

KLEINCOMPUTER



KC 85

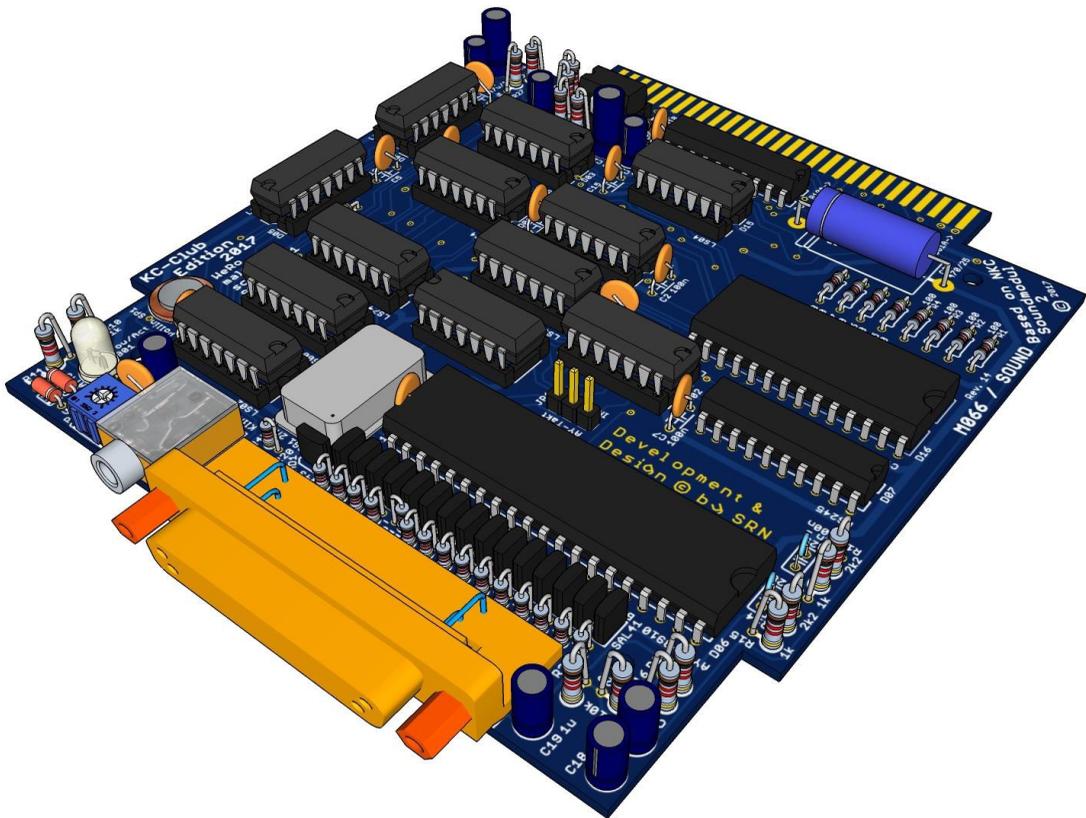
M066

KLANG

Neuentwicklung des Moduls

M066 KLANG

für den KC85/2-5



KLEINCOMPUTER

KC 85

Beschreibung zu M066 KLANG



veb mikroelektronik
'wilhelm pieck'
mühlhausen

KC-CLUB

Rolf Weidlich / René Nitzsche
Sven Haubold / Mario Leubner

Herausgeber: René Nitzsche im Auftrag des KC-Clubs

Autoren: Rolf Weidlich, René Nitzsche

Gestaltung und Layout: René Nitzsche

Fotos und Grafiken: Rolf Weidlich, René Nitzsche

Der Vertrieb dieser Druckschrift erfolgt ausschließlich durch den Herausgeber.

Ohne Genehmigung des Herausgebers ist es nicht gestattet, das Buch oder Teile daraus nachzudrucken oder zu vervielfältigen.

Hinweise, die zur Verbesserung dieser Dokumentation führen, werden gern entgegengenommen.

Redaktionsschluß: Dezember 2017

Vielen Dank für das Korrekturlesen an Silvia Nitzsche. Des weiteren wird dafür Rolf Weidlich, Sven Haubold und Mario Leubner gedankt.

SRN 2017 ©

Inhaltsverzeichnis

| | | |
|---------|---|----|
| 1 | Vorwort..... | 6 |
| 1.1 | Hinweise zu dieser Dokumentation | 6 |
| 1.2 | Danksagungen..... | 7 |
| 2 | Einleitung | 8 |
| 2.1 | Kurzübersicht | 8 |
| 2.2 | Historie | 8 |
| 3 | Modulhandhabung | 10 |
| 3.1 | Stecken und Entfernen des Moduls..... | 10 |
| 3.2 | Modulkennung und Betriebszustände | 11 |
| 3.2.1 | Modulkennung | 11 |
| 3.2.2 | Betriebszustände | 12 |
| 3.2.3 | Steuerbytedefinition | 13 |
| 4 | Technische Beschreibung | 14 |
| 4.1 | Hardware | 14 |
| 4.2 | Systemkonfiguration | 15 |
| 4.2.1 | Systemadressen | 15 |
| 4.2.2 | Beispiel | 16 |
| 4.3 | Tonerzeugung mit dem PSG | 16 |
| 4.3.1 | Wirkungsprinzip | 16 |
| 4.3.2 | Tonhöhe | 17 |
| 4.3.3 | Mischer | 19 |
| 4.3.4 | Lautstärke | 19 |
| 4.3.5 | Hüllkurve | 19 |
| 4.3.6 | AY-Musik..... | 21 |
| 4.3.6.1 | Umsetzung durch das M066 | 22 |
| 4.3.6.2 | Player für STC-Dateien | 22 |
| 4.4 | Programmierung des CTC..... | 22 |
| 4.5 | Verwendung der E/A-Anschlüsse | 23 |
| 4.6 | Benutzung des Taktausganges | 25 |

| | | |
|---------|--|----|
| 5 | Entwicklung, Aufbau und Inbetriebnahme | 27 |
| 5.1 | Modulentwicklung | 27 |
| 5.2 | Aufbauhinweise | 30 |
| 5.3 | Inbetriebnahme und Benutzung des M066 | 33 |
| 5.3.1 | Software für das M066 | 33 |
| 5.3.1.1 | Einzeltonprogrammierung | 33 |
| 5.3.1.2 | AY-3-8910-Sounddemo | 35 |
| 5.3.1.3 | Selbstabspielende Musik | 37 |
| 5.3.2 | Universaler AY-Player | 39 |
| 5.3.2.1 | Include-Datei für AY-Player in Programmen | 39 |
| 5.3.2.2 | Verwendung des AY-Players unter KC-BASIC | 41 |
| 5.3.2.3 | Verwendung des AY-Players unter JKCEMU-Basic | 43 |
| 6 | Anwendungsfall / Musikerzeugung | 47 |
| 7 | Ausblick auf den KC-Tracker für CAOS 4.7 | 48 |
| 8 | Anlage A – Schalt- und Belegungspläne | 49 |
| 8.1 | Belegungsplan – M066 | 49 |
| 8.2 | Schaltplan – M066 | 50 |
| 8.3 | Schaltplan – NKC Soundmodul 2 | 51 |
| 9 | Anlage B – Bestückungsliste | 52 |
| 10 | Anlage C – Anwendersoftware | 54 |
| 10.1 | Softwarepakete | 54 |
| 10.2 | Hinweis zu den verwendeten Arbeitszellen | 55 |
| 11 | Anlage D – Musiktheorie | 56 |
| 11.1 | Übersicht der PSG-Register | 56 |
| 11.2 | Registerwerte für einige Notenfrequenzen | 57 |
| 12 | Anlage E – Soundtracker-Dateiformat STC | 61 |
| 12.1 | Eckdaten und Grenzen des STC-Formats | 61 |
| 12.2 | Begriffe | 61 |
| 12.3 | Dateiaufbau | 62 |
| 12.3.1 | Samples | 62 |
| 12.3.2 | Ornamente | 63 |
| 12.3.3 | Pattern | 64 |
| 12.3.4 | Beispiel "JINGLE.STC" | 65 |

| | | |
|------|---|----|
| 13 | Anlage F – Quellenangaben und Links | 67 |
| 13.1 | Quellen..... | 67 |
| 13.2 | Links..... | 67 |
| 14 | Anlage G – KC85/x-Systeme im Test..... | 69 |
| 14.1 | KC85/2 (CAOS 2.2, 16kB Grund-RAM)..... | 69 |
| 14.2 | KC85/3 (CAOS 3.1, 16kB Grund-RAM)..... | 69 |
| 14.3 | KC85/4 (CAOS 4.2) | 69 |
| 14.4 | KC85/5 (CAOS 4.7) | 69 |
| 14.5 | M066 im KC85/4-Turm (CAOS 4.2)..... | 70 |

1 Vorwort

Musik auf bzw. mit dem KC85/2-5? Eine Tonausgabe gab es schon, allerdings in bescheidenem Rahmen (zweikanalig, per PIO bzw. CTC erzeugt). Zu spielende Musiknoten waren nicht gerade treffsicher in den Frequenzen.

Mancher Besitzer eines KC85 wird vielleicht etwas neidisch auf Gerätschaften wie den ZX spectrum geschaut haben. Die Industrie hatte für den Spielcomputermarkt extra einen Chip konstruiert, mit dem sich zwar keine naturalistischen, aber dennoch neuartige Töne und Lieder erzeugen ließen - einen programmierbaren Klanggenerator (englische Abk. PSG). Dieser PSG wurde 1978 von der Firma General Instrument entwickelt.

Mit der Neuentwicklung eines Moduls, des M066 KLANG, kann nun auch die KC85-Gemeinde diesen Klang genießen.

Hinzu kommt ein umfangreiches Softwarepaket, mit dem auch Nichtmusiker in die Lage versetzt werden, eigene Musikstücke zu entwerfen.

1.1 Hinweise zu dieser Dokumentation

Da die bislang für das M066 erstellte Software nicht ohne dieses Modul funktioniert, wird sie zusammen mit dem Modul vorgestellt und ist als Paket neben dieser Dokumentation erhältlich.

Hexadezimalzahlen werden durch drei verschiedene Schreibweisen dargestellt:

- Mit einem "h" am Ende und einer führenden "0", wenn die Zahl größer als "9Fh" ist.
- Mit einem führenden "#" in den gezeigten Programmcodebeispielen, da diese für den Assembler "ASide" geschrieben sind.
- Eine dritte Schreibweise verwendet weder das "h" noch das "#"-Zeichen und zwar dann, wenn es sich um Bildschirmausgaben des KC85 handelt.

Es werden nur die Schriftarten "Arial" und "Courier New" verwendet:

- `Nichtproportionalschrift` wird für Programmcode und Rechnerausgaben benutzt.
- **Nichtproportionalschrift fett** dient für Hervorhebungen im Programmcode.
- *Kursivschrift* kennzeichnet Zitate.

Die Programmcodebeispiele in Assemblersprache entsprechen der ASide-Assembler-Syntax. Alle anderen Beispiele verwenden entweder die Basic-Syntax des KC85-Interpreters oder des Compilers des Emulators JKCEMU. Sämtliche Beispiele stammen unverändert von Rolf Weidlich.

Die Bildschirmfotos entstanden größtenteils mit dem Emulator JKCEMU¹.

Querverweise in diesem Handbuch sind ebenso wie Internetlinks in blauer Schriftfarbe und unterstrichen dargestellt.

Es findet die Rechtschreibung vor der Reform von 1996 Anwendung.

Es wird keine Haftung für die Richtigkeit, Vollständigkeit und Aktualität der wiedergegebenen Informationen übernommen.

1.2 Danksagungen

Besonderer Dank geht an Sven Haubold, Mario Leubner und Rolf Weidlich.

Mario Leubner war durch seine gewohnt geduldige Art in der Lage, das Wissen zu vermitteln, welches notwendig ist, um für ein Modul für den KC85/2-5 eine Adreßdekodierlogik zu entwickeln. Er half darüber hinaus mit vielen nützlichen Hinweisen.

Sven Haubold studierte das Datenblatt des PSG (AY-3-8910) gründlich und simulierte die Hauptbestandteile der Schaltung mit einem Hardwaresimulationsprogramm. Weiterhin entwarf er eine Schaltung zur Spannungsverstärkung, welche für Teile des Moduls unbedingt benötigt wird. Diese Schaltung ging in veränderter Form in das Modul ein.

Rolf Weidlich machte sich mit großem Fleiß und Eifer daran, die letztendlich für die sinnvolle Verwendung des M066 benötigte Software zu entwickeln. Während der Inbetriebnahme des Prototyps des Moduls half er tatkräftig bei der Fehlersuche. Er gab immer wieder hilfreiche Ratschläge und machte viele Änderungs- und Verbesserungsvorschläge.

Ein spezieller Dank geht an Ralf Däubner. Dieser beschloß Ende des Jahres 2016 "aus heiterem Himmel heraus" Kontakt aufzunehmen, um die Entwicklung dieses Moduls zu initiieren.

Maik Trompeter wird an dieser Stelle ausdrücklich für seine moralische Unterstützung gedankt und nicht zuletzt für die Anfertigung der Modulblende.

¹ Der Emulator JKCEMU ist eine Entwicklung von Jens Müller.

2 Einleitung

Das M066 ist etwas ganz Neues für den KC85/2-5, aber letztendlich ein "alter Hut". Es handelt sich dabei um ein Musikmodul oder auch Klangmodul. Entwickelt wurde es von einer Gemeinschaft im Zeitraum Januar bis Oktober 2017.

2.1 Kurzübersicht

Die Basis des Moduls bildet ein sogenannter "programmable sound generator", ein PSG z.B. des Typs AY-3-8910. Die Grundschialtung basiert auf dem Soundmodul 2 für den NDR-Kleincomputer (NKC).

Im Laufe des Projektes wurde die Schaltung sehr stark modifiziert und erweitert und an die Bedürfnisse des KC85 angepaßt.

Mittels des M066 ist es möglich, Klänge oder Musikstücke in der Qualität abzuspielen, wie sie vom Atari oder anderen Computern her bekannt ist, welche den gleichen PSG einsetzen. Dazu muß das Modul lediglich an eine Stereoanlage mit integriertem Verstärker oder an Aktivboxen angeschlossen werden. Aber woher kommt die Musik?

