MUSIK-EDITOR

für den Commodore 64

Dreistimmiger Tracker für den SID 6581/8580
Dateiformat CMUS · Player MUSKIT

Handbuch zur Fassung V1 · Stand 8. September 2026

Idee, Umsetzung & Projektmanagement von Denis Friske

Inhaltsverzeichnis

1. Wozu der Musik-Editor da ist

2. Starten

3. Der Pattern-Editor

4. Der Instrument-Editor

5. Sequenz und Kopfdaten

6. Laden und Speichern

7. Die Hilfeschirme

8. Vollständige Tastenreferenz

9. Die Effekte 0 bis 15

Anhang A: Das Dateiformat CMUS

Anhang B: MUSKIT im eigenen Spiel benutzen

Anhang C: Die Fallen

Anhang D: Wenn etwas schiefgeht

1. Wozu der Musik-Editor da ist

Der Musik-Editor ist ein Tracker: ein Programm, in dem Musik nicht als Noten auf Linien geschrieben wird, sondern als Tabelle. Jede Zeile der Tabelle ist ein Zeitschritt, jede Spalte eine der drei Stimmen des SID. Man liest sie von oben nach unten, so wie ein Klavierrollenband läuft.

Erzeugt wird damit ein Musikstück im Format **CMUS**, das ein Spiel später mit dem mitgelieferten Player **MUSKIT** abspielt. Der Editor und der Player sind zwei getrennte Dinge: der Editor ist ein BASIC-Programm zum Schreiben, der Player ist Maschinencode, der im Spiel nebenher im Interrupt läuft und dort nichts weiter braucht als den Song im Speicher.

> **Was hier NICHT hineingehört:** kurze Spielgeräusche — Treffer, Tür, Sprung, Münze. Dafür gibt es den Sound-Editor. Musik und Geräusche teilen sich denselben SID und laufen deshalb nie gleichzeitig; wie sie sich abwechseln, steht in Anhang B.

1.1 Was auf der Arbeitsdiskette liegt

Datei	Was es ist
`MUSIKEDITOR`	der Editor selbst, ein BASIC-Programm
`MUSKIT`	der Player, Maschinencode für 40960–43263 (unter dem BASIC-ROM)
`MUSTRAMP`	Trampolin bei 53120–53247: die Einsprünge, über die BASIC den Player erreicht
`MDISKMOD`	Lade- und Speichermodul für CMUS-Dateien
`MU-...`	gespeicherte Musikstücke

Der Player liegt unter dem BASIC-ROM, weil dort Platz ist, den BASIC nicht braucht. Erreichbar ist er trotzdem, nämlich über das Trampolin MUSTRAMP im normalen RAM — BASIC ruft immer das Trampolin, nie den Player direkt.

1.2 Die Grenze von 21 Patterns

Ein Song darf höchstens **21 Patterns** haben. Das ist keine Bequemlichkeit, sondern Arithmetik: MUSKIT, MAPKIT und der Song teilen sich dieselben 8192 Byte unter dem BASIC-ROM. Der Editor lässt es gar nicht erst so weit kommen und meldet beim Anhängen ab 21 `MEHR ALS 21 PATTERNS GEHEN NICHT`.

> **Ältere Dateien:** bis zum 14. August 2026 waren 28 Patterns erlaubt, davor sogar 32. Eine alte Datei mit mehr als 21 Patterns wird beim Laden mit `ABGELEHNT (MUERR 3)` zurückgewiesen. Sie ist nicht verloren — sie muss in einer älteren Fassung geöffnet, auf 21 Patterns gekürzt und neu gespeichert werden.

2. Starten

Von der Arbeitsdiskette:

```
LOAD"MUSIKEDITOR",8
RUN
```

Der Editor lädt dann selbst nach, was er braucht, und meldet sich mit dem Pattern-Editor. Das dauert einen Moment: das Programm ist rund 22 KB groß, und eine 1541 liest das nicht in einer Sekunde.

> **Für Testläufe im Emulator:** ein Bildschirmfoto nach zu kurzer Laufzeit zeigt nur `RUN` und sieht dann aus wie ein Absturz. Es ist keiner — das Laden war schlicht noch nicht fertig. In VICE braucht der Start rund 150 Millionen Taktzyklen, bis das Gitter steht.

3. Der Pattern-Editor

Das ist der Hauptschirm, auf dem man die meiste Zeit verbringt. Alles andere erreicht man von hier aus und kommt hierher zurück.

```
MUSIK-EDITOR V1
PAT1/2 TMP2 OKT4 INS0 ST1 >1:9
ZL  STIMME1  STIMME2  STIMME3
00> C-4 0 - E-4 0 - C-3 0 -
01  --- - - --- - - --- - -
02  --- - - --- - - --- - -
   ...
08  A-3 0 - C-4 0 - A-2 0 -
   ...
16  C-4 0 - E-4 0 - C-3 0 -
   ...
19  --- - - --- - - --- - -
--------------------------------------
F1 HILFE F2 INS F3 SPIEL F4 SEQ F7 F8
```

3.1 Die Kopfzeile

Anzeige	Bedeutung
`PAT1/2`	erstes von zwei Patterns. Beide Zahlen zählen ab 1 — bei zwei Patterns sind `PAT1/2` und `PAT2/2` die einzigen Werte, kein `PAT0/2`.
`TMP2`	Tempo: Frames je Patternzeile (klein = schnell)
`OKT4`	eingestellte Oktave für die Notentasten
`INS0`	Instrument, das eine neu gesetzte Note bekommt
`ST1`	Stimme, in der der Cursor steht: 1, 2 oder 3
`>1:9`	nur während der Wiedergabe: Sequenzposition 1, Patternzeile 9

> **Welches Feld — Note, Instrument oder Effekt — gerade ausgewählt ist, steht nicht in der Kopfzeile, sondern wird direkt im Gitter gelb hervorgehoben.** `CRSR` links/rechts wandert durch die drei Felder einer Stimme und springt am Rand zur Nachbarstimme; die gelbe Markierung zeigt dabei jederzeit, wo eine Zifferneingabe oder eine Note landen würde.

