nach Abschluss ist Ihr Modul bereit.

Genau wie beim Schreiben des Flash-Speichers in einem Durchgang gibt es keine Vorabprüfung der Dateigröße. Es liegt in Ihrer eigenen Verantwortung, dass die Dateien die richtige Länge haben.

Das Flash-Tool auf das Modul schreiben

Das Flash-Utility kann sich selbst auf das Modul flashen. Dies kann mit folgenden Befehlen durchgeführt werden:

won

twrit

Nach einem Neustart des Computers startet dieser mit dem Flash-Tool vom Modul.

Das Flash-Tool auf Diskette schreiben

Wenn Sie ein Modul mit vorinstalliertem Flash-Tool erhalten haben und das Tool auf Diskette schreiben möchten, können Sie dies wie folgt tun. Wählen Sie das richtige Device:

10

Legen Sie den Dateinamen fest:

file:fc3flasher

Tool auf Diskette schreiben:

tdump

Befehlsreferenz

\$ - Inhaltsverzeichnis

Dieser Befehl zeigt das Inhaltsverzeichnis auf dem aktiven Device an. Platzhalter werden unterstützt, z.B. zeigt "\$a" alle Dateien an, die mit A beginnen.

file - Dateinamen festlegen

Mit diesem Befehl teilen Sie dem Flash-Tool mit, welche Datei Sie lesen oder schreiben möchten. Alle Auslese- und Flash-Schreibbefehle verwenden diesen Dateinamen. Zum Beispiel:

file:mycartridge.bin

... weist das Tool an, von/auf "mycartridge.bin" zu lesen/schreiben. Platzhalter werden unterstützt und sofort erweitert, z.B. "file:m*".

seca - Startsektor festlegen

Wenn Sie einen Teil des Flash-ROMs programmieren möchten, können Sie mit diesem Befehl den ersten zu programmierenden Sektor des Flash-ROMs festlegen. Zum Beispiel:

seca:2

... um das Flashen bei Sektor 2 zu beginnen.

secz - Endsektor festlegen

Wenn Sie einen Teil des Flash-ROMs programmieren möchten, können Sie mit diesem Befehl den letzten zu programmierenden Sektor des Flash-ROMs festlegen. Zum Beispiel:

secz:3

... um nach dem Flashen von Sektor 3 zu stoppen.

exit - Zurück zu BASIC

Beendet das Flash-Tool. Wenn Sie das Tool von BASIC aus gestartet haben, kehren Sie zu BASIC zurück. Wenn Sie das Tool vom Modul gebootet haben, wird der Bootvorgang des Computers fortgesetzt.

fdump - Flash in Datei auslesen

Dieser Befehl liest den Inhalt des Flash-ROMs in die mit dem Befehl "file" festgelegte Datei aus. Das Auslesen beginnt bei dem mit "seca" angegebenen Flash-ROM-Sektor und endet nach dem mit "secz" angegebenen Flash-ROM-Sektor.

fwrit - Flash aus Datei schreiben

Dieser Befehl schreibt den Inhalt der mit dem Befehl "file" festgelegten Datei in das Flash-ROM. Das Flashen beginnt bei dem mit "seca" angegebenen Flash-ROM-Sektor und endet nach dem mit "secz" angegebenen Flash-ROM-Sektor.

tdump - Flash-Tool in Datei auslesen

Dieser Befehl schreibt das Flash-Tool selbst in die mit dem Befehl "file" angegebene Datei. Dies ist nützlich, wenn Sie das Flash-Tool im Modul-ROM haben, es überschreiben werden und das Tool behalten möchten.

twrit - Flash-Tool in Flash schreiben

Dieser Befehl schreibt das Flash-Tool selbst in das Flash-ROM. Nach Abschluss startet der Computer das Flash-Tool bei

einem Reset automatisch.

won - Schreiben aktivieren

Dieser Befehl aktiviert die Befehle, die in den Flash-Speicher und das EEPROM schreiben. Wenn Sie den Befehl eingeben, versucht das Tool, den Flash-Speicherchip zu erkennen. Weiterhin testet es, ob wiederholte Schreibbefehle zuverlässig am Flash-ROM ankommen.