Ausschlaggebend für den Erfolg des Projektes dürfte wohl die Beteiligung eines Musikbegeisterten sein, welcher als Letzter zum Projekt stieß. Bis dahin hatte er kaum oder keine Erfahrung mit der Entwicklung von Musik. Denn was ist eine Hardware wert, wenn es keine Software gibt, um diese zu nutzen. Mit der entwickelten Software ist jeder Anwender in der Lage, fertige Musikstücke über das M066 abzuspielen und selbst zu entwerfen.

Neben der Tonerzeugung ist ein eigener CTC (Zähler-Zeitgeber-Schaltkreis) Hauptmerkmal des Moduls. Mit diesem besteht die Möglichkeit, die Musik interruptgesteuert laufen zu lassen, wodurch auch die Entwicklung von Spielen möglich ist. Des weiteren sind zwei kleine, nette Spielereien vorhanden: Eine blaue LED, die im Takt der Musik flackert und die ständig leuchtende rote LED damit überdeckt und ein Mikrolautsprecher, dessen Lautstärke über einen Regler eingestellt werden kann. Dieser ist bei der Benutzung der Klinkenbuchse inaktiv (stumm).

Zur Bedienung des Moduls wird auf das Kapitel ["5.3 Inbetriebnahme und Benutzung des M066"](#) verwiesen. Die technischen Details des M066 sind im Kapitel ["4 Technische Beschreibung"](#) zu finden.

2.2 Historie

Faszinierend ist es, daß schon vor mehr als 22 Jahren, beim ersten KC-Club-Treffen, vom 07.-09.04.1995 in Gusow, ein zukünftiges Soundmodul vorgestellt wurde. Nachfolgend ein gekürztes Zitat aus einem damaligen Teilbericht zum Treffen von Klaus Witzenhausen.

Zitat/Anfang

Für unseren KC bricht demnächst ein neues Zeitalter an, in dem es heißt: "Es werde Sound!".

In unserer Gruppe wird mittlerweile kräftig an einem Soundmodul gebastelt. Das Grundkonzept steht, hardwaremäßig sind wir auch schon am Aufbau und die Software soll auch nicht mehr das Problem numero uno sein. Das Grundkonzept sieht ungefähr so aus:

1. *Nutzung der Schaltkreise AY-3-8910 oder AY-3-8912. Das sind Soundgeneratoren, welche schon im KC-compact erprobt worden sind.*

Der AY-3-8912 umfaßt 11 Oktaven. Also bietet er sich buchstäblich für unser Modul an. Im ZX-81 Computerclub ist auch ein Soundmodul entwickelt worden, allerdings mit dem AY-3-8910.

Eigener ROM für die Treibersoftware und ein eigener RAM für die erstellten Meisterwerke der Tonkunst sollen weitere Bestandteile werden. Wozu ein eigener ROM und RAM? Da unsere Software immer speicherintensiver wird, wäre es eine Zumutung, vom knappen Speicher auch noch Platz für Soundtreiber und Musik zur Verfügung zu stellen.

Da aber auch gute Musik ein "Speicherfresser" ist, soll der RAM mindestens 256 kB haben. Dieser Speicher hat aber noch einen Vorteil - er kann als "normaler" Speicher genutzt werden, wenn er nicht für Musik benötigt wird.

2. *Folgende Eigenschaften stehen ebenfalls im Konzept:*
 - a) *Stereosound*
 - b) *Ausgang an vorhandene Stereoanlagen*
 - c) *Eingang von einer Stereoanlage*
 - d) *ein interner Verstärker für Kopfhörer und Lautsprecher*
 - e) *ein Mikrofonanschluß*

Das Ganze soll in ein Steckmodulgehäuse reinpassen. Es wird also ziemlich eng. Das Modul wird außerdem eine eigene Kennung und eine Modulnummer bekommen. Über einen Standard für zukünftige Modulnummern und Kennbytes müssen wir uns unbedingt einigen. Keinesfalls sollten wir unterhalb von M 200 anfangen!

Zitat/Ende

Der vollständige Text kann unter dem im Kapitel "[13.2 Links](#)" bereitgestellten Internetlink nachgelesen werden.

3 Modulhandhabung

3.1 Stecken und Entfernen des Moduls

Das M066 – KLANG kann prinzipiell in jedem Modulsteckplatz betrieben werden. Jedoch ist dabei die Modulpriorität in der gewählten Systemkonfiguration zu berücksichtigen. Es sind weiterhin die nachfolgenden Informationen und die unter Kapitel "[4.2 Systemkonfiguration](#)" aufgeführten Hinweise zu beachten.

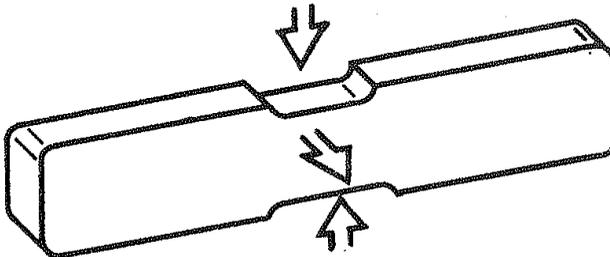
Die Modulprioritätskette muß immer geschlossen bleiben. Es sind also erst im Grundgerät der Steckplatz 8 (rechts), danach der Steckplatz C (links) und anschließend weitere Steckplätze von Erweiterungsaufsätzen in der vorgegebenen Reihenfolge zu belegen.

Achtung !

Das Stecken bzw. Entfernen des Moduls darf nur bei ausgeschaltetem Computer bzw. Aufsatz erfolgen!

Das Modul ist durch folgende Handgriffe zu stecken:

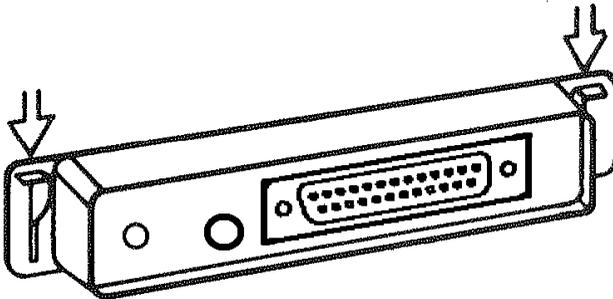
- a) Den Computer bzw. Aufsatz ausschalten.
- b) Die Kappe des Modulschachtes ist durch leichten Druck von Daumen und Zeigefinger auf die Griffflächen abzunehmen.



- c) Das Modul bis zum fühlbaren Einrasten einschieben (hervorstehender Rand des Moduls liegt unmittelbar an der Gerätewand an).
- d) Nun kann der Computer bzw. Aufsatz eingeschaltet werden.

Zum Entfernen des Moduls aus dem System sind folgende Schritte notwendig:

- a) Den Computer bzw. Aufsatz ausschalten.
- b) Den linken und rechten Zeigefinger unter den Modulkopf legen und mit den Daumen die seitlich am Modul befindlichen Hebel gleichzeitig nach unten drücken. Dabei rastet das Modul aus und wird etwa einen Zentimeter aus dem Gerät herausgeschoben. Nun das Modul aus dem Schacht nehmen.



- c) Die Kappe auf die Schachttöffnung stecken.

3.2 Modulkennung und Betriebszustände

3.2.1 Modulkennung

Jedes Modul erhält eine charakteristische Modulkennung, die durch das Kennbyte gekennzeichnet ist. Dieses Kennbyte widerspiegelt den Modultyp bzw. die innere Struktur des Moduls. Vom Nutzer kann hierauf softwaremäßig kein Einfluß genommen werden, da es sich um funktionsbedingte Festlegungen des Herstellers handelt. Der Prozessor ist in der Lage, das Kennbyte auch im inaktiven (nicht eingeschalteten) Zustand des Moduls zu lesen. Dadurch hat der Anwender jederzeit die Möglichkeit, sich in einem ausgebauten System einen Überblick über die verfügbaren Module zu verschaffen und in Abhängigkeit davon seine Entscheidung zu treffen.

Das M066 hat das Kennbyte 0DCh.

Aus diesem Grund ist es auch nicht sinnvoll, wenngleich es möglich ist, zwei M066 im System zu betreiben, da nur das höchstpriorisierte Modul verwendet werden kann.

3.2.3 Steuerbytedefinition

Eine Verarbeitung des Steuerbytes durch das Modul ist nicht möglich, da es ständig aktiviert ist und nicht abgeschaltet werden kann.

4 Technische Beschreibung

4.1 Hardware

Das Modul M066 ermöglicht mit dem darin enthaltenen "programmable sound generator" (PSG) eine dreikanalige Tonerzeugung (A, B, C). Sowohl "reine" Töne als auch Rauschen mit wählbarer Frequenz können in der Lautstärke variiert und/oder mit Lautstärke- und/oder Frequenzeffekten versehen und gemischt werden.

Die Ausgabe erfolgt in Stereoton (Kanal A = Rechts, Kanal C = Links, Kanal B wird mit jeweils etwa halbem Pegel auf Links und Rechts gemischt).

Zum Ausgabeanschluß steht an der Frontseite des Moduls eine 3,5mm-Stereo-Klinkenbuchse zur Verfügung, die mittels geeignetem Kabel mit einem Aktiv-Lautsprecher (bzw. NF-Verstärker) zu verbinden ist. Der maximale Ausgangspegel beträgt ca. $750 \text{ mV}_{\text{SS}}$ an $10 \text{ k}\Omega$.

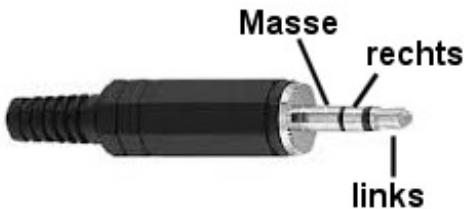


Abbildung 1: Belegung des Klinkensteckers

Wurde kein Kabel an der Klinkenbuchse angesteckt, so ist ein interner Miniaturlautsprecher aktiv (Lautstärke einstellbar), der alle drei Kanäle summiert ausgibt.

Eine blaue LED einer RGB-LED leuchtet im Rhythmus der Musik auf und dann zusammen mit der ständig leuchtenden roten LED (power-on).

Für eine Anpassung der Tonlagen (siehe Kapitel "[4.3.2 Tonhöhe](#)") besteht die Möglichkeit, den Takt des PSG mittels Steckbrücke vom Standardwert 2 MHz (Quarzoszillator) auf 1,77 MHz (KC-Takt) umzustellen (siehe dazu auch Kapitel "[11.2 Registerwerte für einige Notenfrequenzen](#)").

Der im M066 enthaltene CTC dient als Interruptquelle beim Abspielen von Musikstücken. Wird er nicht gerade für die Musikerzeugung benutzt, kann er auch anderweitig eingesetzt werden (siehe dazu "[4.6 Benutzung des Taktausganges](#)"). Der Ausgang von drei fest kaskadierten Kanälen ist mit auf den D-SUB-Steckverbinder gelegt.

Anwenderprogramme können das Vorhandensein eines M066 sowohl regulär über das Strukturbyte als auch über eine spezielle Kennungsabfrage prüfen.

Weiterhin kann man bei Bedarf unabhängig von der Tonerzeugung zwei bidirektionale 8-Bit-Anschlüsse nutzen, die an einem D-SUB-Steckverbinder zur Verfügung stehen.

Was mit dieser Hardware möglich ist, hängt vor allem von der zur Verfügung stehenden Software ab. Erste Experimente sind auch in BASIC möglich.

Prinzipielle Möglichkeiten dazu sind:

- Tonerzeugung per Tastatur ("Mini-Piano")
- Erzeugung einfacher oder komplexer Töne, Geräusche, Musiken
- Abspielen von AY-"Chiptunes" (Format je nach Player), siehe auch Kapitel "[4.3.6 AY-Musik](#)"
- Nutzung als Takt-/Tongenerator in der Elektronik

4.2 Systemkonfiguration

Das M066 KLANG ist in das Erweiterungskonzept des KC85/2 und seiner Nachfolgetypen voll eingebunden. Es können auch mehrere M066 im System vorhanden sein, was jedoch aufgrund der ständigen Aktivität eines M066 nicht sinnvoll ist.

Alle Module des Kleincomputers verfügen über eine Modulprioritätssteuerung, d.h. die Priorität fällt mit steigender Modulschachtnummer. Die Module sind in einer Kette aneinandergereiht, die durch die Signale MEI und MEO gebildet wird. Damit ist es möglich, alle in einem E/A-Bereich aktiv geschalteten gleichartigen Module in ihrer Priorität so zu steuern, daß nur das jeweils höchstpriorisierte Modul für den Prozessor verfügbar ist. Alle anderen aktiven Module bleiben für den Prozessorzugriff gesperrt. Beim M066 kann nur das höchstpriorisierte Modul verwendet werden.

4.2.1 Systemadressen

| | | |
|-------------------|------|-------------------------------------|
| Strukturbyte: | 0DCh | |
| E/A-Basisadresse: | 38h | => Auswahl Register, Lesen Register |
| | 39h | => Schreiben in Register |
| | 3Ah | => Lesen der Modulkennung (DC) |
| CTC-Basisadresse: | 3Ch | => Interrupterzeugung für Musiken |

4.2.2 Beispiel

| Funktion | Assembler | BASIC |
|---|-----------------------|-------------|
| Auswahl Register 7 | LD A,7 OUT (#38),A | OUT 56,7 |
| Lesen aktuelles Register | IN A,(#38) | X = INP(56) |
| Schreiben aktuelles Register mit Wert n | LD A,n OUT (#39),A | OUT 57,n |

Tabelle 1: Ansteuerbeispiel für den PSG

4.3 Tonerzeugung mit dem PSG

Im M066 wird in aller Regel als PSG ein AY-3-8910² zum Einsatz kommen. Datenblätter über die Funktionsweise und Bedienung des PSG sind im Internet verfügbar. Hier sollen nur einige Aspekte insoweit aufgeführt werden, als sie für das Grundverständnis und die Erstellung eigener Klänge nötig sind.

4.3.1 Wirkungsprinzip

Bei einer herkömmlichen digitalen Tonausgabe schaltet die CPU den Zustand einer Ausgabeleitung um und muß daher zur Tonerzeugung aktiv sein. Der PSG hat demgegenüber eine Besonderheit: Nachdem der gewünschte Ton durch Einschreiben von Daten in den PSG eingestellt wurde, arbeitet der PSG selbständig. Während der Tonausgabe kann die CPU etwas anderes tun oder theoretisch auch in den HALT gehen - der Ton erklingt weiter (sofern gewollt).