3.2 Ein Pattern hat 32 Zeilen, der Bildschirm 25

Sichtbar sind 20 der 32 Zeilen. Der Cursor bewegt sich frei darin, und erst wenn er oben oder unten anstößt, wandert das Fenster mit. Das ist das Verhalten, das jeder Tracker hat.

3.3 Die Zelle: `NNN I E`

Jede Stimme belegt sieben Zeichen — drei für die Note, eines für das Instrument, eines für den Effekt.

Feld	Anzeige	Bedeutung
NNN	`C-4`, `C#4`	Note und Oktave. Halbtöne bekommen das `#` statt eines Bindestrichs, damit drei Zeichen reichen.
NNN	`---`	leer: hier ändert sich nichts, die laufende Note klingt weiter
NNN	`===`	Note aus: das Gate fällt, die Release-Phase beginnt
NNN	`???`	unbrauchbarer Notenwert aus einer fremden Datei — siehe Anhang C
I	`0`–`F`	Instrumentnummer, hexadezimal, damit ein Zeichen reicht
E	`0`–`F`, `-`	Effektnummer; `-` steht für „kein Effekt"

Die Instrumentnummer wird nur ausgewertet, wenn in derselben Zelle auch eine Note steht. Eine leere Zelle mit Effekt lässt das laufende Instrument der Stimme unberührt.

3.4 Noten eingeben

Zwei Tastenreihen, wie in jedem Tracker. Die untere Reihe spielt in der eingestellten Oktave, die obere eine Oktave höher:

```
untere Oktave   Z  S  X  D  C  V  G  B  H  N  J  M   , L . ; /
                C  C# D  D# E  F  F# G  G# A  A# H   C C# D D# E

obere Oktave    Q  2  W  3  E  R  5  T  6  Y  7  U   I 9 O 0 P
                C  C# D  D# E  F  F# G  G# A  A# H   C C# D D# E
```

Steht der Cursor auf der Instrument- oder Effektspalte, setzen die Tasten **0**–**9** und **A**–**F** direkt den Wert; die Notentasten sind dort wirkungslos.

Taste	Wirkung
+ **−**	Oktave tiefer / höher (0–7)
LEERTASTE	„Note aus" setzen
DEL	Zelle löschen — Note, Instrument und Effekt auf 0
CRSR	Zeile, Spalte und über die Stimmengrenze hinweg

3.5 Patterns verwalten

Taste	Wirkung
`<` `>`	voriges / nächstes Pattern
`@`	ein neues, leeres Pattern anhängen (bis höchstens 21)
`SHIFT-C`	aktuelles Pattern in die Zwischenablage
`SHIFT-V`	Zwischenablage in das aktuelle Pattern

Im Instrument-Editor wirken `SHIFT-C` und `SHIFT-V` auf Instrumente statt auf Patterns.

3.6 Abspielen

Taste	Wirkung
`F3`	Song ab Sequenzposition 0 abspielen; nochmal `F3` hält an
`F5`	ab der aktuellen Sequenzposition abspielen

Während die Musik läuft, bleibt der Editor bedienbar. Neu eingetragene Noten sind sofort zu hören, weil der Player denselben Songpuffer liest, den das BASIC beschreibt. Das ist der angenehme Nebeneffekt der Entscheidung, den Song im Editor *nicht* unter das ROM zu legen.

Nachgezogen wird während der Wiedergabe nur die Zeilenmarkierung, nicht der ganze Bildschirm. Ein voller Neuaufbau sechzigmal je Sekunde ist in BASIC unmöglich; der Player im Interrupt ist der schnelle Teil, die Anzeige der langsame.

4. Der Instrument-Editor

Erreichbar mit `F2`, zurück mit `F2` oder `F1`. Ein Instrument ist eine statische SID-Einstellung plus Vibrato und Sweep — mehr nicht. Es gibt 16 davon, nummeriert 0 bis 15.

```
INSTRUMENT-EDITOR
INSTRUMENT 0   CRSR LINKS UND RECHTS

  ATTACK                   0
  DECAY                    8
  SUSTAIN                 15
  RELEASE                 10
  WELLENFORM           SAEGE
  PULSWEITE             2048
  PW-SWEEP                 0
  VIB-TIEFE                0
  VIB-TEMPO                0
  RINGMOD                AUS
  SYNC                   AUS
  FILTER                 AUS
  HARD-RESTART            AN
  SLIDE                    0
```

Es wird immer nur *ein* Instrument gezeigt, nicht eine Liste aller sechzehn. Grund ist die Zeichengeschwindigkeit: eine Textzeile kostet in BASIC rund 67 Millisekunden, und eine Liste neben den vierzehn Feldern müsste bei jedem Wechsel komplett neu gezeichnet werden.