Das Tool versucht zunächst, 5V-Flash-ROMs ohne Anlegen von 12V zu erkennen. Es prüft auch auf mögliche nicht unterstützte 5V-Flash-ROMs. Wenn kein 5V-Flash-ROM erkannt wird, legt das Tool 12V an und prüft auf das Vorhandensein von 12V-Flash-ROMs.

Nur wenn ein unterstützter Chip erkannt wird und das Flash-ROM Befehle zuverlässig empfängt, wird die Schreibfunktion aktiviert. Der Grund, warum Sie die Schreibbefehle explizit aktivieren müssen, ist, dass das Tool auf diese Weise bei Schreibinkompatibilitäten weiterhin zum Auslesen von Modulen nützlich ist.

Dateiformate

Das Flash-Utility arbeitet mit binären ROM-Images, das .crt-Dateiformat wird nicht unterstützt. Sie können auf dem PC mit dem cartconv-Utility zwischen Dateiformaten konvertieren. Zum Beispiel:

```
~> cartconv -i cartridge.crt -o cartridge.bin  
~> cartconv -i cartridge.bin -t fc3 -o cartridge.crt
```

Wenn die Datei auf einem C64-Speichermedium gespeichert wird, sollte der Dateityp SEQ sein.

1541 UI- Modus

Das 1541 und viele 1541-kompatible Devices haben einen UI- Befehl, der das Gerät etwas schneller macht, aber dann nicht mehr korrekt mit eingeschaltetem Bildschirm funktioniert. Wenn Sie den UI- Befehl eingeben, erkennt das Tool dies und deaktiviert den Bildschirm während TALK-Operationen mit dem Gerät. Wenn Sie also keinen KERNAL-Schnelllader haben, können Sie dennoch von etwas schnelleren Diskettenzugriffen profitieren.

Um zu vermeiden, dass das Tool verfolgen muss, welches Gerät sich im UI- Modus befindet und welches nicht, können Sie nicht mehr zu einem anderen Gerät wechseln, wenn Sie UI- aktivieren. Geben Sie UI+ ein, bevor Sie zu einem anderen Gerät wechseln. Aufgrund der Code-Einfachheit gibt es keine Fehlermeldungen; stattdessen werden Anfragen zum Wechseln zu anderen Geräten nicht mehr als Befehle erkannt.

UI- ist immer noch recht langsam und sollte verwendet werden, wenn nichts Besseres verfügbar ist: Ein KERNAL-Schnelllader ist der richtige Weg, um mehr Leistung aus dem seriellen CBM-Bus herauszuholen.

Unterstützte Flash-ROMs

Ihr FC3 101% enthält einen der vielen unterstützten Flash-ROMs. Der FC3101% ist mit 3 Arten von Flash-ROMs kompatibel:

- 256KB 5V Flash-ROMs
- 512KB 5V Flash-ROMs
- 256KB Flash-ROMs (12V)

Flash-Speicher muss gelöscht werden, bevor er beschrieben werden kann. Flash-ROMs können in Sektoren unterteilt werden, die einfach einzeln gelöscht werden können. Daher müssen Sie bei einem Flash-ROM mit mehreren Sektoren nicht den gesamten Flash-Speicher auf einmal schreiben.

Das Flash-Tool versteht das Konzept der Sektoren und ermöglicht es Ihnen, den Flash-Speicher teilweise zu beschreiben.