Um einen Ton mittels PSG zu erzeugen, ist das Beschreiben von bis zu 14 Registern nötig. Für bestimmte Funktionalitäten (z.B. Hochzählen eines Registerinhalts ohne zusätzliche Merkwelle) ist auch das Lesen von Registern möglich. In den Registern werden Tonhöhe, Lautstärke und deren Effekte sowie Informationen zum benutzten Kanal (A, B, C) abgebildet. Eine Gesamtübersicht der Register und deren Bedeutung sind in der Anlage "[11.1 Übersicht der PSG-Register](#)" dargestellt.

Um ungewollte Effekte zu vermeiden, kann es sinnvoll sein, zumindest zu Beginn immer alle 14 Register zu programmieren.

Komplexere Klänge (oder das Abspielen von Musik) erfordern jedoch wieder den Eingriff der CPU. Nach dem Initialisieren des Klangs wird nach

² Statt des AY-3-8910 kann auch der ansonsten pin- und registerkompatible Yamaha-Typ YM2149 benutzt werden. Er hat auch mit typ. 25 mA gegenüber dem AY-3-8910 mit typ. 45 mA eine geringere Stromaufnahme.

einer gewissen Zeitspanne ein bestimmter Wert (z.B. die Tonhöhe) neu programmiert. Beispiele dafür sind auch in den Datenblättern zu finden. Das erzeugte Tonsignal ist keine Sinuswelle und hat prinzipiell steile Flanken. Deshalb ist es oberwellenhaltig und klingt etwas "hart". Allerdings besitzt das Signal (im Gegensatz zu den herkömmlichen 1-Bit-Tonausgaben) nicht nur zwei Zustände, sondern es existieren je nach Lautstärke/Hüllkurvenform die verschiedensten Zwischenpegel.

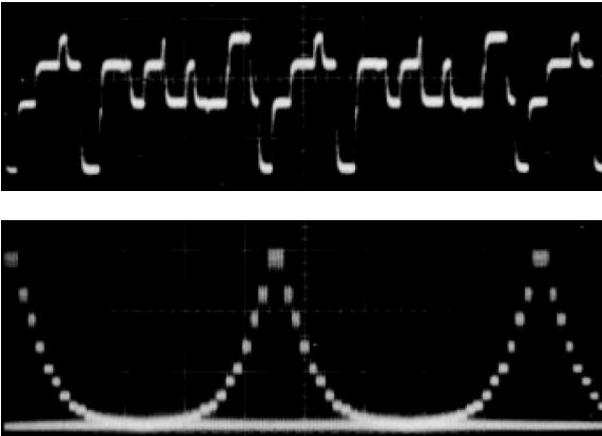


Abbildung 2: Kurvenform-Beispiele aus dem Datenblatt

4.3.2 Tonhöhe

Die Kanäle A, B und C lassen sich unabhängig voneinander über die Register 0...5 in der Tonhöhe programmieren. Für die Höhe stehen 12 Bit zur Verfügung, aufgeteilt in je zwei Register (4 Bit grob und 8 Bit fein). Je größer der Wert ist, desto tiefer erklingt der Ton. Der Wert 0 schaltet den Ton aus.

Um eine bestimmte Frequenz zu erzeugen, sind die benutzte Taktfrequenz und die PSG-interne Teilung durch 16 zu berücksichtigen. Daraus ergibt sich folgende Rechenvorschrift (dezimale Angaben) für die beiden Tonhöhenregister.

$$\text{TON-Frequenz} = (\text{Taktfrequenz} / 16) / \text{Teilerwert}$$

| Note | TON-Frequenz in Hz | | Teiler TP= INT(TAKT/ (16*TON)) | Grobwert INT (TP/256) | Feinwert TP MOD 256 |
|----------|--------------------|-------|--------------------------------------|-----------------------------|------------------------|
| | Soll | Ist | | | |
| c1 | 261,6 | 261,5 | 477 | 1 | 222 |
| maximal: | 125 000 | | 1 | 0 | 1 |
| minimal: | 30,5 | | 4095 | 15 | 255 |

Tabelle 2: Beispiel zur Rechenvorschrift für die Taktfrequenz von 2 MHz

Mit 12 Bit Auflösung lassen sich 4096 verschiedene Werte für die Tonhöhe angeben. Bei 2 MHz Taktfrequenz beträgt die niedrigste Ausgabefrequenz ca. 30 Hz und die höchste 125 kHz, wobei die Einstellgenauigkeit nach oben hin extrem abnimmt. Dabei werden die Notenfrequenzen zunehmend nicht mehr gut getroffen (Abweichungen siehe Kapitel "[11.2 Registerwerte für einige Notenfrequenzen](#)").

Der in den Publikationen oft angegebene Tonumfang von "8 Oktaven" sieht in der Praxis wie folgt aus. Er entspricht mit Ausnahme der nicht erreichbaren beiden tiefsten Töne dem Umfang eines normalen Klaviers.

$$H_1 (30.868 \text{ Hz}) \dots C^5 (4186.010 \text{ Hz})$$

Höhere Töne sind zwar möglich, jedoch wenig gebräuchlich und die Frequenzabweichungen vom Sollwert nehmen zu.

Die Taktfrequenz des PSG geht proportional in den Teilerwert ein, d.h. bei gleichem Teilerwert ist bei einer höheren Taktfrequenz der Ton auch höher, z.B.: Teilerwert = 477

Takt 2 MHz => f = 261,5 Hz, ca. Notenwert "c¹"
 1,77 MHz => f = 231,9 Hz, ca. Notenwert "ais"
 (2 Halbtöne tiefer)

Um eine Note eine Oktave höher zu spielen, ist der Teilerwert zu halbieren und umgekehrt zu verdoppeln, um diese eine Oktave niedriger erklingen zu lassen. Fertige Registerwerte für die Notenfrequenzen ("gleichstufige Stimmung") können dem Kapitel "[11.2 Registerwerte für einige Notenfrequenzen](#)" entnommen werden.

Das Register 6 ermöglicht es, anstatt eines Tones ein "Rauschen" auszugeben, was im Frequenzspektrum ("Tonhöhe" des Rauschens) mit 5 Bit variabel einstellbar ist. Der Zusammenhang von "Rauschtonhöhe" und Teilerangabe ist der gleiche wie bei der Tonhöhenberechnung.

$$\text{Rauschtonhöhe} = (\text{Taktfrequenz} / 16) / \text{Teilerwert}$$

4.3.3 Mischer

Das Register 7 ermöglicht die Festlegung, was auf welchem Kanal ausgegeben werden soll. Zu beachten ist, daß die Freigabe mit 0 erfolgt. Um alle Ausgaben zu sperren, ist das Register also mit 0FFh zu beschreiben. Das entspricht einer "Stummschaltung", die Generatoren laufen aber weiter, solange deren Frequenz- und Lautstärkewerte nicht auf 0 gesetzt werden.

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------------------------|-------|----------|-------|-------|-------|-------|-------|
| B | A | C | B | A | C | B | A |
| Freigabe E/A-Kanal für Eingabe | | Rauschen | | | Ton | | |

Tabelle 3: Register 7 (Mischer-Register) des PSG

So läßt sich z.B. mit dem Wert 0EEh auf Kanal B ein Rauschen und Kanal A ein Ton ausgeben. Darüber hinaus werden über den Mischer auch die beiden E/A-Kanäle (für die Eingabe) freigeschaltet bzw. gesperrt (siehe dazu auch Kapitel [4.5 Verwendung der E/A-Anschlüsse](#)).

4.3.4 Lautstärke

Die Kanäle A, B und C lassen sich unabhängig voneinander mit den Registern 8-10 in der Lautstärke programmieren. Mit den nutzbaren 4 Bit stehen 16 Lautstärkestufen (0-15) zur Verfügung, wobei der Wert 15 der Maximallautstärke entspricht.

Ist im Registerwert jedoch zusätzlich Bit 4 gesetzt, so wird die Einstellung der Lautstärke ignoriert und diese stattdessen über eine "Hüllkurve" gesteuert.

4.3.5 Hüllkurve

Wurde die Benutzung der Hüllkurve festgelegt, so kann damit die Lautstärke eines Tones mit einem Effekt versehen werden, der für eine festgelegte Dauer und/oder auch wiederholt eintritt.

Der Hüllkurven-Effekt beeinflusst ausschließlich die Lautstärke, nicht die Tonhöhe. Zu einem Zeitpunkt ist immer nur eine Hüllkurve möglich, die dann für einen oder mehrere Kanäle (je nach deren Lautstärkewert) gilt.

Die Definition der Hüllkurve erfolgt in den Registern 11-13. Vier Bits von Register 13 legen die Form fest.

4.3.6 AY-Musik

Will man nicht nur einzelne Töne, sondern eine komplette Musik ("Chiptune") hören bzw. ausgeben, so muß dem PSG eine Folge von Registerwerten übermittelt werden. Das läßt sich auch als Datei abbilden, wofür diverse Formate existieren.

Solche Dateien lassen sich z.B. mit den sogenannten "Soundtrackern" erzeugen und abspielen. Es werden "einfach" nur die Noten eingegeben sowie ggf. die Art des Effekts. Ein Beispiel dafür ist der "Sound Tracker 1.1" für den ZX Spectrum.

Die Bedienung erfordert, statt der genauen Kenntnis des PSG und seiner Register, "nur noch" musikalische Fähigkeiten. Dafür muß mit "Positionen", "Pattern", "Samples" und "Ornamenten" umgegangen werden.



Abbildung 3: Sound Tracker 1.1 des ZX Spectrum

Im Internet gibt es zahlreiche Musikveröffentlichungen in den verschiedensten Dateiformaten. Will man diese auf einem beliebigen Rechner abspielen oder gar bearbeiten, so muß man ein Programm haben, welches das entsprechende Format unterstützt.

Wie bei der Einzeltonerzeugung dargestellt, ist die Tonhöhe der Musik von der Taktfrequenz des PSG abhängig. Der Unterschied zwischen einem 2-MHz-Takt und dem KC-Systemtakt von 1,77 MHz beträgt ca. 2 Halbtöne.

4.3.6.1 Umsetzung durch das M066

Prinzipiell besteht eine AY-Musik aus einer Folge von Registerwerten, die fortlaufend (also im Gegensatz zum einzelnen Geräusch mit ständiger Aktivität der CPU) in den PSG geschrieben werden. Resultierend aus der damals benutzten Technik wurden diese Schreibvorgänge in einem Rhythmus von 1/50 Sekunde vorgenommen. Damit lässt sich ein fließender Klang erzeugen.

Als "Taktgeber" wird im M066 ein eigener CTC benutzt, welcher einen Interrupt dieser Frequenz erzeugt. So wird der Rechner durch die Musikausgabe nicht blockiert und das Gesamtsystem nur wenig belastet.

4.3.6.2 Player für STC-Dateien

Zur Demonstration der Funktionsfähigkeit wurde ein Player benutzt, der das STC-Format beherrscht. Derartige Dateien sind relativ klein (ca. 4 KB), im Gegensatz zu den "besser klingenden" (aber wesentlich größeren) MOD-Dateien für AMIGA & Co, mit Samples von Klängen echter Musikinstrumente. Andere Formate sind natürlich auch möglich, erfordern dann aber einen anderen Player.

Der weiter vorn schon erwähnte "Soundtracker" arbeitet mit nicht kompilierten Musiken (Dateiformat "*.ST1"). Es gibt dafür aber Zubehörprogramme, womit eine Umwandlung (ST1 ↔ STC) möglich ist. Beim STC-Format handelt es sich um eine kompilierte Form, d.h. die Register-Programmierwerte für den PSG werden nicht direkt sondern komprimiert abgelegt. Das spart natürlich Platz (kürzere Datei für die Musik), erfordert aber ein de-kompilieren durch den Player.

Der Player liest die komplett in den RAM geladene STC-Datei stückweise ein und ermittelt (de-kompiliert) den tatsächlichen Registerwert. Damit wird nun im o.a. Takt der PSG programmiert und der Ton beginnt zu erklingen.

Durch die wiederholte Abfolge von Programmierungen des PSG wird ein fließender Klang erzeugt. Das ist eigentlich schon alles.

4.4 Programmierung des CTC

Der M066-interne CTC wird für das Abspielen von AY-Musik benötigt. In der Player-Engine (siehe dazu Kapitel ["5.3.1 Software für das M066"](#)) und damit den Demo-Programmen und Musikstücken wird er so programmiert, daß er alle 20 ms (Frequenz 50 Hz) einen Interrupt erzeugt, wie nachfolgend erläutert:

- Kanal 0 wird am Eingang mit 500 kHz getriggert und arbeitet als Rückwärtszähler. Mit einem einzustellenden Teilerverhältnis von 80:1 liefert Kanal 0 am Ausgang eine Frequenz von 6.25 kHz.
- Diese wird durch den kaskadierten Kanal 1 nochmals um 125:1 geteilt, sodaß an seinem Ausgang 50 Hz anliegen. Für Kanal 0 ist kein Interrupt vorgesehen, jedoch für Kanal 1.
- Die Kanäle 2 und 3 werden für diesen Fall nicht benötigt.

Damit ergeben sich folgende dargestellte Programmierschritte.

| Kanal | Art | Wert | Bemerkung | Kode CTC EQU #3C |
|-------|------------------|------|--|---------------------------|
| 0 | Interrupt-vektor | #00 | Mit I-Register = 1 ergibt sich für Kanal 1 die Adresse #0102. Dort muß die Adresse der Routine stehen, die bei Interrupt durch CTC/K1 aufgerufen wird. | LD A,#00 OUT (CTC),A |
| 0 | Kanal-Steuerwort | #47 | Zählermodus, kein Interrupt, Zeitkonstante folgt | LD A,#47 OUT (CTC),A |
| | Zeitkonstante | #50 | Teilerverhältnis 80:1 500 kHz an K0 liefern 6,25 kHz an K1 | LD A,#50 OUT (CTC),A |
| 1 | Kanal-Steuerwort | #C7 | Zählermodus, Interrupt zugelassen, Zeitkonstante folgt | LD A,#C7 OUT (CTC+1),A |
| | Zeitkonstante | #7D | Teilerverhältnis 125:1 K1 liefert Interrupt (sowie 50 Hz an K2) | LD A,#7D OUT (CTC+1),A |

Tabelle 5: Programmierung des CTC für einen Interrupt

4.5 Verwendung der E/A-Anschlüsse

Die beiden 8-Bit-Anschlüsse des PSG werden für die Tonerzeugung nicht benötigt und können unabhängig davon programmiert und benutzt werden. Der Anschluß erfolgt mit einem 25-poligen D-SUB-Steckverbinder.

| | |
|-----------------------------------|--|
| M M M M M A1 A3 A5 A7 B1 B3 B5 B7 | Buchsenbelegung, Sicht von vorn |
| | Ax = Leitungen Anschluß A Bx = Leitungen Anschluß B M = Masse T = Taktausgang (von CTC) T/2 = wie T, aber f/2 und Tastverhältnis 1:1 |
| M M T T/2 A0 A2 A4 A6 B0 B2 B4 B6 | |

Tabelle 6: Belegung der D-SUB-Buchse an der Frontseite des M066

Es stehen 2 x 8 Bit zur Verfügung; eine Interruptverarbeitung und Handshake-Leitungen wie bei einer Z80-PIO gibt es nicht.