4.1 Bedienung

Taste	Wirkung
CRSR hoch/runter	Feld wählen
CRSR links/rechts	Instrument 0–15 wählen
+ **−**	Wert ändern; Pulsweite in 64er-Schritten, PW-Sweep in 4er-Schritten, Schalter kippen
LEERTASTE	Testton

Der Testton ist die wichtigste Funktion dieses Schirms — ohne sofortiges Hören ist ADSR-Einstellung Ratearbeit. Er geht nicht über MUSKIT, sondern schreibt direkt in den SID: C-4, Gate für 0,6 Sekunden, danach Release und Stille.

4.2 Die vierzehn Felder

Feld	Bereich	Bedeutung
ATTACK	0–15	Anstiegszeit der Hüllkurve
DECAY	0–15	Abfallzeit bis zum Haltepegel
SUSTAIN	0–15	Haltepegel
RELEASE	0–15	Ausklingzeit nach „Note aus"
WELLENFORM	DREIECK, SAEGE, PULS, RAUSCH	Klangfarbe
PULSWEITE	0–4095	nur bei PULS von Bedeutung
PW-SWEEP	−128…127	Wanderung der Pulsweite je Frame; wirkt nur bei Effekt 9
VIB-TIEFE	0–15	Vibratotiefe; wirkt nur bei Effekt 4
VIB-TEMPO	0–15	Vibratogeschwindigkeit
RINGMOD	AN/AUS	Ringmodulation mit der Nachbarstimme
SYNC	AN/AUS	Synchronisation mit der Nachbarstimme
FILTER	AN/AUS	diese Stimme durch das Filter schicken
HARD-RESTART	AN/AUS	Vorgabe AN — siehe unten
SLIDE	0–15	Schrittweite für die Effekte 1 und 2; der Player rechnet ×16

4.3 Hard-Restart

Vor jeder neuen Note setzt der Player für einen Frame die Hüllkurve auf 0 und das Gate aus. Ohne das bleibt der Hüllkurvengenerator des SID bei schnellen Notenfolgen hängen, und einzelne Noten fallen stumm aus. Das ist ein bekannter Hardwarefehler des Chips, kein Programmfehler — deshalb ist Hard-Restart die Vorgabe und muss ausdrücklich abgeschaltet werden, etwa für weich gebundene Basslinien.

> **Hard-Restart und Tempo 1 vertragen sich nicht.** Der Hard-Restart kostet den ersten der beiden Frames einer neuen Note. Bei Tempo 1 — eine Zeile je Frame — bleibt davon nichts übrig: die Note wäre nur noch Hard-Restart. Tempo 1 gehört mit abgeschaltetem Hard-Restart benutzt oder gar nicht. Ab Tempo 2 ist es unkritisch.

4.4 Zwei Warnungen in der Statuszeile

Beide blockieren nicht, sie weisen nur hin:

- **Wellenform PULS mit Pulsweite 0 oder 4095.** Die Stimme bleibt stumm. Das ist der häufigste Anfängerfehler am SID.
- **RINGMOD oder SYNC gesetzt.** Beide brauchen eine klingende Nachbarstimme — Stimme 1 greift auf Stimme 3 zurück, Stimme 2 auf 1, Stimme 3 auf 2. Ohne die passiert nichts Hörbares.

5. Sequenz und Kopfdaten

Erreichbar mit **F4**. Hier steht, in welcher Reihenfolge die Patterns gespielt werden, und hier stehen die Eigenschaften des ganzen Stücks.

```
SEQUENZ UND KOPFDATEN
SONG: TESTSONG           N AENDERT NAME

00:>00 01 00 01 .. .. .. ..
08: .. .. .. .. .. .. .. ..
16: .. .. .. .. .. .. .. ..
24: .. .. .. .. .. .. .. ..
32: .. .. .. .. .. .. .. ..
```

5.1 Zwei Zonen, RETURN schaltet um

Sequenzraster und Kopfdaten brauchen verschiedene Cursorlogik — im Raster bewegt **CRSR** hoch/runter um *acht* Positionen, in den Kopfdaten um *ein* Feld. Zwei getrennte Cursor gleichzeitig auf einem Schirm wären schwerer zu verstehen als ein Umschalter, deshalb schaltet **RETURN** zwischen den Zonen um.

Im Sequenzraster

Taste	Wirkung
CRSR links/rechts	eine Position
CRSR hoch/runter	acht Positionen
+ **−**	Patternnummer an dieser Position
L	Loop-Punkt auf die Cursorposition legen

Darstellung	Bedeutung
`03`	Position spielt Pattern 3
`..`	Position jenseits der Sequenzlänge
`*03`	Loop-Punkt
`>03`	Cursor — gewinnt, wenn beides zusammenfällt

Ein Pattern darf mehrfach vorkommen. Genau das ist der Zweck der Sequenz und der Grund, warum ein Stück mit vier Patterns mehrere Minuten dauern kann.