256KB 5V Flash-ROMs

256KB Flash-ROMs sind etwas seltsam: Nur sehr wenige Hersteller haben einen "normalen" 256KB Flash-ROM. Beispielsweise hat AMD einen 512KB AM29F040 hergestellt, aber keinen 256KB AM29F020. Stattdessen gibt es sogenannte "Non-Uniform-Sector" Flash-ROMs, wie zum Beispiel den AM29F002B. Diese Flash-ROMs haben Sektoren mit unterschiedlichen Größen, mit denen Sie einen Bootsektor am Anfang oder Ende des Flash-Speichers erstellen können. Ein Bootsektor ist für den Zweck des FC3101% völlig nutzlos.

Das Flash-Tool unterstützt Non-Uniform-Sector Flash-ROMs, lässt sie aber als einheitliche Flash-ROMs erscheinen. Beispielsweise erscheint der AM29F002B als hätte er 4 Sektoren mit je 64KB.

Viele 256KB Flash-ROMs haben einen Schreibschutzmechanismus, der durch Anlegen von 12V außer Kraft gesetzt werden kann. Das Flash-Utility geht davon aus, dass Sie, wenn Sie das Utility verwenden, wirklich in den Flash-Speicher schreiben möchten, und legt 12V an, wenn es einen schreibgeschützten Sektor erkennt.

Derzeit unterstützt das Tool die folgenden Flash-ROMs:

A29002T
A29002U
AM29F002T
AM29F002B
AT49F002A
AT49F002T
AT49F020
EN29F002T
EN29F002B
HY29F002T
HY29F002B
M29F002T
M29F002B
MBM29F002T
MBM29F002B
MX29F002T
MX29F002B
SST39SF020
SST39VF020
W39L020
W49F002B
W49F002U

Einige der oben genannten Flash-ROMs sind tatsächlich 3,3V-Flash-ROMs und keine 5V-Typen. Das Flash-Utility unterstützt deren Algorithmus, aber der FC3101% hat keine Spannungswandler-Hardware für 3,3V-Flash-ROMs an Bord.

512KB 5V Flash-ROMs

5V-Flash-ROMs haben im Allgemeinen einheitliche Sektoren und sind daher weniger komplex als 256KB-Flash-ROMs. Es kann attraktiv sein, 512KB-Flash-ROMs zu verwenden und nur 256KB zu nutzen. Die letzte Adressleitung verwendet den Pin, den 256KB-Flash-ROMs zum Anlegen von 12V nutzen. Wenn das Tool daher einen 512KB-Flash-ROM erkennt, schaltet es den 12V-Generator nicht ein.

Derzeit unterstützt das Tool die folgenden Flash-ROMs:

AM29F040 (einschließlich Klone AS29F040 und TMS29F040)
A29040B (einschließlich Klon AS29CF040)
EN29F040
HY29F040
M29F040
MBM29F040
MX29F040
SST39SF040
SST39VF040
W39L040

Auch hier sind einige der oben genannten Flash-ROMs tatsächlich 3,3V-Flash-ROMs und keine 5V-Typen. Das Flash-Utility unterstützt deren Algorithmus, aber der FC3101% hat keine Spannungswandler-Hardware für 3,3V-Flash-ROMs an Bord.

Flash-ROMs mit Sektorschutzfunktion müssen entsperrt werden. Das Tool erkennt geschützte Sektoren und verweigert deren Programmierung, kann diese Sektoren aber nicht selbst entsperren.

12V 256KB Flash-ROMs

Im Gegensatz zu 5V 256KB Flash-ROMs sind ihre 12V-Gegenstücke einheitlich und nicht in Sektoren unterteilt. Daher lässt das Flash-Tool ein 12V-Flash-ROM als einen einzelnen Sektor mit einer Größe von 256KB erscheinen.

Eine Ausnahme bilden MX28F002B/MX28F002T, diese Flash-ROMs haben nicht einheitliche Sektoren und erscheinen als zwei Sektoren mit je 128KB Größe.