Die beiden E/A-Anschlüsse des PSG sind Registern zugeordnet. Diese Register sind wie die Klangregister vor dem Schreiben oder Lesen auszuwählen. Die Richtungswahl für die Anschlüsse erfolgt durch die Art der Freigabe im Mischer. Ist das dem Anschluß zugeordnete Bit Null, so wirkt der gesamte Anschluß als Eingang, andernfalls als Ausgang. Ein Aufteilen eines Anschlusses in unabhängige Richtungsleitungen ist daher nicht möglich.

Bei gleichzeitiger Benutzung von Klang und den E/A-Anschlüssen ist besonders auf diese Freigabebits im Mischer zu achten.

Mittels einfacher Ein- und Ausgabeanweisungen können Daten vom ausgewählten Anschluß gelesen oder auf den ausgewählten Anschluß geschrieben werden.

Die Pegel sind TTL-kompatibel. Als Ausgang können die Anschlußleitungen eine Standard TTL-Last (1.6 mA bei low-Pegel) bzw. 3 bis 4 LS-TTL-Eingänge treiben. Ein als Eingang benutzter Anschluß enthält interne pull-up-Widerstände; offene Leitungen werden damit als high-Pegel gelesen. Angaben zur Last eines Einganges liegen nicht vor.

Die Anschlußleitungen sind auf dem M066 mit einem Vorwiderstand und einer Schutzdiode versehen. Dennoch ist bei der Beschaltung mit externer Hardware Vorsicht geboten.

Nachfolgend wird ein Programmierbeispiel (in BASIC) dargestellt.

Anschluß A soll als Eingabeanschluß und Anschluß B als Ausgabeanschluß dienen:

```

100  REM PORTS VORBEREITEN
110  OUT 56,7      :REM REGISTER 7 AUSWÄHLEN (MISCHER)
120  X=INP (56)   :REM REGISTER 7 LESEN
130  X=X OR 128   :REM BIT 7 AUF 1 (PORT B IST AUSGANG)
140  X=X AND 191  :REM BIT 6 AUF 0 (PORT A IST EINGANG)
150  OUT 57,X     :REM REGISTER 7 SCHREIBEN, TON UNBEEINFLUSST
160  REM PORT A LESEN
170  OUT 56,14    :REM REGISTERAUSWAHL: 14 = PORT A
180  X=INP (56)   :REM REGISTER 14 = PORT A LESEN
190  REM PORT B SCHREIBEN
200  OUT 56,15    :REM REGISTERAUSWAHL: 15 = PORT B
210  OUT 57,X     :REM REGISTER 15 = PORT B MIT X BESCHREIBEN

```

Listing 1: Programmierbeispiel für die E/A-Anschlüsse des PSG

Nach der Registervorbereitung sind also pro E/A-Operation an einem Anschluß jeweils zwei Anweisungen nötig:

1. Auswahl des Anschlusses OUT 56,n
n=14 für Anschluß A
n=15 für Anschluß B
2. Ausgabe bzw. Eingabe OUT 57,x
X=INP(56)

4.6 Benutzung des Taktausganges

Am D-SUB-Steckverbinder Pin 23 kann bei Bedarf ein TTL-Rechtecksignal ausgegeben werden. Drei kaskadierte CTC-Kanäle ermöglichen eine variable Ausgangsfrequenz. Jeder Kanal kann dazu durch Wahl der entsprechenden Zeitkonstante ($ZK=0 \rightarrow 256$) auf ein Teilverhältnis von 1:1 bis 256:1 eingestellt werden.

Der Takt an diesem Ausgang besteht aus High-Impulsen mit der entsprechenden Wiederholfrequenz. Die Impulsbreite beträgt nur ca. 0,5 μ s und ist vom programmierten Teilverhältnis unabhängig.

Ein internes Flipflop erzeugt eine halbierte Frequenz mit dem Tastverhältnis 1:1, ausgegeben am D-SUB-Steckverbinder Pin 22.

Da die Hardware-Taktung des CTC-K0 mit 500 kHz erfolgt, können quartzgenaue Ausgangsfrequenzen am Ausgang T/2 mit einem Tastverhältnis von 1:1 im Bereich von

$$500 \text{ kHz} / [1-256] / [1-256] / [1-256] / 2 \Rightarrow 250 \text{ kHz bis ca. } 0.015 \text{ Hz}$$

erzeugt werden.

Läßt man einen Kanal als Zeitgeber arbeiten, so lassen sich auch Bruchteile der KC85-Taktfrequenz erzielen.

Die beiden Taktausgänge sind TTL-kompatibel und können wie folgt belastet werden:

- Pin 23 (T): CTC direkt \rightarrow 2 mA bei low
- Pin 22 (T/2): Ausgang eines LS74 \rightarrow 8 mA bei low

Vorsicht bei der Beschaltung mit externer Hardware, eine Schutzbeschaltung gibt es hier nicht.

Nachfolgend wird tabellarisch ein Programmierbeispiel gegeben, wie eine mit der Anode an Pin 22 des D-SUB-Steckverbinders und mit der Katode über einen 2.2 k Ω Vorwiderstand gegen GND angeschlossene Niedrigstrom-LED nun infolge des internen Flipflops im Sekundenrhythmus blinkt.

| Kanal | | Wert | Bemerkung | CTC: EQU #3C |
|-------|---------------|------|-------------------------|-----------------------------|
| 0 | Steuerwort | #47 | Zählermodus, ohne INT. | LD A, #47 OUT (CTC), A |
| 0 | Zeitkonstante | #50 | Teiler 80:1 => 6,25 kHz | LD A, #50 OUT (CTC), A |
| 1 | Steuerwort | #47 | Zählermodus, ohne INT. | LD A, #47 OUT (CTC+1), A |
| 1 | Zeitkonstante | #7D | Teiler 125:1 => 50 Hz | LD A, #7D OUT (CTC+1), A |
| 2 | Steuerwort | #47 | Zählermodus, ohne INT. | LD A, #47 OUT (CTC+2), A |
| 2 | Zeitkonstante | #19 | Teiler 25:1 => 2 Hz | LD A, #19 OUT (CTC+2), A |

Tabelle 7: Programmierbeispiel des CTC - Sekundenblinker

Um die Blink-Ausgabe zu stoppen genügt es beispielsweise, den Kanal 2 anzuhalten bzw. abzuschalten.

| | |
|--|---|
| LD A, 3 OUT (CTC+2), A | Zähler wird angehalten |
| ... | Ausgabe ist ausgeschaltet |
| LD A, #47 OUT (CTC+2), A LD A, #32 OUT (CTC+2), A | Eingabe einer neuen Zeitkonstante. So wird im o.a. Beispiel der Blinker wieder eingeschaltet, jetzt aber nur halb so schnell... |

Tabelle 8: Programmierbeispiel des CTC – Sekundenblinker / Fortsetzung

5 Entwicklung, Aufbau und Inbetriebnahme

Das Modul M066 – KLANG wurde im Zeitraum Dezember/2016 bis Oktober/2017 entwickelt. Während dieser Zeit flossen neue Gedanken und Ideen mit ein, so daß aus der ursprünglichen Intention, nur das Soundmodul 2 des NKC für den KC85/2-5 zu portieren, etwas sehr Innovatives entstanden ist. Zum Vergleich zwischen den beiden Modulen wird auf das Kapitel "[8 Anlage A – Schalt- und Belegungspläne](#)" verwiesen, welches auch den Schaltplan des Soundmoduls 2 beinhaltet.

5.1 Modulentwicklung

Wie es bei der Entwicklung eines Produktes üblich ist, wurde nicht auf "gut Glück" ein Schaltplan entworfen und ein Layout gezeichnet. Nach dem der Schaltplan einen Entwicklungsstand erreichte, welcher die Anfertigung eines Prototypen rechtfertigte, wurde ein Layout gezeichnet und der Prototyp hergestellt. Der Prototyp entspricht der Revision 14.

Während des Aufbaus und der Erprobung des Prototypen ergaben sich sowohl Änderungen am Schaltplan als auch am Layout.

Für den internen Lautsprecher und die Rhythmus-LED wird ein OPV zur Spannungsverstärkung eingesetzt. Nach einigen Tests wurden die optimalen Werte der Verstärkung ermittelt.

Ursprünglich war als Tongeber ein Piezo-Schallwandler vorgesehen. Da dieser jedoch frequenzabhängig ist, was die erreichbare Tonhöhe bzw. Lautstärke anbelangt, wurde ein Mikrolautsprecher eingesetzt. Es wurden verschiedene Typen getestet und wie es oft so ist, fiel die Wahl auf den Kleinsten (klein aber oho). Dadurch ergibt sich eine kleine Layoutänderung, da auch das Potentiometer zur Einstellung der Lautstärke dieses Lautsprechers seinen Platz im Inneren des Moduls findet (die Lautstärke wird in aller Regel nicht ständig geändert werden müssen).

Die Rhythmus-LED arbeitet zuverlässig und wie beabsichtigt. Die Steckbrücken, welche die Deaktivierung der LED ermöglichen, wurden entfernt.

Während der Tests wurde festgestellt, daß an den Ausgängen des OPV eine Störspannung (Schwingspannung) anliegt. Diese hat ihren Ursprung in der verrauschten 12-Volt-Grundspannung des KC85. Da diese Störspannung nicht erwünscht ist, wurden verschiedene Versuche unternommen, diese zu verringern. Schon der Einsatz des Mikrolautsprechers sorgt für eine Minderung, da er keine kapazitive Last in das System einbringt, so wie es der Piezo-Schallwandler tut. Letztendlich ergab sich durch den Test verschiedener OPV-IC eine zufriedenstellende

Reduzierung der Störspannung auf rund 80 mV_{SS}. Der nun verwendete OPV ist ein LM358. Auch das Einbringen eines zusätzlichen Kondensators über dem Lautstärkepotentiometer brachte eine Minderung der Störspannung.

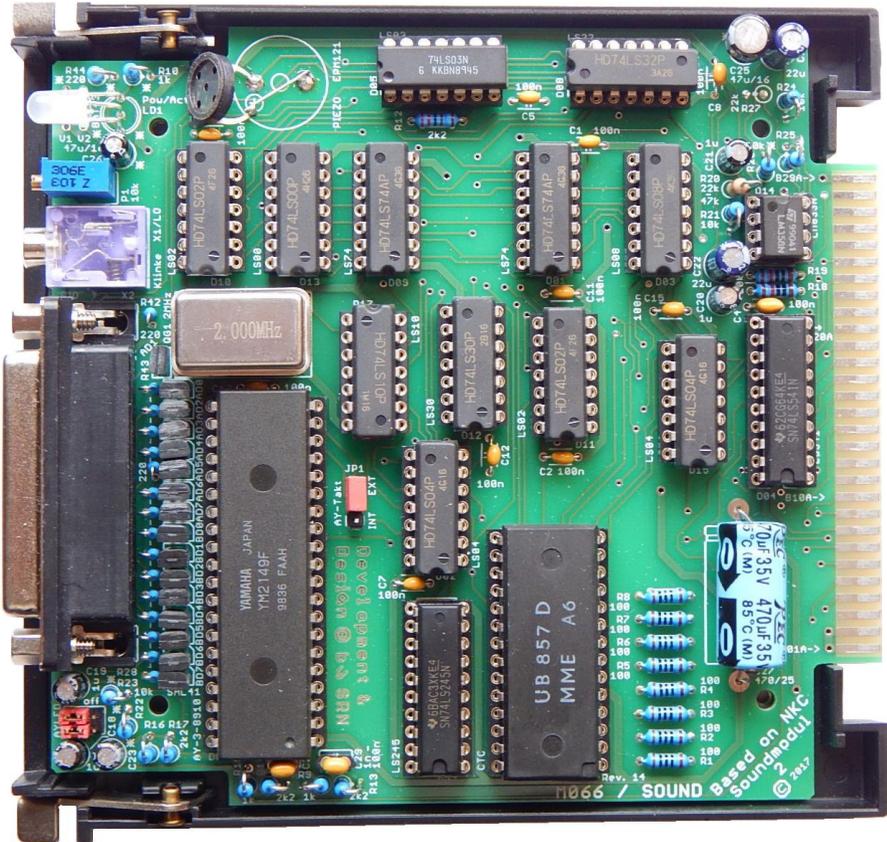


Abbildung 4: Prototyp des M066 – vollständig aufgebaut

Die Funktionstests umfaßten den gesamten Umfang des M066.

Es wurde die Tonausgabe über den internen Lautsprecher und über eine per Klinkensteckerkabel angeschlossene Stereoanlage getestet. Die Tonausgabe über die Stereoanlage ist von beeindruckender Qualität und Klarheit.

Des weiteren wurden zwei verschiedene Aktivboxen-Systeme getestet. Zum einen ein System von Trust (Xilo Compact 2.0 Speaker). Dieses System ist ausschließlich für die Kopfhörerausgänge von PC oder

Notebooks geeignet und ist aufgrund seiner geringen Empfindlichkeit für das M066 nicht zu empfehlen. Zum anderen ein System von Logitech (S-120 2.0 Speaker System). Dieses System ist wenig teurer als das von Trust und eine klare Empfehlung für das M066. Es bietet eine klare, klirrfreie und rauscharme Wiedergabe der vom M066 abgespielten Musikstücke.

Das Abspielen einiger Musikstücke beinhaltete die Verwendung des M066-eigenen CTC, so daß damit auch dieser Schaltkreis funktionsfähig getestet wurde.