In den Kopfdaten

Feld	Bereich	Bedeutung
TEMPO	1–255	Frames je Patternzeile. 3 ist sehr schnell, 6 die Vorgabe, 12 sehr getragen.
PATTERNS	1–21	Anzahl der Patterns im Stück
SEQUENZLAENGE	1–64	wie viele Positionen der Sequenz gültig sind
LOOP-POSITION	0…Länge−1	ab hier wird wiederholt; nur wirksam, wenn SCHLEIFE an ist
LAUTSTAERKE	0–15	Grundlautstärke des Stücks
FILTER-CUTOFF	0–255	Grenzfrequenz. Der SID hat nur *ein* Filter für alle drei Stimmen — deshalb gehört es dem Stück, nicht dem Instrument.
RESONANZ	0–15	Filterresonanz
FILTERTYP	AUS, TIEF, BAND, HOCH	Filtercharakteristik
SCHLEIFE	AN/AUS	AUS heißt: einmal durchspielen, dann Stille

`N` ändert den Songnamen, `F4` oder `F1` führt zurück zum Pattern-Editor. Beim Verlassen läuft MUINIT, damit der Player die geänderten Kopfdaten übernimmt.

5.2 Patternzahl ändern

Beim Erhöhen wird das neue Pattern genullt. Beim Verringern werden Sequenzeinträge, die dann ins Leere zeigen würden, auf die letzte gültige Patternnummer heruntergezogen — sonst spielt der Player dort stillschweigend Pattern 0, und der Song klingt anders, als er aussieht.

> **Gesperrt während der Wiedergabe:** Patternzahl und Sequenzlänge lassen sich nicht ändern, solange die Musik läuft, mit rotem Hinweis in der Statuszeile. Der Player könnte sonst mitten in eine gerade verschobene Struktur laufen.

6. Laden und Speichern

Taste	Wirkung
`F7`	Laden. Öffnet die Verzeichnisbox; `CRSR` wählt, `RETURN` lädt, jede andere Taste bricht ab.
`F8`	Speichern. Fragt in der Statuszeile nach dem Namen; `RETURN` speichert, `RETURN` bei leerem Namen bricht ab, `DEL` löscht das letzte Zeichen.

Beim Speichern setzt der Editor das Präfix `MU-` davor, wie es die anderen Werkzeuge des Projekts mit `CH-`, `IT-` und `DQ-` auch tun. Man tippt also nur den Rest. Das Präfix ist kein Schmuck — die Sammeldisketten sortieren die Dateien allein danach ein. Der Songname in den Kopfdaten ist davon unabhängig und darf abweichen.

Beide Funktionen halten vorher die Musik an. Ein Player-Interrupt mitten in einer seriellen Bitfolge ist genau die Störung, die auf echter Hardware sporadisch zuschlägt und im Emulator nicht auffällt.

> Die Verzeichnisbox überdeckt die Zeilen 4 bis 19 und wird nach dem Schließen einfach überzeichnet. Eine Bildschirmsicherung gibt es bewusst nicht — sie würde Speicher kosten, den der Song besser gebrauchen kann.

7. Die Hilfeschirme

`F1` öffnet die Hilfe. Seite 1 zeigt die Tastenbelegung des Pattern-Editors, Seite 2 die Effekttabelle im Klartext samt der beiden SID-Fallen. `LEERTASTE` oder `CRSR` links/rechts blättert, jede andere Taste führt zurück — und zwar in den Schirm, aus dem man gekommen ist.

Die Hilfe steht im Programm, nicht auf der Diskette. Das ist Absicht: von Diskette nachladen hieße, für jeden Blick in die Hilfe die Musik anzuhalten. Für zwei Textseiten wäre das absurd.

8. Vollständige Tastenreferenz

8.1 Pattern-Editor

Taste	Wirkung
Z S X D C V G B H N J M , L . ; /	Noten in der eingestellten Oktave
Q 2 W 3 E R 5 T 6 Y 7 U I 9 O 0 P	Noten eine Oktave höher
CRSR	Zeile, Spalte, Stimme
0–9 A–F	Ziffer für Instrument oder Effekt
LEERTASTE	Note aus
DEL	Zelle löschen
+ −	Oktave tiefer / höher
< >	Pattern wechseln (voriges / nächstes)
@	Pattern anhängen
SHIFT-C	Pattern kopieren
SHIFT-V	Pattern einfügen
F1	Hilfe
F2	Instrument-Schirm
F3	Song spielen, nochmal Stop
F4	Sequenz und Kopfdaten
F5	ab Sequenzposition spielen
F7 F8	Laden, Speichern

8.2 Instrument-Editor

Taste	Wirkung
CRSR hoch/runter	Feld wählen
CRSR links/rechts	Instrument wählen
+ −	Wert ändern
LEERTASTE	Testton
SHIFT-C SHIFT-V	Instrument kopieren / einfügen
F2 F1	zurück

8.3 Sequenz und Kopfdaten

Taste	Wirkung
`RETURN`	zwischen Raster und Kopfdaten umschalten
`CRSR`	im Raster: 1 bzw. 8 Positionen; in den Kopfdaten: ein Feld
`+` `−`	Wert ändern
`L`	Loop-Punkt setzen
`N`	Songnamen ändern
`F4` `F1`	zurück

9. Die Effekte 0 bis 15

Die Effekte sind **parameterlos** — sie passen in ein halbes Byte. Die Feineinstellungen (Vibratotiefe, Slide-Schrittweite, Pulsweiten-Sweep) hängen am Instrument, nicht an der Notenzeile. Das ist der eigentliche Unterschied zwischen diesem Tracker und einem vollen.