Derzeit unterstützt das Tool die folgenden Flash-ROMs:

AM28F020
CAT28F020
i28F020
IS28F020
M28F201
MX28F002T
MX28F002B

Häufig auftretende Probleme

Fehlermeldung 62, file not found,00,00

Wenn diese Fehlermeldung erscheint, haben Sie tatsächlich eine Datei angegeben, die nicht existiert. Tipps:

- Sehen Sie sich das Inhaltsverzeichnis mit dem Befehl \$ an.
- Achten Sie auf Dateiendungen – wenn die Datei firmware.bin heißt, müssen Sie file:firmware.bin angeben
- Verwenden Sie einen Platzhalter. Wenn es eine Datei namens firmware.bin auf Ihrem Speichergerät gibt, können Sie sie möglicherweise mit file:firm* auswählen

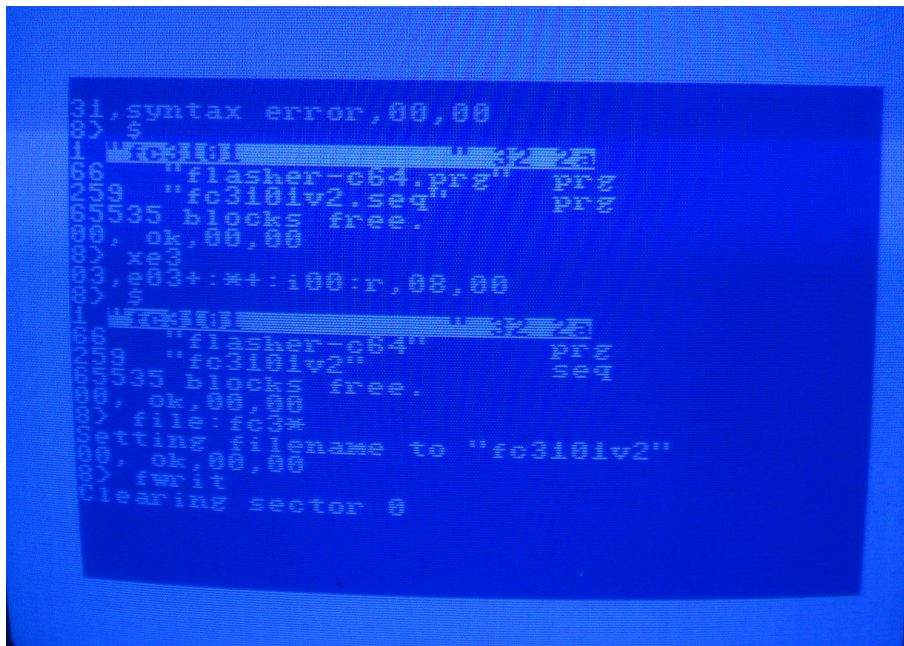
Fehlermeldung 64, file type mismatch,00,00

Wenn diese Fehlermeldung erscheint, ist der Commodore-Dateityp Ihrer Datei nicht SEQ. Ändern Sie den Dateityp auf SEQ, um das Problem zu lösen.

Verwendung eines SD2IEC zum Flashen

Falls Sie von einem SD2IEC flashen und das Firmware-Verzeichnis auf eine SD-Karte kopieren (Sie verwenden kein D64-, D71- oder D81-Image), stoßen Sie auf das Hindernis, dass das SD2IEC standardmäßig alle Dateien als PRG anlegt. Das Flash-Utility erwartet eine Datei vom Typ SEQ. Wenn Sie eine Datei namens „FC3101V2.SEQ“ auf die SD-Karte kopieren, sieht der Commodore 64 eine Datei „FC3101V2.SEQ“ mit dem Typ PRG und daher ist es nicht möglich, die Datei zu flashen.

Das SD2IEC unterstützt einen Befehl „XE3“, um das SD2IEC anzuweisen, anhand der Dateiendungen den C64-Dateityp zu bestimmen. Wenn Sie den Befehl „XE3“ eingeben, sieht der Commodore 64 eine Datei namens „FC3101V2“ mit dem Typ SEQ.