Weiterhin wurden die beiden E/A-Anschlüsse des PSG getestet, sowohl zur Eingabe als auch zur Ausgabe. Es wurde eine einfache LED-Schaltung aufgebaut und der Anschluß A als Eingang und Anschluß B als Ausgang verwendet. Wird ein Bit des Anschlusses A auf low gesetzt, leuchtet die am korrespondierenden Bit des Anschlusses B befindliche LED auf.

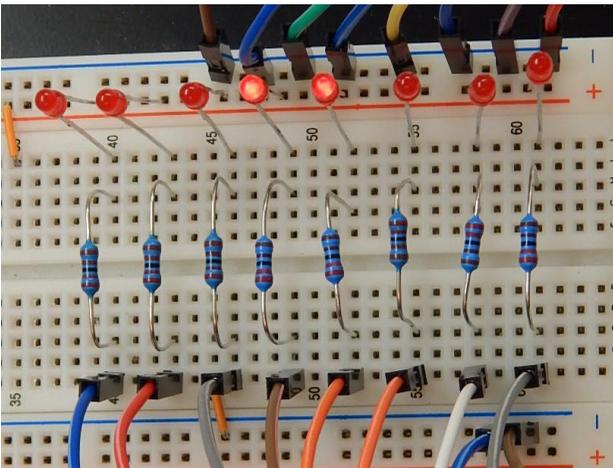


Abbildung 5: Test der E/A-Anschlüsse des PSG

Der kaskadierte Ausgang des CTC, welcher ebenfalls am D-SUB-Steckverbinder anliegt wurde, mit einer einfachen Blinkerschaltung auf Funktionalität überprüft.

Das Modul gliedert sich reibungslos und fehlerfrei in das Gesamtsystem des KC85 ein. Es wurden keine störenden Wechselwirkungen zwischen dem M066 und anderen Module beobachtet.

Um dies zu testen, wurde ein Programm gestartet, welches die Module M001 und M051 verwendet und interruptgesteuert arbeitet. Danach wurde

stehend) sind sorgfältig einzulöten, um keine Kurzschlüsse herzustellen. Dies gilt insbesondere für die Dioden. Bei diesen empfiehlt sich, immer erst ein Beinchen anzulöten, dann abzuschneiden und dann das nächste Beinchen anzulöten. Bei den Dioden ist weiterhin zu beachten, daß diese mit sanftem Druck in die Lötäugen hineingedrückt werden müssen, bis sie tief genug sitzen, um im Modulgehäuse Platz zu finden. Hier ist Fingerspitzengefühl gefragt.

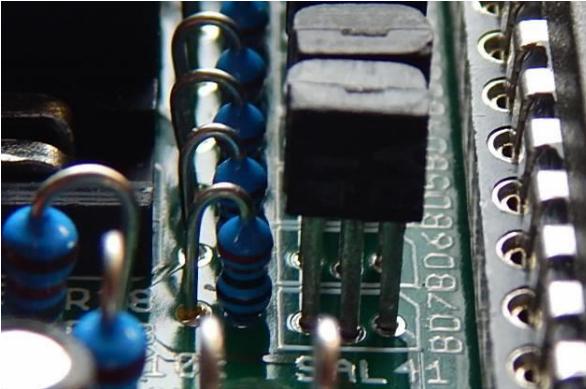


Abbildung 7: Doppeldioden SAL41

Der empfohlene interne Minilautsprecher sollte geneigt, in einem Winkel von ca. 45 Grad nach hinten, eingebaut werden. Damit wird die beste Resonanzwirkung erreicht.

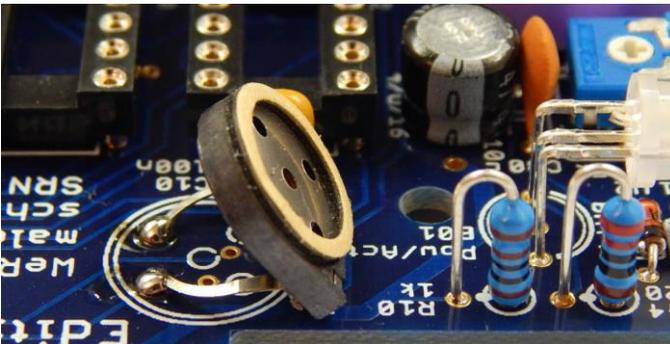


Abbildung 8: Einbau des Minilautsprechers

Die Anordnung der Lötstifte der Klinkenbuchse entspricht nicht ganz der der Lötäugen. Daher ist zunächst nur ein Beinchen anzulöten und dann die Buchse durch mäßiges Hineindrücken plan an der Platinvorderkante auszurichten. Danach können die restlichen Beinchen verlötet werden.

Der "grüne" Pin der RGB-LED kann entfernt werden, da nur die Farben Rot und Blau verwendet werden. Das Abwinkeln der Beinchen sollte vor dem Einlöten vorgenommen werden. Alternativ kann auch eine DUO-LED (zum Beispiel rot/grün, Reichelt "LED 5 RG-3") verwendet werden. In dem Fall ist eventuell der Wert des Vorwiderstandes R10 anzupassen.

Zwei der Kondensatoren (C28/C29) sind optional zu bestücken. Diese stellen einen Tiefpaß dar und verringern die Flankensteilheit der vom PSG erzeugten Töne, welche damit "weicher" klingend erscheinen. Damit verbunden ist jedoch ein gewisser Höhenverlust. Die Werte können zwischen 1 nF und 100 nF eingestellt werden, je nach Bedarf.

Die liegend einzubauenden Widerstände sind für die Bauform 0207 ausgelegt. Da es jedoch herstellerspezifische Unterschiede, sowohl was die Form als auch die Größe anbelangt, geben kann, empfiehlt es sich, ggf. auf Widerstände der Bauform 0204 zurückzugreifen.

Der Kondensator C14 (Stützkondensator für den OPV), welcher sich im Innenraum der Schaltung befindet, muß exakt mittig eingebaut werden, damit die DIL8-Fassung plan auf die Platine montiert werden kann. Es empfiehlt sich, einen liegenden Kondensator zu verwenden.

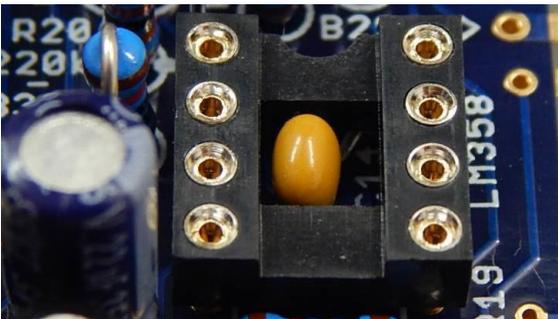


Abbildung 9: Innenliegender Kondensator C14

Das Modulgehäuse weist bei einigen Tonfrequenzen und der Verwendung des internen Lautsprechers Resonanzen auf, welche sich in einer Art "Surren" äußern. Abhilfe kann mit folgenden Maßnahmen erreicht werden:

- Lautsprecher mit einem Stück Schaumgummi festklemmen
- etwas Schaumgummi zwischen Platine und Unter-/Oberschale
- ggf. in hartnäckigen Fällen die Verstärkung des OPV heruntersetzen (Verringern des Rückkopplungswiderstandes R20)

5.3 Inbetriebnahme und Benutzung des M066

Die Inbetriebnahme gestaltet sich denkbar einfach. Wird das KC-System eingeschaltet, ist auch das M066 sofort betriebsbereit. Es muß jetzt nur noch mit Daten zur Musik- bzw. Tonausgabe versorgt werden.

Dazu entstanden parallel zur Modulentwicklung neben den bereits gezeigten Ansteuerbeispielen einige weitere Programme, sowohl speziell für das M066 als auch universal für andere Rechnertypen.

5.3.1 Software für das M066

Nachfolgend werden die Anwendungsmöglichkeiten und Software beschrieben. Zu allen Beispielen liegen ausführbare Dateien (KCC bzw. KCS) sowie Quelltexte (ASM, BASIC) vor. Mit Ausnahme der "AY-3-8910-Sounddemo" enthalten die Programme keinen Test auf Vorhandensein eines M066. Fehlt dieses, bleibt das Programm "stumm".

Die gesamte Software liegt einschließlich der Quelltexte als ZIP-Archiv vor.

Auf eine Verwendung der Speicherausbaumöglichkeiten bei den "großen" KC85/x wurde nicht zurückgegriffen - hier bestehen noch viele Möglichkeiten für Musikenthusiasten. Alle Beispiele sollten deshalb auf allen KC85/2...5 (ab 16 KB Grund-RAM) lauffähig sein.

5.3.1.1 Einzeltonprogrammierung

Die Einstellung der PSG-Register für eine Einzeltonerzeugung ist recht einfach. Nachfolgend ist ein Beispiel für einen Dreiklangton ("Dampfersirene") dargestellt.

```

10   BASE=56           :REM AY-Basisadresse
20   FOR R=0 TO 13    :REM Registernummer
30   READ W           :REM zu schreibenden Registerinhalt holen
40   OUT BASE,R       :REM Register auswählen
50   OUT BASE+1,W     :REM Inhalt schreiben
60   NEXT
70   END
80   DATA 221,1,123,1,62,1,0,248,16,16,16,255,16,13

```

Listing 2: Dampfersirene – BASIC-Beispiel

Um den damit angestoßenen Dauerton abzustellen, ist das Mischerregister neu zu setzen.

In BASIC kann zum Beispiel nur die Lautstärke auf 0 gesetzt werden.

```

OUT BASE,7
OUT BASE+1,255

```


5.3.1.2 AY-3-8910-Sounddemo

Zum Test und Nachweis der Funktionsfähigkeit des M066 wurde ein Demonstrationsprogramm mit folgenden Anforderungen entwickelt:

- Test auf Vorhandensein des M066 im System
- Test der Kanaluordnung
- Erzeugung von Tonausgaben per Tastatur ("Mini-Piano")
- Abspielen einfacher Geräusche per Kennbuchstaben
- Abspielen eingebauter Soundtracker-Dateien per Kennziffer

```

*** AY3-8910 Sounddemo - WeRo 05/2017***
PING (1x)
L=Links = Kanal C
R=Rechts = Kanal A
M=Mitte = Kanal B

C-D-E-F-G-A-H-C...
P = Piano

GERAEUSCHE
D = Dampfer
A = Amboss
H = Hubschrauber (Doku A5105)
W = Wasser (JKCEMU)
S = Schuss (1x) (JKCEMU)

STC-SOUNDMODULE (Internet)
1 = Jingle
2 = Song
3 = Song
4 = Song
5 = Song
6 = Song
7 = Song

Leertaste = STOPP
0 = Demo beenden

Auswahl? _

```

Abbildung 10: AY-3-8910 Sounddemo (Quelle in Klammern)

Die Bedienung ist nicht schwer. AY.KCC laden (Adresse 0300h...ca. 227Fh, davon ca. 6 KB für 7 STC-Soundmodule). Die Datei ist autostartfähig. Wird kein M066 gefunden, so wird dies angezeigt und das Programm abgebrochen. Die weitere Bedienung erfolgt per Menü (Kommandoauswahl per Einzeltaste). Nach Beendigung mit "0" ggf. aus dem Menü mit "AY" neu starten.

Nachfolgend werden die einzelnen Funktionen und Bestandteile des Programms genauer beschrieben.

Ping

- Im M066 wurde folgende Zuordnung der Kanäle in der Stereo-Ausgabe festgelegt:
 Kanal A => Rechts
 Kanal B => Mitte (monofon, auf beide Kanäle gleichzeitig)
 Kanal C => Links
- Mit den Tasten L, R und M läßt sich diese Zuordnung per Tonausgabe prüfen.

Piano

- Es werden die Tasten Q bis P abgefragt. Jede Taste entspricht einer Note (C-D-E-F-G-A-H-C-D-E).
- Die Tondauer ist fest, ein "Anschlagseffekt" ist vorhanden.
- Nach Tastendruck werden die der Notenfrequenz zugeordneten Registerwerte aus einer Tabelle geholt.

```
;REGISTER 0, 1, 2 3 4 5 6 7 8 9 10 11 12 13
; -----TONHÖHEN----- LAUTSTÄRKE HÜLLKURVE
; ---A--- ---B--- ---C--- R MIX A B C FEIN GROB FORM
C1: DEFB #DD,#01,#00,#00,#00,#00,#00,#FE,#10,#00,#00,#00,#30,#01
D1: DEFB #A9,#01,#00,#00,#00,#00,#00,#FE,#10,#00,#00,#00,#30,#01
E1: DEFB #7B,#01,#00,#00,#00,#00,#00,#FE,#10,#00,#00,#00,#30,#01
F1: DEFB #65,#01,#00,#00,#00,#00,#00,#FE,#10,#00,#00,#00,#30,#01
G1: DEFB #3E,#01,#00,#00,#00,#00,#00,#FE,#10,#00,#00,#00,#30,#01
A1: DEFB #1C,#01,#00,#00,#00,#00,#00,#FE,#10,#00,#00,#00,#30,#01
H1: DEFB #FD,#00,#00,#00,#00,#00,#00,#FE,#10,#00,#00,#00,#30,#01
C2: DEFB #EE,#00,#00,#00,#00,#00,#00,#FE,#10,#00,#00,#00,#30,#01
D2: DEFB #D4,#00,#00,#00,#00,#00,#00,#FE,#10,#00,#00,#00,#30,#01
E2: DEFB #BD,#00,#00,#00,#00,#00,#00,#FE,#10,#00,#00,#00,#30,#01
```

Listing 4: Registerwerte der zugeordneten Notenfrequenzen

- Anschließend erfolgt das Einschreiben in den AY-3-8910, wodurch der Ton abgespielt wird.

```
AYBASE: EQU #38 ;EA-ADRESSE
PROG: LD HL,TON ;LISTENANFANG 14 REGISTERWERTE
LD DE,#000D
LD A,E ;A=#0D = LETZTES REGISTER
ADD HL,DE ;LISTEN-ENDE
LD C,AYBASE+1 ;REGISTERINHALT SCHREIBEN,
P0: OUT (AYBASE),A ;REGISTER AUSWÄHLEN
OUTD ;BYTE AUF (HL) IN AKT. REG. SCHREIBEN
DEC A ;NÄCHSTES REGISTER (RÜCKWÄRTS)
JP P,P0 ;BIS INCL. REGISTER 0
RET
```

Listing 5: Programmierung des AY mit der Notenfrequenz

Der Abbruch der Piano-Funktion und die Rückkehr zum Hauptmenü des AY-Demoprogramms erfolgt mit <BRK> bzw. auf der D005 oder einer PC-Tastatur mit <STRG>+<C>.