Nr.	Wirkung
0	kein Effekt. Beendet auch einen laufenden Slide oder ein Portamento. Vibrato bleibt an — das schaltet nur Effekt 5 aus.
1	Slide aufwärts, Schrittweite aus dem Instrumentfeld SLIDE
2	Slide abwärts, ebenso
3	Portamento: gleitet zur Note in dieser Zelle, statt sie hart zu setzen. Ohne Note wirkungslos.
4	Vibrato ein (Tiefe und Tempo aus dem Instrument)
5	Vibrato aus
6	Arpeggio Dur — wechselt je Frame zwischen 0, +4, +7 Halbtönen
7	Arpeggio Moll — 0, +3, +7
8	Arpeggio Septime — 0, +4, +7, +10
9	Pulsweiten-Sweep ein (Schrittweite aus dem Instrument)
10	Pulsweiten-Sweep aus
11	Gate nicht neu setzen — weiche Bindung an die Vornote
12	diese Stimme in das Filter schicken
13	diese Stimme aus dem Filter nehmen
14	Patternende: Rest überspringen, weiter mit der nächsten Sequenzposition
15	Songende: zur Loop-Position springen oder anhalten

Die Effekte 14 und 15 wirken songweit, gleich in welcher Stimme sie stehen. Stehen sie in mehreren Stimmen derselben Zeile, gilt der niedrigere zuerst — das ist aber ein Bedienfehler, und der Editor färbt ihn rot ein.

> Die Arpeggien sind fest verdrahtet statt frei wählbar. Freie Arpeggien würden ein drittes Byte je Zelle kosten. Wer einen anderen Akkord braucht, schreibt ihn als drei schnelle Einzelnoten.

> **Slides laufen nicht über.** Beide Slides werden bei $FF00 nach oben und $0100 nach unten begrenzt. Ohne diese Grenzen läuft der 16-Bit-Frequenzwert über und die Tonhöhe springt schlagartig auf den anderen Anschlag — beim Effekttest am 14. August 2026 gemessen: ein Abwärts-Slide landete nach gut zwei Sekunden bei 3800 Hz, statt leise auszulaufen. Die Grenzen liegen bewusst nicht bei $FFFF und 0, sondern lassen 256 Einheiten Luft für ein gleichzeitig laufendes Vibrato.

Anhang A: Das Dateiformat CMUS

CMUS ist eine PRG-Datei mit den üblichen zwei Ladeadressbytes am Anfang. **Alle Offsets hier sind relativ zum ersten Datenbyte**, also nach diesen zwei Bytes.

> **Die Ladeadresse in der Datei ist bedeutungslos.** Sie enthält schlicht die Adresse, an der der Song zuletzt lag. Geladen wird immer reloziert — im Editor nach 33000 (sichtbares RAM, per PEEK und POKE erreichbar), im Spiel nach 44800 (unter dem ROM). Dieselbe Datei passt damit in beide Fälle.

A.1 Gesamtaufbau

Offset	Größe	Inhalt
+0000…0031	32	Kopf
+0032…0095	64	Sequenz
+0096…0223	128	Instrumente, 16 × 8 Byte
+0224…	AP × 192	Patterns

> **Dateilänge = 224 + AP × 192** (ohne die zwei Ladeadressbytes)
>
> Kleinste Datei: 224 + 1 × 192 = 416 Byte (2 Blöcke)
> Größte Datei: 224 + 21 × 192 = 4256 Byte (17 Blöcke)

> **Diese Gleichung wird beim Laden geprüft.** Ein Kopf, der mehr Patterns angibt, als die Datei enthält, wurde bis zum 15. August 2026 anstandslos angenommen — der Player las dann hinter dem Dateiende weiter und spielte fremden Speicherinhalt als Noten. Das fällt nicht als Fehler auf, es klingt nur falsch. Seither meldet der Editor `DATEI PASST NICHT ZUM KOPF`.
>
> **Wer CMUS-Dateien per Skript baut, muss die Gleichung selbst einhalten.** MUSKIT kann das nicht prüfen — der Player sieht nie eine Datei, sondern nur einen Zeiger auf Speicher.

A.2 Der Kopf, 32 Byte

Offset	Inhalt
+0000…0003	Kennung, vier Zeichen: `CMUS`
+0004	Formatversion, zurzeit 1
+0005…0020	Songname, 16 Zeichen, rechts mit Leerzeichen aufgefüllt
+0021	Tempo: Frames je Patternzeile, 1–255
+0022	Anzahl Patterns AP, 1–21
+0023	Sequenzlänge SL, 1–64
+0024	Loop-Position, 0…SL−1
+0025	Anzahl benutzter Instrumente, 1–16 (nur Anzeigewert; der Block ist immer 128 Byte groß)
+0026	Filter-Cutoff, oberes Byte → $D416
+0027	Resonanz (Bit 4–7) und Filtertyp (Bit 0 tief, 1 band, 2 hoch)
+0028	Grundlautstärke 0–15 → untere 4 Bit von $D418
+0029	Flags: Bit 0 Stimme 3 ausleihbar, Bit 1 Schleife, Bit 2 Szenenmusik, Bit 3–7 Reserve
+0030…0031	Reserve, 0

Die Kennung ist das einzige verlässliche Erkennungsmerkmal. Auf Diskette ist CMUS schlicht „PRG", genau wie STKA, CCHR, CITM und alles andere im Projekt — wer den Typ prüfen will, muss die ersten vier Datenbytes lesen.

A.3 Sequenz, 64 Byte ab +0032

Ein Byte je Position: die Nummer des Patterns, das dort gespielt wird. Nur die ersten SL Bytes sind gültig, der Rest ist 0.