Nach Eingabe des „XE3“-Befehls sollten Sie keine Probleme beim Flashen der Firmware mit einem SD2IEC haben.

Erstellen eines Images mit der Firmware

Wenn Sie ein D64-, D71- oder D81-Image mit der Firmware und dem Flash-Utility darin erstellen müssen, können Sie dies mit dem c1541-Utility von VICE wie folgt tun:

```
linux> c1541
c1541 #8> format fc3firmware,f3 d71 fc3firmware.d71 8
c1541 #8> write fc3_101v2.bin fc3101v2,s
c1541 #8> write flasher-c64.prg fc3flasher
exit
```

Dies erstellt ein Image fc3firmware.d71 und speichert die Dateien „fc3_101v2.bin“ und „flasher-c64.prg“ als „fc3101v2“ und „fc3flasher“ im Image. Bitte beachten Sie die Verwendung von „s“, um eine Datei vom Typ SEQ zu erstellen.

Sie können dann das Image fc3firmware.d71 mit Ihrer bevorzugten Speicherlösung verwenden, um auf die Dateien auf einem C64 zuzugreifen.

Flash-Chip nicht erkannt

Wenn das Utility nach der Eingabe von „won“ keinen Flash-Chip erkennen kann, liegt wirklich ein Problem vor. Bitte melden Sie das Problem, damit wir Kenntnis über diese Probleme erlangen und Schritte zur Verbesserung der Hardware unternehmen können.

Mögliche Ursachen für diesen Fehler:

- Die 12V-Schaltung auf dem Modul funktioniert aus irgendeinem Grund nicht
- Es liegt ein Bus-Timing-Problem vor. Der FC3101% benötigt, dass der Adressbus des C64 120ns nach dem PHI2-Anstieg stabil ist und der Datenbus bis zu 35ns nach dem PHI2-Abfall stabil sein sollte. Dieses Timing ist recht entspannt und sollte mit allen klassischen C64- und C128-Computern kompatibel sein. Einige „moderne“ C64-Computer sind möglicherweise nicht ausreichend mit dem originalen C64 kompatibel.
- Es gibt einen schlechten Kontakt – entweder hat das Modul an einigen Pins keinen richtigen Kontakt mit dem Modulport, oder das Flash-ROM im PLCC-Sockel hat keinen richtigen Kontakt.

Dinge, die Sie ausprobieren können:

- Überprüfen Sie die 5V-Spannung Ihres C64. Bei starker Unterspannung kann die Modul-Hardware

möglicherweise keine 12V erzeugen.

- Versuchen Sie einen anderen Commodore 64.
- Wenn Sie einen Mega65 verwenden: Verwenden Sie den C64-Core anstelle des nativen Cores.
- Wenn Sie einen Ultimate64 verwenden: Experimentieren Sie mit dem Cartridge Bus Mode.

Leerer blauer Bildschirm

Wenn Sie einen leeren blauen Bildschirm sehen, versucht der C64, das Modul zu starten, aber der Startcode des Moduls stürzt ab. Dies kann folgende Ursachen haben:

- Fehlerhafte Firmware auf dem Modul. Wenn die Firmware auf dem Modul ungültig ist, tritt dieses Problem auf. Installieren Sie die Firmware auf dem Modul neu.
- Schlechter Kontakt. Wenn das Modul keinen richtigen Kontakt im Modulschacht Ihres C64 herstellt, tritt dieses Problem auf. Versuchen Sie, den schlechten Kontakt zu beheben.

Wenn Sie einen Commodore 128 haben, können Sie die Firmware auf dem Modul manuell mit dem eingebauten Monitor überprüfen. Stellen Sie den Schalter auf dem Modul in den „PROG“-Modus. Rufen Sie dann den Monitor mit dem Commodore 128 Befehl „MON“ auf und zeigen Sie den ROM-Inhalt mit „M88000“ an. Prüfen Sie, ob die Ausgabe stabil ist und ob Sie die CBM80-Signatur sehen können.