Geräusche

- Ähnlich der Piano-Funktion werden mit Kennbuchstaben (Anfangsbuchstaben) Geräusche geladen und abgespielt (Prinzip des Abspielens: siehe unter "Piano").
- Bis auf "Schuß" laufen die Geräusche endlos in einer Schleife.
- Ein Abbruch erfolgt entweder mit der Leertaste (Klang stoppen), einem neuen Menübuchstaben oder "0" für Programmende.

```

;REGISTER 0, 1, 2 3 4 5 6 7 8 9 10 11 12 13
;
; -----TONHÖHEN----- LAUTSTÄRKE HÜLLKURVE
; ---A--- ---B--- ---C--- R MIX A B C FEIN GROB FORM
DAMPFE: DEFB #DD,#01,#7B,#01,#3E,#01,#00,#F8,#10,#10,#10,#00,#08,#0D
AMBOSS: DEFB #6F,#00,#5F,#00,#4F,#00,#00,#F8,#10,#10,#10,#FF,#20,#08
HUBSCH: DEFB #6F,#00,#00,#00,#00,#00,#1E,#F6,#10,#00,#00,#2C,#01,#0E
MEER: DEFB #00,#00,#00,#00,#00,#00,#14,#CF,#00,#10,#08,#00,#32,#0E
SCHUSS: DEFB #00,#00,#00,#00,#00,#00,#14,#EF,#00,#10,#00,#00,#1F,#01

```

Listing 6: Benutzte Registerwerte für die Geräusche

STC-Soundmodule

- In der Demo sind sieben STC-Musikdateien enthalten.
- Die Betätigung einer Taste 1...7 startet das Abspielen der jeweiligen Musik. Wie bei den meisten Soundtrackern/Playern erfolgt ein endloses Abspielen.
- Die Betätigung der Leertaste stoppt die aktuelle Musik; mit einem anderen Kennbuchstaben (oder einer Ziffer) wird eine neue Musik abgespielt oder mit der Taste "0" das Programm verlassen.

5.3.1.3 Selbstabspielende Musik

Dazu wurde für den KC85/2-5 ein neuer Dateityp definiert. Analog KCC und KCB gibt es für "Sound" jetzt noch KCS. Eine solche Datei

- entspricht der "Norm" der KC-85/2-5-Dateien (inklusive Vorblock),
- dient einzig dem Abspielen von Musik (ggf. mit grafischer Ausgabe als "Untermalung"),
- kann verschiedene Klangdaten enthalten (Format ist nicht festgelegt),
- enthält neben den Klangdaten einen hierfür geeigneten Player,
- arbeitet nicht mit anderen CAOS-Programmen zusammen, wenn sich die Speicherbereiche überschneiden.

Das Laden erfolgt mittels der Befehle ULOAD, FLOAD oder LOAD. Die Dateierweiterung "KCS" ist dabei immer mit anzugeben.

Die KCS-Dateien starten selbständig den eingebauten Player, welcher die enthaltene STC-Musik abspielt. Wird "MUS01" aus dem Menü aufgerufen, kann hinter dem Programmnamen die Anzahl der Wiederholungen angegeben werden (max. 0FFh = 255 mal).



Abbildung 11: Grafische Untermauerung während des Abspielens einer KCS-Datei

Demo "MUS01.KCS"

- Speicherbereich: 0300h... ca. 167Fh (je nach Musiklänge)
- Titel-Anzeige der STC-Musik (normalerweise immer "SONG BY ST COMPILE")
- Fortschrittsanzeige der Pattern
- Pegelanzeige der drei Kanäle (rot: Hüllkurve ist aktiv)
- Durch Drücken einer beliebigen Taste kann das Abspiel beendet werden.

Für die anderen Musikbeispiele MUS02 bis MUS10 gilt dies entsprechend.

Eigene KCS-Musiken lassen sich wie folgt erstellen:

- Eine vorhandene KCS-Beispiel-Datei nehmen.
- Mittels HEX-Editor die alte Musik löschen (ab roter Markierung bis Dateiende).

| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|
| 00000660 | B7 | C9 | F5 | CD | DC | 08 | F1 | B7 | 0E | 10 | 28 | 1D | FE | 10 | 38 | 0C | .Éü.ä... (p.8. |
| 00000670 | D6 | 10 | F5 | CD | D8 | 08 | F1 | FE | 00 | 20 | 01 | 3C | 47 | 3E | 10 | 90 | Ö.äiø.äp... <G>. |
| 00000680 | 4F | 3E | 5B | CD | 03 | F0 | 00 | 10 | F8 | 41 | AF | B8 | C8 | 3E | 20 | CD | O>[í.ä...äA^,É> í |
| 00000690 | 03 | F0 | 00 | 10 | F8 | C9 | 07 | 96 | 03 | BF | 03 | 22 | 04 | 53 | 4F | 4E | .ä...äÉ...ä...SON |
| 000006A0 | 47 | 20 | 42 | 59 | 20 | 53 | 54 | 20 | 43 | 4F | 4D | 50 | 49 | 4C | 45 | 51 | G BY ST COMPILEQ |
| 000006B0 | 0B | 02 | 0D | 00 | 00 | 0D | 80 | 00 | 0C | 80 | 00 | 0C | 80 | 00 | 0B | 80 | |

- An der gleichen Stelle die neue Musik im STC-Format einfügen.
- Das Dateiende im Vorblock korrigieren (Wert hier nur Beispiel).

| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------------|
| 00000000 | 4D | 55 | 53 | 30 | 31 | 20 | 20 | 20 | 4B | 43 | 53 | 00 | 00 | 00 | 00 | 00 | MUS01 KCS..... |
| 00000010 | 03 | 00 | 03 | 5E | 13 | 08 | 08 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ...^..... |

- Ggf. den Lade- und Menünamen ändern.

5.3.2 Universaler AY-Player

Obwohl nachfolgend nur auf die Rechnertypen KC85/2-5 Bezug genommen wird, ist der Einsatz des AY-Players universal auch auf anderen Rechnern möglich, z.B. auf K1520-Bus-basierten Typen.

5.3.2.1 Include-Datei für AY-Player in Programmen

Bei der Assembler-Include-Datei "AYPLAY.INC" handelt es sich um eine "Player-Engine", die AY-Musik im STC-Format auf dem KC85/2-5 mit dem Modul M066 abspielt. Darüber hinaus ist sie auch für das Erzeugen einfacher Töne und Geräusche per CALL geeignet (einmalige Programmierung des PSG mit einer Liste von 14 Registerwerten).

Die Engine enthält alle nötigen Routinen inklusive der Interrupterzeugung (50Hz). Für andere Rechnertypen als KC85/2-5 muß die in der Include-Datei enthaltene Interrupterzeugung (das Klangmodul M066 enthält einen eigenen CTC) ggf. modifiziert werden.

Im Anwenderprogramm muß einfach nur die Anfangsadresse der gewünschten Musik und die Anzahl der Wiederholungen (max. 255) übergeben werden. Per einfachem CALL wird das Abspielen gestartet und (wenn nötig) vorzeitig gestoppt. Über eine Tabelle ist der Zugriff auf diverse Musikparameter möglich.

Im Folgenden werden die Bedingungen für den Einsatz der Include-Datei beschrieben.

Im eigenen Assembler-Programm ist der Beginn der Arbeitszellen (WORK) festzulegen. Standardmäßig ist dafür der Bereich ab 0080h (...00C7h) vorgesehen, kann aber bei Erforderlichkeit anders definiert werden.

Der INCLUDE-Eintrag für den Player ist an geeigneter Stelle einzufügen. In den Beispielen wurde dieser an den Anfang (PLAYER) gesetzt.

Es dürfen keine Labels X0001-X0100 enthalten sein (diese werden in der INCLUDE-Datei benutzt).

Der Zugriff auf die AY-Player-Daten und -Routinen erfolgt über die folgende Wertetabelle:

```

WORK          ;WORD, ÜBERGABE ANFANGSADESSE SONG/GERÄUSCH
WORK+#02     ;BYTE, ÜBERGABE ANZAHL SONG-WIEDERHOLUNGEN
;
WORK+#0D     ;BYTE, RÜCKGABE PATTERNANZAHL IM SONG
WORK+#35     ;BYTE, RÜCKGABE AKTUELLE PATTERN-POSITION
WORK+#3F     ;BYTE, RÜCKGABE AKT. LAUTSTÄRKE KANAL A
WORK+#40     ;BYTE, RÜCKGABE AKT. LAUTSTÄRKE KANAL B
WORK+#41     ;BYTE, RÜCKGABE AKT. LAUTSTÄRKE KANAL C
;
;WEITERE BEZÜGE SIEHE QUELLTEXT DER INCLUDE-DATEI...
;CALLS:
PLAYER       ;SONG LADEN+ABSPIELEN
PLAYER+3     ;GERÄUSCH LADEN+ABSPIELEN
PLAYER+6     ;SOUND OFF

```

Listing 7: AY-Player-Daten und -Routinen

Die Adressen 0100h...0107h werden durch die Interruptvektoren des CTC belegt. Sollen AY-Musiken abgespielt werden, darf der Bereich nicht für andere Zwecke verwendet werden. Vom Anwender benötigte weitere Arbeitszellen können von 00D0h...00FFh liegen (siehe auch Kapitel "[10.2 Hinweis zu den verwendeten Arbeitszellen](#)").

STC-Musiken lassen sich fest im Programm verankern. Eine Nachlademöglichkeit verschiedener Musiken vom Datenträger ist auch denkbar. Es gibt jedoch z.Z. keine (mit den KC85-Bordmitteln) allgemein nutzbare Laderoutine. Erst ab CAOS 4.7 ist dies möglich.

Der mit der Include-Datei erzeugte Player ist (im Gegensatz zur AY-Demo) ROM-fähig und auf beliebige Adressen assemblierbar.

Die Datei liegt im ASide-Format vor, läßt sich jedoch leicht an andere Assembler-Dialekte anpassen.

Beispiel

```

WORK: EQU #0080 ;ANFANG ARBEITSZELLEN (BIS WORK+#47)
PLAYER: EQU #0300 ;PLAYER (AM DATEIANFANG)
;CAOS VORBLOCK-----
ORG PLAYER-#80
DEFM 'TEST' ;NAME FÜR LOAD, 8 STELLEN
DEFM 'KCC'
DEFS 5,0
DEFB 3 ;3=MIT AUTOSTART
DEFW PLAYER, ENDE, START
DEFS 105,0
ORG PLAYER
;!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
INCLUDE "AYPLAY.INC" ;AY-PLAYER-ENGINE (LÄNGE ENGINE CA. #4FF)
;!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
ORG PLAYER+#0500 ;WEGEN PLAYER-LÄNGE!
DEFB #7F, #7F ;PROLOG
DEFM 'TEST', 1 ;MENÜEINTRAG
;-----
START: ;...HIER ANWENDERPROGRAMM
;LD HL, SONG ;ANFANGSADRESSE SONG
;LD (WORK), HL ;ÜBERGEBEN
;CALL PLAYER ;SONG ABSPIELEN
;...
;CALL PLAYER+6 ;SONG STOPPEN
;...
;LD HL, SCHUSS ;GERÄUSCH LADEN
;LD (WORK), HL
;CALL PLAYER+3 ;UND ABSPIELEN
;...
RET
;-----
SCHUSS: DEFB #00, #00, #00, #00, #00, #00, #14, #EF, #00, #10, #00, #00, #1F, #01
;-----
SONG:
INCLUDEBINARY "SONG.STC" ;HIER WIRD DER SONG EINGEBUNDEN
ENDE:

```

Listing 8: Beispiel für die Verwendung der Include-Datei für den AY-Player

5.3.2.2 Verwendung des AY-Players unter KC-BASIC

Das Abspielen von einfachen Tönen und Geräuschen ist ziemlich einfach und unkritisch. Das kann mit wenigen OUT-Anweisungen erfolgen. Für das Abspielen kompletter Musik ist ein Player nötig, der als Maschinenkodeprogramm aus BASIC heraus aufgerufen wird.

Folgende Probleme bestehen dabei:

- Player und Musik müssen im Grund-RAM liegen, die "hinteren" Speicherbänke (z.B. hinter dem IRM) lassen sich noch nicht nutzen.
- Es muß geklärt werden, wie Player und Musik geladen werden können, ggf. ist die AY-Musik fest im BASIC-Programm zu verankern.

Eine Lösungsmöglichkeit gestaltet sich folgendermaßen:

- Das obere Speicherende des BASIC-RAM wird auf 5FFFh herabgesetzt (23484 Bytes frei) und somit ist Platz für Player und Musik vorhanden.
- CAOS 4.7 ermöglicht eine Verfahrensweise mit nachladbarem Player und Musik.

Ein Player AYPLAY60.KCC steht als Maschinenprogramm (Ladeadresse 06000h) zur Verfügung. Bei Bedarf kann er auch auf andere Adressen assembliert werden. Er besitzt feste Einsprungadressen.

06000h Abspiel starten

06006h Abspiel beenden

Des weiteren verwendet der Player feste Arbeitszellen.

0080h Übergabe Musik-Anfangsadresse

0082h Anzahl der Wiederholungen (1-255)

Eine Prüfung auf Vorhandensein des M066 erfolgt nicht; das Programm arbeitet dann nur ohne Klंगाusgabe.

```

10 REM-AY-PLAYER-BASIC-DEMO FUER CAOS 4.7
20 CLS
30 CLEAR50,24575:REM PLATZ SCHAFFEN
40 COLOR1
50 DEVICE 1:REM USB EINSTELLEN
60 PRINT"LADE PLAYER..."
70 BLOAD"AYPLAY60.KCC"
80 PRINT"LADE SONG..."
90 AY$="AYSONG.KCT"
100 BLOAD AY$
110 COLOR5
120 DOKE 128,25856:REM ANFANG SONG
130 POKE 130,2:REM WIEDERHOLUNGEN
140 REM-ANWENDER-PROGRAMM-----
150 PRINT"SPIELE AY-SONG"
160 CALL*6000:REM PLAYER STARTEN
170 REM-ANWENDER-ROUTINEN...
180 FOR I=1 TO 100
190 PRINT"HALLO WELT ";
200 PAUSE 5

```

```

210 NEXT
220 PRINT:PRINT"PROGRAMM-ENDE"
230 PRINT"SONG WIRD AUCH BEENDET..."
240 CALL*6006:REM PLAYER STOPPEN
250 END
    
```

Listing 9: Beispielprogramm für KC-BASIC

Wird das Programm mit <BRK> abgebrochen, so läuft eine gestartete Musik weiter. Diese muß manuell per "CALL*6006" gestoppt werden.