> **Beispiel:** SL=8, Loop=2, Sequenz = 0, 1, 2, 3, 2, 3, 2, 4
> Intro 0 und 1 einmal, danach ewige Wiederholung ab Position 2.

A.4 Instrumente, 8 Byte je Stück ab +0096

Instrument *n* liegt bei +0096 + *n* × 8.

Byte	Inhalt
+0	Attack (Bit 4–7) und Decay (Bit 0–3)
+1	Sustain (Bit 4–7) und Release (Bit 0–3)
+2	Wellenform — das Steuerregister *ohne* Gate-Bit: $10 Dreieck, $20 Sägezahn, $40 Puls, $80 Rauschen. Das Gate setzt der Player selbst.
+3	Pulsweite, unteres Byte
+4	Pulsweite, oberes Byte (nur Bit 0–3)
+5	Pulsweiten-Sweep je Frame, Zweierkomplement
+6	Vibrato: Bit 0–3 Tiefe, Bit 4–7 Geschwindigkeit
+7	Bit 0 Ringmod, Bit 1 Sync, Bit 2 Filter, Bit 3 Hard-Restart **AUS**, Bit 4–7 Slide-Schrittweite

> **Bit 3 ist invertiert.** Der Editor zeigt HARD-RESTART als AN oder AUS an, im Byte steckt aber das Gegenteil: Bit 3 gesetzt heißt Hard-Restart *aus*. Der Editor dreht das beim Anzeigen und beim Schreiben um — wer das Byte von Hand anfasst, muss daran denken.

A.5 Patterns, 192 Byte je Stück ab +0224

Pattern *n* beginnt bei +0224 + *n* × 192. Ein Pattern hat 32 Zeilen zu je drei Stimmen zu je zwei Byte. Die Zellen liegen **zeilenweise**, nicht stimmenweise — das ist die für den Player günstige Reihenfolge, er liest eine Zeile als sechs aufeinanderfolgende Bytes.

> **Offset einer Zelle** = 224 + *n* × 192 + Zeile × 6 + (Stimme − 1) × 2

Byte 0 — Note:

Wert	Bedeutung
0	leer, keine Änderung — die laufende Note klingt weiter
1…95	Note. 1 = C-0, 13 = C-1, … C-4 ist Note 49 (das eingestrichene C).
96	**UNBRAUCHBAR** siehe Anhang C
97	Note aus — Gate fällt, Release beginnt
98…255	Reserve, vom Player wie 0 behandelt

Byte 1 — Instrument und Effekt: Bit 4–7 Instrumentnummer 0–15, Bit 0–3 Effektnummer 0–15.

A.6 Was das Format bewusst nicht kann

- Keine getrennte Sequenz je Stimme — eine Patternnummer gilt für alle drei.
- Keine variable Patternlänge; ein vorzeitiger Sprung geht über Effekt 14.
- Keine freien Effektparameter, keine frei definierten Arpeggien, kein Tempowechsel mitten im Stück.
- Keine Wellenform- oder Frequenztabellen je Instrument, wie GoatTracker sie kennt.
- Keine Samples und kein 4-Bit-Digi über $D418 — das Register gehört dem Player allein.

Nachrüstbar ohne Formatbruch wären die Reservebytes +0030/+0031 und die Notenwerte 98–255. Alles andere — ein drittes Zellbyte, getrennte Sequenzen — wäre ein echter Bruch mit Konverterbedarf.

Anhang B: MUSKIT im eigenen Spiel benutzen

Der Player belegt 40960–43263 unter dem BASIC-ROM. BASIC erreicht ihn nie direkt, sondern über das Trampolin bei 53120–53247 im normalen RAM. Der Song liegt im Spiel bei **44800** und darf 4352 Byte belegen.

B.1 Der Parameterblock 53184–53199

Er liegt in sichtbarem RAM und ist deshalb aus BASIC mit PEEK und POKE erreichbar.

Adresse	Name	Bedeutung
53184	MUSTAT	0 = gestoppt, 1 = läuft, 2 = einmal durchgelaufen
53185	MUPOS	aktuelle Sequenzposition 0–63
53186	MUROW	aktuelle Patternzeile 0–31
53187	MUVOL	Lautstärke 0–15, jederzeit setzbar (Ein- und Ausblenden)
53188/89	MUSONGL/H	Songadresse — **vor** MUINIT setzen
53190	MULEND	Stimme 3 ausgeliehen: 0 nein, 1 ja
53191	MUERR	nach MUINIT: 0 ok, 1 keine CMUS-Kennung, 2 Version zu neu, 3 AP oder SL unplausibel
53192	MUFSTP	Schrittweite für das Ausblenden, 1–15
53194	SIDWHAT	gehört dem Sound-Editor

B.2 Die Sprungtabelle 53200–53211

SYS	Name	Wirkung
53200	MUINIT	Song einlesen und vorbereiten. Prüft die Kennung, übernimmt Tempo, Filter und Lautstärke, hängt den Interrupt ein. Spielt noch nicht. Ergebnis in MUERR.
53203	MUSTART	Wiedergabe ab der in MUPOS stehenden Position starten
53206	MUSTOP	Wiedergabe sofort anhalten. Gibt SIDOWN frei (auf 0), aber NUR wenn dort noch 1 steht — gehört die Sperre inzwischen den Geräuschen, rührt MUSTOP sie nicht an. Der IRQ bleibt eingehängt.
53120	MuHush	**Notausschalter.** Bringt den SID sofort zum Schweigen und gibt die Sperre frei — auch aus fremdem Code.

B.3 Die SID-Sperre SIDOWN

Musik und Geräusche teilen sich denselben Chip und laufen **nie gleichzeitig**. Statt Stimmen und Register aufzuteilen, sperren sie sich gegenseitig über ein einziges Byte bei **53143**:

Wert	Bedeutung
0	frei
1	Musik (MUSKIT)
2	Geräusche (Sound-Editor)

Die Regel gilt für beide Seiten: vor dem Starten prüfen — steht 0 oder der eigene Wert drin, darf man; steht der andere Wert drin, lässt man es und spielt einfach nicht. Beim Starten den eigenen Wert eintragen, beim Stoppen wieder auf 0 setzen, aber nur, wenn der eigene Wert drinsteht.

> **Reihenfolge beim Spielstart:** SNINIT (Sound) muss vor MUINIT laufen, weil SNINIT die Sperre bedingungslos auf 0 setzt. MUINIT selbst lässt eine 2 stehen — dann läuft gerade ein Geräusch, und das wird nicht gestohlen.

B.4 Musik vor jedem Diskettenzugriff anhalten

Vor OPEN, LOAD, SAVE und jedem Verzeichnisscan erst `SYS 53206`, danach wieder `SYS 53203`. Die seriellen KERNAL-Routinen sind zeitkritisch, und ein Player-Interrupt von 2000 Zyklen mitten in einer Bitfolge ist genau die Art Störung, die auf echter Hardware sporadisch zuschlägt und im Emulator nicht auffällt. Dieselbe Vorsichtsregel wie `POKE 53269,0` vor Laufwerkszugriffen, und aus demselben Grund.

Anhang C: Die Fallen

Diese Liste steht in keinem anderen Dokument. Sie sammelt, was beim Bauen tatsächlich schiefgegangen ist — jeder Punkt hat einmal Zeit gekostet.

C.1 Note 96 klingt falsch, sieht aber richtig aus

Note 96 wäre H-7. Sie bräuchte den SID-Frequenzwert 67276 und passt damit nicht in 16 Bit. Die Notentabelle begrenzt sie auf 65535 — das klingt rund 45 Cent zu tief, also hörbar verstimmt. Note 95 (A#7) ist mit 63504 die höchste saubere Note; alle 95 darunter sind exakt.

Der Editor sperrt die 96 an **zwei** Stellen: bei der Eingabe (mit der Meldung `ZU HOCH - OKTAVE MIT - SENKEN`) und bei der Anzeige. Kommt eine 96 aus einer fremden Datei ins Gitter, erscheint sie als `???` statt als H-7.

> **Warum die zweite Sperre nötig war:** ohne sie sähe der Song richtig aus und klänge falsch. Das ist der unangenehmste Fehlertyp überhaupt — man hört ihn, aber man sieht ihn nirgends. Über die Tastatur ist die 96 gar nicht erst erreichbar.

C.2 Ein gelogener Kopf spielt fremden Speicher als Noten

Siehe Anhang A.1. Bis zum 15. August 2026 nahm der Editor eine Datei an, deren Kopf mehr Patterns angab, als die Datei enthielt. Der Player las dann hinter dem Dateiende weiter. Aufgefallen ist das erst durch eine eigens gebaute Grenzfall-Datei — im normalen Betrieb hätte es nur „irgendwie schräg" geklungen.

C.3 Pulsweite 0 oder 4095 macht stumm

Wellenform PULS mit Pulsweite 0 oder 4095 ergibt keine Schwingung, sondern Stille. Häufigster Anfängerfehler am SID. Der Editor warnt in der Statuszeile, verbietet es aber nicht — es gibt Fälle, in denen ein Sweep von dort losläuft.

C.4 Ringmodulation und Sync brauchen einen Nachbarn

Stimme 1 greift auf Stimme 3 zurück, Stimme 2 auf 1, Stimme 3 auf 2. Schweigt die Nachbarstimme, passiert nichts Hörbares. Ringmodulation wirkt zudem nur bei Dreieckwelle.

C.5 Hard-Restart frisst Tempo 1 auf

Siehe Kapitel 4.3. Bei Tempo 1 bleibt von der Note nichts als der Hard-Restart.

C.6 Slides ohne Begrenzung springen an den anderen Anschlag

Siehe Kapitel 9. Gemessen am 14. August 2026: ein Abwärts-Slide landete nach gut zwei Sekunden bei 3800 Hz statt leise auszulaufen.

C.7 Patternzahl verkleinern kann die Sequenz verbiegen

Der Editor zieht Sequenzeinträge, die ins Leere zeigen würden, auf die letzte gültige Patternnummer

herunter. Ohne das spielte der Player dort stillschweigend Pattern 0 — der Song klänge anders, als er aussieht.

C.8 RUN/STOP + RESTORE hinterlässt einen Dauerton

Der Vorgang setzt die Interruptvektoren auf die KERNAL-Vorgaben zurück. Der Musik-Interrupt ist damit ausgehängt, der SID klingt aber mit dem zuletzt gesetzten Ton *weiter*. Ohne Gegenmaßnahme bleibt ein Dauerton stehen, den nur ein Reset beendet. Der Editor fängt das seit dem 14. August 2026 ab; in fremdem Code hilft `SYS 53120` (MuHush).

C.9 Das Instrument-Flagbyte ist an einer Stelle invertiert

Siehe Anhang A.4. Bit 3 gesetzt heißt Hard-Restart *aus*, nicht an.