Die Musik muß bei diesem Player vom Typ STC sein, ihre Länge darf bis zu 6 KB betragen. Sie ist mit einem Vorblock (ohne Autostart!) zu versehen, da BLOAD nur solche Dateien laden kann:

| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | | |
|----------|----|--------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------------|
| 00000000 | 41 | 59 | 53 | 4F | 4E | 47 | 20 | 20 | 4B | 43 | 54 | 00 | 00 | 00 | 00 | 00 | 00 | AYSONG KCT..... |
| 00000010 | 02 | 00 | 65 | 98 | 73 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ..els..... |
| 00000020 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000030 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000040 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000080 | 06 | 5C | 04 | 91 | 04 | 36 | 05 | 53 | 4F | 4E | 47 | 20 | 42 | 59 | 20 | 53 | 00 | \\..6.SONG BY S |
| 00000090 | 54 | 20 | 43 | 4F | 0D | 50 | 49 | 4C | 45 | 98 | 0E | 01 | 0F | 20 | 00 | 0F | 00 | T COMPILE |
| ... | | | | | | | | | | | | | | | | | | |
| 00000F00 | 48 | 48 | 4C | 4F | A7 | 4D | A5 | 80 | FF | B0 | 30 | A4 | 3A | 37 | A8 | 33 | 00 | HHLOSM# ÿ*0#;7"3 |
| 00000F10 | B0 | 35 | A4 | 3C | 37 | A8 | 30 | FF | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | *5#<7"0ÿ |
| 00000F20 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000F30 | 00 | Endeadresse: #0F18 | | | | | | | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000F40 | 00 | + Offset #5480 | | | | | | | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000F50 | 00 | = Dateiende #7398 | | | | | | | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000F60 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000F70 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |

Abbildung 12: Ändern der Adreßwerte der einzubindenden STC-Datei

5.3.2.3 Verwendung des AY-Players unter JKCEMU-Basic

Wer nicht komplett mit Assembler arbeiten kann/möchte, dem sei der "JKCEMU-Basic"-Compiler empfohlen. Damit ist es möglich, im BASIC-Stil mit den vom KC-BASIC her bekannten Anweisungen (und einigen mehr) ein Programm zu erstellen, was dann zur Ausführung keinen BASIC-Interpreter mehr benötigt. Ein weiterer Vorteil ist, daß sich die unter KC-BASIC bestehende Speicherplatzproblematik für die AY-Musik entschärft.

```

REM---VORARBEITEN-----
A=ASM(" LD HL,MUSI"):REM ANFANGSADRESSE SONG ERMITTELN
DOKE 128,A:          REM AN PLAYER UEBERGEHEN
POKE 130,5:         REM ANZAHL WIEDERHOLUNGEN UEBERGEHEN
REM---DAS ANWENDERPROGRAMM-----
CLS
PRINT "STARTE AY-SONG..."
CALL *0300:         REM SONG STARTEN
PAUSE 10
PRINT "MACHE HIER IRGEND ETWAS..."
FOR I=1 TO 100
  PRINT I;
  PAUSE 2
NEXT I
PRINT
PRINT "PROGRAMM ZUENDE, BEENDE AUCH SONG"
CALL *0306:         REM SONG STOPPEN
END
REM---EINBINDEN SONGDATEI-----
ASM DATA "MUSI:"
ASM DATA " BINCLUDE 'E:\song.stc'"

```

Listing 10: Beispielprogramm für den JKCEMU-Basic-Compiler

In diesem Programm wird mit wenigen Anweisungen das Abspielen der Musik an der gewünschten Stelle gestartet bzw. gestoppt.

Für den Test sind die beiden Anweisungen

```
CALL *0300
```

```
CALL *0306
```

zunächst als Kommentar zu schreiben, da noch kein Player vorhanden ist.

Die STC-Musik kann original übernommen werden, ihr Name und Pfad sind anzupassen

Auf die gleiche Art und Weise können weitere Musiken (natürlich mit neuen Bezeichnungen) in das Endprodukt aufgenommen werden. Um dann zu einer anderen Musik zu wechseln, sind vor jedem Abspielaufruf die unter "Vorarbeiten" angeführten Anweisungen mit der entsprechenden Musikbezeichnung auszuführen.

Läuft das BASIC-Programm im JKCEMU zufriedenstellend (Test noch ohne Musik), dann werden die Kommentaranweisungen (REM) vor den beiden Musik-CALLS entfernt. Nun wird endgültig kompiliert. Dabei sind einige Einstellungen vorzunehmen, welche in den nachfolgenden Abbildungen dargestellt werden.

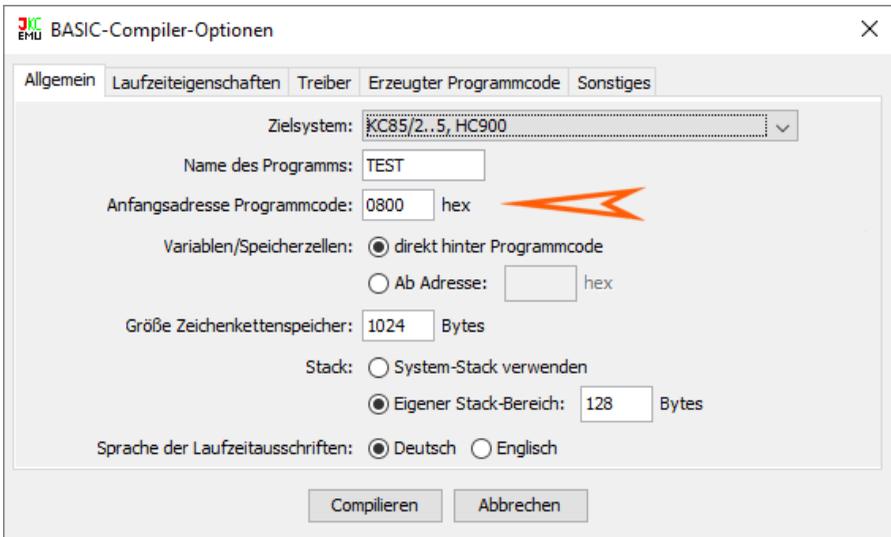


Abbildung 13: Die Anfangsadresse des Programms liegt bei 0800h

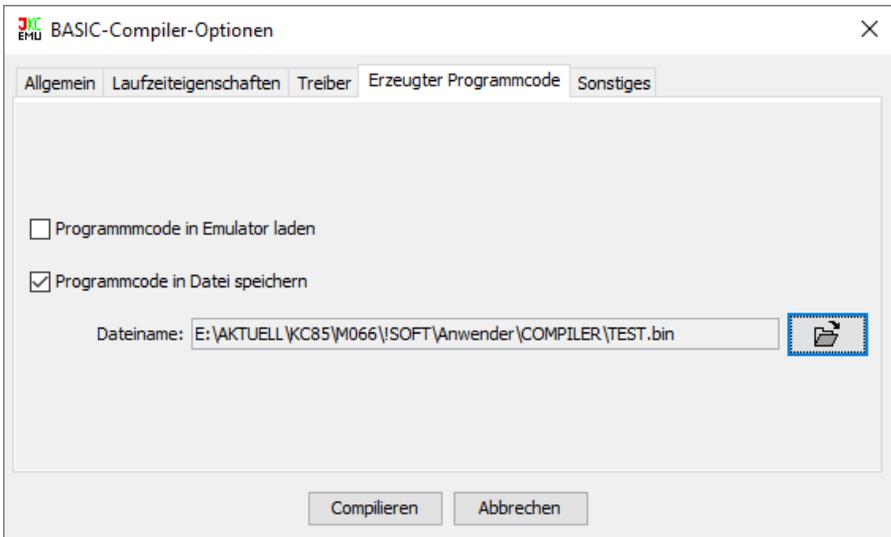


Abbildung 14: Der Name der zu erzeugenden BIN-Datei (raw-Format)

Andere Compiler-Einstellungen können je nach Kenntnis und Bedarf vorgenommen werden. Nach dem Betätigen der "Compilieren"-Schaltfläche erscheinen die Fertigmeldung oder die Fehlermeldungen.

Zum Schluß wird per HEX-Editor das Kompilat "TEST.BIN" an ein "Gerüst" angehängt. Dabei handelt es sich um ein vorgefertigtes Binärfile AYPLAYMC.BIN, welches den CAOS-Vorblock sowie einen AY-Player für STC-Files beinhaltet und ab Adresse 0300h beginnt.

Für problemloses Laden sollte dann am Ende noch mit 00 auf volle Blöcke (128 Bytes) aufgefüllt werden. Die gesamte Datei ist nun unter einem neuen Namen (TEST.KCC) abzuspeichern und der Vorblock etwas anzupassen.

| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------------------------------|
| 00000000 | 54 | 45 | 53 | 54 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | TEST |
| 00000010 | 03 | 00 | 03 | DD | 0E | 07 | 08 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ...ÿ..... |
| 00000020 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000030 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000040 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000080 | C3 | 34 | 03 | C3 | 1D | 03 | C3 | 22 | 03 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | Ä4.Ä.Ä"..... |
| 00000090 | 00 | 00 | 00 | 00 | 00 | 00 | FF | 00 | 00 | 00 | 00 | 00 | 01 | 2A | 80 | 00 | █.....ÿ.....* . |
| 000000A0 | 18 | 07 | 3E | 43 | D3 | 3D | 21 | 0F | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ...>CÓ=!.....i |
| ... | | | | | | | | | | | | | | | | | |
| 00000570 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000580 | 7F | 7F | 54 | 45 | 53 | 54 | 01 | ED | 73 | E5 | 0E | 31 | 6B | 0F | 2A | 9C | TEST.isä.1k.* |
| 00000590 | B7 | 7C | B5 | 20 | 0D | 2A | 9E | B7 | 7C | FE | 20 | 38 | 05 | 7D | FE | 28 | µ .* b 8.)b(|
| 000005A0 | 30 | 2E | 21 | 00 | 00 | 22 | 9C | B7 | 21 | 28 | 20 | 22 | 9E | B7 | 3A | 9B | 0.!.! !(" . |
| 000005B0 | B7 | 6F | 26 | 00 | 4D | 44 | 29 | 29 | 09 | 29 | 11 | 9C | B9 | 19 | 36 | 00 | ·o&.MD)). ¹.6. |
| 000005C0 | 23 | 36 | 00 | 23 | 36 | 28 | 23 | 36 | 20 | 23 | 3E | 0C | CD | 03 | F0 | 00 | #6.#6(#6 #>.î.ä. |
| ... | | | | | | | | | | | | | | | | | |
| 00000C30 | FF | A1 | 63 | 24 | 24 | 24 | 24 | 24 | 24 | 24 | 24 | 24 | 24 | 24 | 24 | 24 | ÿic\$\$\$\$\$\$\$\$\$\$\$\$ |
| 00000C40 | 24 | 24 | 24 | 24 | 24 | 24 | 24 | 24 | 24 | 24 | 24 | 24 | 64 | 24 | 24 | 24 | \$\$\$\$\$\$\$\$\$\$\$\$sd\$\$\$ |
| 00000C50 | 24 | 24 | 24 | 24 | 24 | 24 | 24 | 24 | AC | 61 | 70 | 00 | FF | 00 | 00 | 00 | \$\$\$\$\$\$\$\$-ap.ÿ... |
| 00000C60 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 00000C70 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |

Einsprungadresse: #0587
 + Offset #0280
 = Autostartadresse #0807

Endeadresse: #0C5D
 + Offset #0280
 = Dateiende #0EDD

Abbildung 15: Ändern der Adreßwerte im Vorblock der TEST.KCC-Datei

Änderungen im Vorblock:

- Dateiname
- Dateilänge
- Autostartadresse

Nun kann die Datei "TEST.KCC" geladen werden.

6 Anwendungsfall / Musikerzeugung

Die Motivation für die Neuentwicklung des M066 war genaugenommen die letztendlich erfolgreiche Entwicklung des Moduls M041 - 2 x 16 KB EEPROM. Durch die Möglichkeit, ein Autostartprogramm zu entwickeln, welches auch Musik beinhaltet, war die Entwicklung eines Klangmoduls sehr verlockend, denn damit besteht die Möglichkeit, das Autostartprogramm mit einer Hintergrundmusik zu versehen, welche aufgrund der Interruptfähigkeit des Moduls keinen störenden Einfluß auf den Programmablauf haben wird. Hinzukommt die wesentlich bessere Qualität der erklingenden Musik, welche mit dem PSG erzeugt werden kann.

Das erwähnte Autostartprogramm ist eine kleine Bilderschau, in der Fotos aus ganz Deutschland gezeigt werden. Während des Programmstarts erklingt eine kleine Melodie – Kein schöner Land – welche über den KC85-internen Lautsprecher ausgegeben wird. Aufgrund der beschränkten Hardwaremöglichkeiten klingt diese Melodie entsprechend.

Kein schöner Land Anton Wilhelm von Zuccalmaglio

Kein schö-ner Land in die-ser Zeit, als hier das uns - re weit und
breit, wo wir uns fin - den wohl un - tern Lin - den zur A - bend -
zeit, wo wir uns fin - den wohl un - tern Lin - den zur A - bend - zeit.

Abbildung 16: Auszug aus den Noten des Liedes "Kein schöner Land"

Mit dem M066 soll eine Art Hintergrundmusik verwirklicht werden, welche während der ganzen Programmdauer erklingt.

Aber ganz so einfach ist es nicht, denn es erfordert doch ein gutes Maß an Wissen zur Programmierung des PSG und vor allem Kenntnisse der Musikerzeugung (Noten lesen).

Auch hier hat Rolf Weidlich sehr viel Vorarbeit geleistet und letztendlich eine Software entwickelt, mit der am KC85 selbst Melodien komponiert werden können: Den KC-Tracker.

7 Ausblick auf den KC-Tracker für CAOS 4.7

Zum Zeitpunkt der Drucklegung dieses Handbuches befand sich der KC-Tracker noch im Entwicklungsstadium.

Rolf Weidlich hat als Testmelodie doch tatsächlich "Kein schöner Land" gewählt und damit natürlich beste Voraussetzungen zur Entwicklung der weiter oben genannten Hintergrundmusik geschaffen.