> **Für Werkzeuge, die CMUS-Dateien erzeugen:** die drei Punkte, die man beim ersten Mal garantiert trifft, sind die Längengleichung (A.1), das invertierte Bit 3 (A.4) und die gesperrte Note 96 (C.1). Wer die drei beachtet, baut Dateien, die der Editor annimmt.

Anhang D: Wenn etwas schiefgeht

Jede Meldung, die der Editor rot und invers unten anzeigt (`GOSUB 910` im Quelltext), aus `musik-editor.bas` ausgelesen — nicht aus der Erinnerung rekonstruiert. Normale Statuszeilen (z. B. „GESPEICHERT: …“) stehen nicht hier, nur echte Fehlerfälle.

Meldung	Bedeutung	Nächster Schritt
`ZU HOCH - OKTAVE MIT - SENKEN`	Eingegebene Note läge bei 96 (H-7) — der SID-Frequenzwert passt nicht mehr in 16 Bit, siehe C.1	Oktave mit **-** senken, Note erneut eingeben
`MEHR ALS 21 PATTERNS GEHEN NICHT`	Patternzahl ist bereits beim Höchstwert 21	Vorhandenes Pattern wiederverwenden oder Song kürzen
`ERST STOPPEN, DANN AENDERN`	Änderung während laufender Wiedergabe verlangt	**F3** stoppt die Wiedergabe, dann erneut versuchen
`SID IST BELEGT - KEINE MUSIK` / `SID IST BELEGT`	Die SID-Besitzsperre SIDOWN (53143, siehe Anhang B.3) gehört gerade der Musik bzw. den Geräuschen eines anderen Programms	Warten, bis der andere Besitzer freigibt; im Notfall `SYS 53120` (MuHush)
`ERST DIE MUSIK STOPPEN`	Änderung im Instrument-Editor verlangt, während ein Song läuft (MUSTAT ≠ 0)	**F3** stoppt die Wiedergabe
`ZWISCHENABLAGE IST LEER` / `KEIN INSTRUMENT KOPIERT`	SHIFT-V (einfügen) ohne vorheriges SHIFT-C (kopieren)	Erst mit SHIFT-C kopieren
`KEINE DATEIEN GEFUNDEN`	Die Diskette im Laufwerk enthält keine ladbaren Dateien	Andere Diskette einlegen
`ZU GROSS FUER EINEN CMUS-SONG`	Song überschreitet beim Speichern den zulässigen Umfang (mehr als 17 Blöcke)	Song kürzen (weniger Patterns oder kürzere Sequenz)
`DATEI PASST NICHT ZUM KOPF`	Dateilänge stimmt nicht mit `224 + AP×192` überein — der Kopf nennt eine falsche Patternzahl (Prüfung seit 15.08.2026, siehe Anhang A.1)	Datei ist beschädigt oder von einem fehlerhaften Werkzeug erzeugt — nicht verwenden
`ABGELEHNT (MUERR n) - NEU ANGELEGT`	Geladene Datei ist kein gültiger CMUS-Song. n=1 keine CMUS-Kennung, n=2 Version zu neu, n=3 Patternzahl/Sequenzlänge unplausibel (siehe Anhang B.1). Der Editor hat sofort einen leeren Song angelegt, nichts geht verloren	Richtige Datei auswählen
`LADEFEHLER n` / `SPEICHERFEHLER n`	n ist der rohe KERNAL-Diskettenstatus (Register ST) — Tabelle unten	siehe Tabelle unten
`DISK:` <Text>	Rohe Meldung aus dem Fehlerkanal des Laufwerks (Kanal 15)	Text lesen — meist Diskette voll oder schreibgeschützt
`LAUFWERK n ANTWORTET NICHT`	Gerätenummer n meldet sich nicht auf dem seriellen Bus	Laufwerk einschalten, Kabel prüfen, richtige Gerätenummer einstellen

D.1 Die Zahl hinter LADEFEHLER/SPEICHERFEHLER

Diese Codes sind der Standard-KERNAL-Diskettenstatus (1541/1571) — **hier angelesen aus der allgemeinen CBM-DOS-Referenz, nicht am eigenen Gerät jeden Fehler einzeln nachgestellt** (Diskette absichtlich schreibschützen o. ä. wurde für dieses Kapitel nicht gemacht). Nur die in diesem Editor tatsächlich erreichbaren Fälle:

Code	Bedeutung	Nächster Schritt
62	DATEI NICHT GEFUNDEN	Dateiname prüfen, richtige Diskette einlegen
63	DATEI VORHANDEN	Beim Speichern: anderen Namen wählen oder alte Datei erst löschen
64	DATEITYP FALSCH	Datei ist kein PRG oder falsch adressiert
21, 23, 27	LESEFEHLER (verschiedene Ursachen am Medium)	Diskette/Laufwerk prüfen, erneut versuchen
25	SCHREIBFEHLER	Diskette prüfen, erneut versuchen
26	SCHREIBSCHUTZ AKTIV	Schreibschutzkerbe der Diskette schließen
72	DISKETTE VOLL	Platz schaffen oder andere Diskette einlegen
73	DOS-VERSION / LAUFWERK GERADE EINGESCHALTET	Meldet sich normalerweise nur einmalig direkt nach dem Einschalten — kein echter Fehler
74	LAUFWERK NICHT BEREIT	Diskette einlegen, Laufwerkstür schließen