Die nachfolgende Abbildung zeigt die Entwicklung des Musikstücks KSL.ST1 am KC-Tracker in dessen Version 0.5.

```

Länge      002   KSL      .ST1   F1=Info
Position   001   Aktuell X01   F2=Laden
Muster     001   Kan.A: EIN   F3=Sichern
Tonlage    000   Kan.B: EIN   F4=Leeren
*-----*   Kan.C: EIN   F5=Abspielen
IM066-SOUND|   F6=Bearbeiten
IKC-TRACKER|  SH+F6=Spezial
*-----*   Haupt-Menü   Ende

```

| | | | | | | |
|----|-----|------|------|------|------|-----------|
| 60 | A-4 | 0000 | --- | 0000 | --- | 0000 |
| 61 | --- | 0000 | --- | 0000 | --- | 0000 |
| 62 | G-4 | 0000 | --- | 0000 | A#3 | 0000 |
| 63 | --- | 0000 | --- | 0000 | --- | 0000 |
| -> | 00 | C-4 | 1000 | IC-2 | A000 | IF-3 1000 |
| 01 | --- | 0000 | --- | 0000 | --- | 0000 |
| 02 | C-4 | 0000 | --- | 0000 | --- | 0000 |
| 03 | --- | 0000 | --- | 0000 | --- | 0000 |
| 04 | C-4 | 0000 | --- | 0000 | --- | 0000 |

```

KC-TRACKER VERS. 0.5 WeRo '2017
Medium      :USB
Songname    :KSL      .ST1
Spieldauer  :ca.00:17 min
Positionen  :002
Musteranzahl: 04
Musterlänge: 64
Tempo       : 7

```

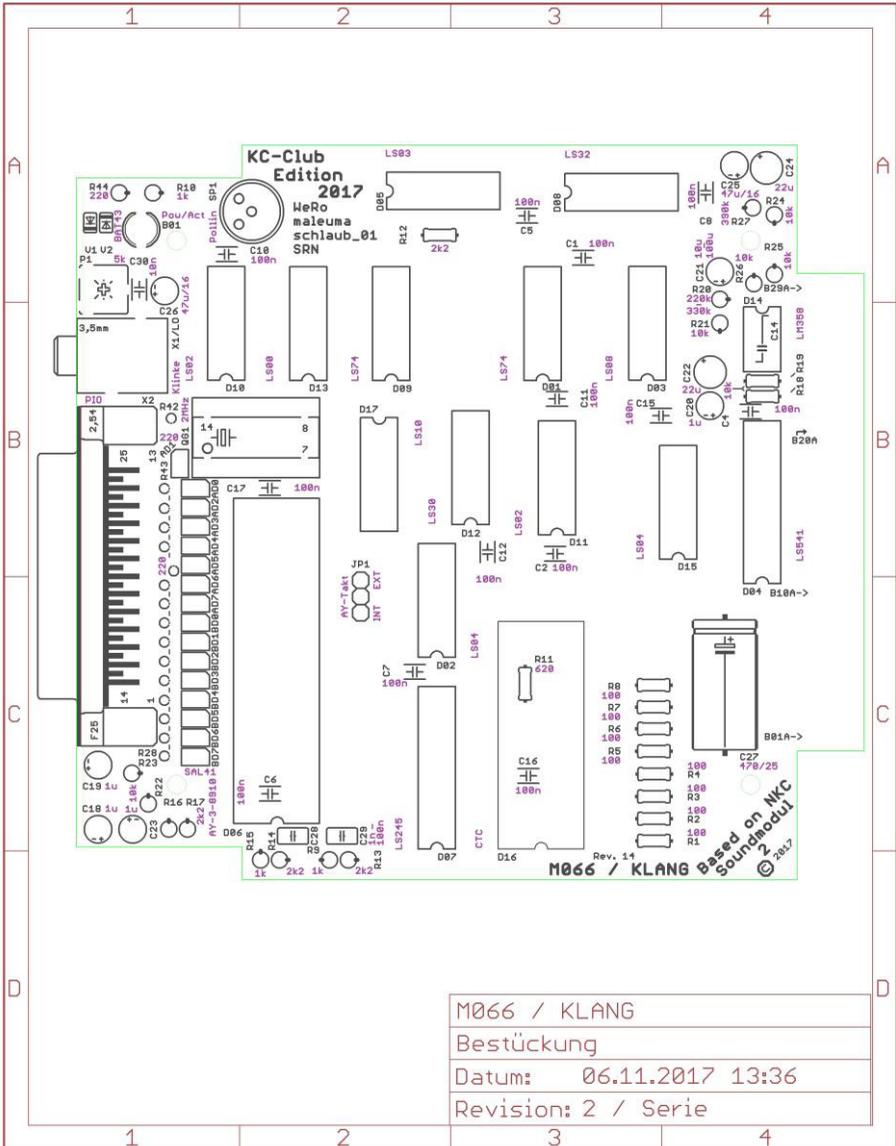
<TASTE>_

Abbildung 17: KC-Tracker im Entwicklungsstadium

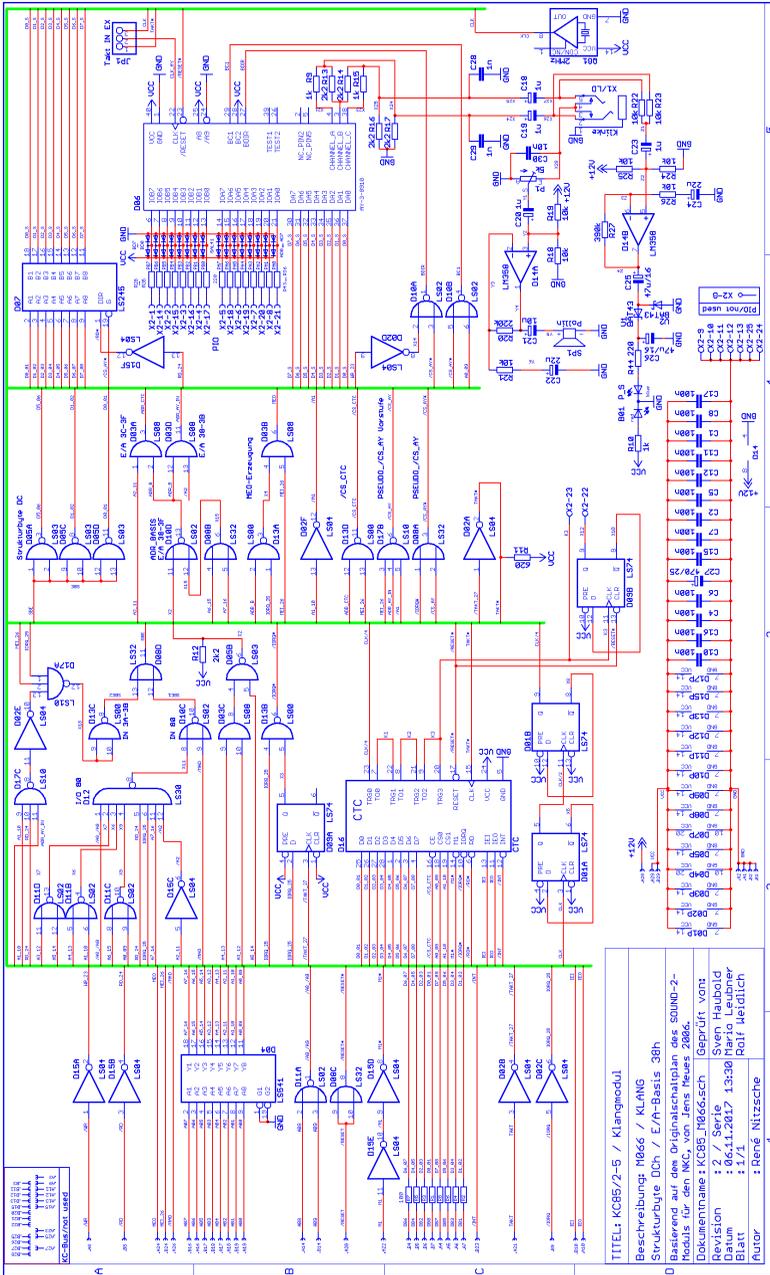
Der Zeitpunkt zur endgültigen Fertigstellung des KC-Trackers wird nach der Fertigung der Serienplatine des M066 und nach der Drucklegung dieses Handbuches liegen. Der KC-Tracker wird zusammen mit einem USB-STC-Player als separates Softwarepaket, unabhängig von dem im Kapitel "[10 Anlage C – Anwendersoftware](#)" beschriebenem Softwarepaket zur Verfügung gestellt und kann dann über den im Kapitel "[13.2 Links](#)" vorhandenen Internetlink bezogen werden.

8 Anlage A – Schalt- und Belegungspläne

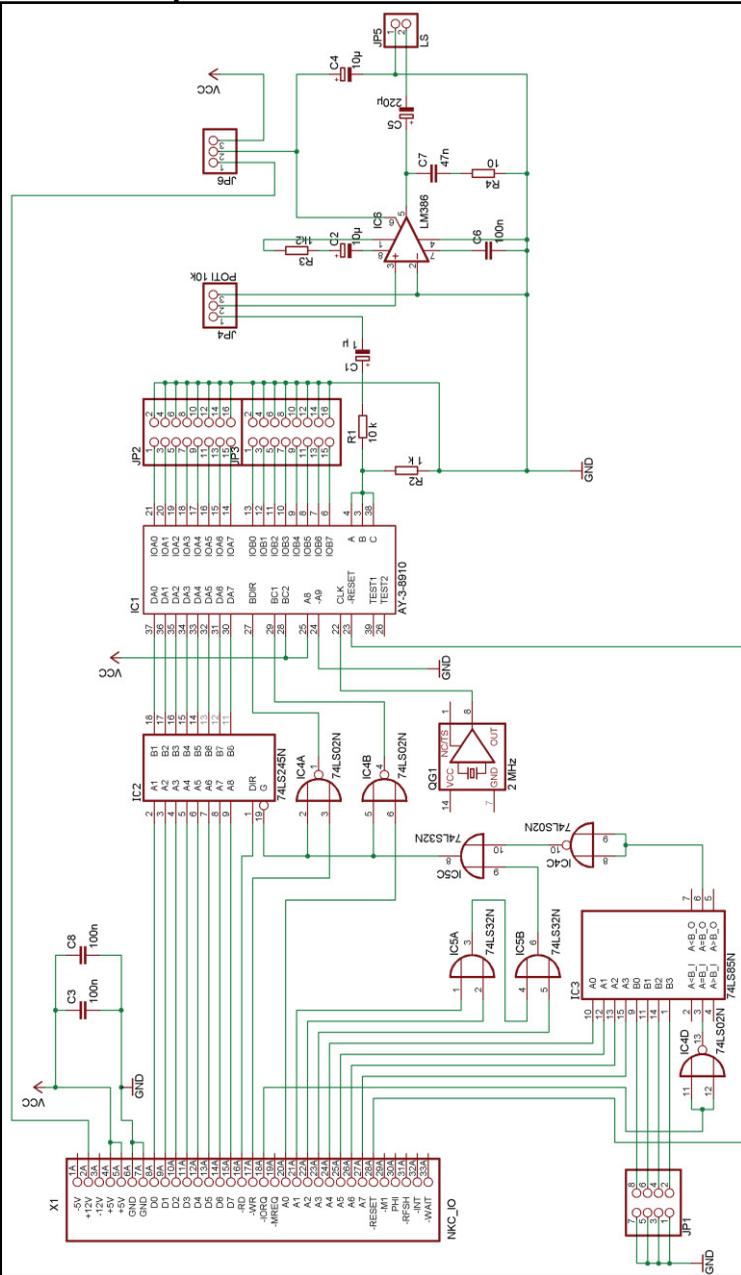
8.1 Belegungsplan – M066



8.2 Schaltplan – M066



8.3 Schaltplan – NKC Soundmodul 2



9 Anlage B – Bestückungsliste

| Name | Bauteil | Bezeichnung / Wert | Bemerkung |
|------------------------------------|-------------|----------------------|---------------------------|
| B01 | Diode | RGB-LED | gem. Katode, Niedrigstrom |
| AD0-AD7 | Diode | SAL41 | |
| BD0-BD7 | Diode | SAL41 | |
| V1/V2 | Diode | BAT43 | |
| C1, C2, C4-C8, C10-C12, C14-C17 | Kondensator | 100 nF | RM 5.0 |
| C18-C20 | Kondensator | 1 µF | Subminiatur, RM 1.5 |
| C21 | Kondensator | 10 µF - 100 µF | Subminiatur, RM 2.0 |
| C22 | Kondensator | 22 µF | RM 2.0 |
| C23 | Kondensator | 1 µF | Subminiatur, RM 1.5 |
| C24 | Kondensator | 22 µF | RM 2.0 |
| C25, C26 | Kondensator | 47µF/16V | Subminiatur, RM 2.0 |
| C27 | Kondensator | 470 µF/25V | axial, liegend |
| C28, C29 | Kondensator | 1 nF - 100 nF | RM 2.5 |
| C30 | Kondensator | 10 nF | RM 5.0 |
| D01 | Schaltkreis | LS74 | |
| D02 | Schaltkreis | LS04 | |
| D03 | Schaltkreis | LS08 | |
| D04 | Schaltkreis | LS541 | |
| D05 | Schaltkreis | LS03 | |
| D06 | Schaltkreis | AY-3-8910 | oder YM2149 |
| D07 | Schaltkreis | LS245 | |
| D08 | Schaltkreis | LS32 | |
| D09 | Schaltkreis | LS74 | |
| D10 | Schaltkreis | LS02 | |
| D11 | Schaltkreis | LS02 | |
| D12 | Schaltkreis | LS30 | |
| D13 | Schaltkreis | LS00 | |
| D14 | Schaltkreis | LM358N | |
| D15 | Schaltkreis | LS04 | |
| D16 | Schaltkreis | UB857D / Z8430B | |
| D17 | Schaltkreis | LS10 | |
| JP1 | Steckbrücke | Stiftleiste, 3-polig | oder Reichelt: NK 236 |

| Name | Bauteil | Bezeichnung / Wert | Bemerkung |
|-------------------|------------------|---------------------------------|------------------------------|
| Kleinlautsprecher | Lautsprecher | 32 Ω , AC | z.B. Pollin: 641148 |
| QG1 | Quarz-Oszillator | 2MHz | |
| P1 | Potentiometer | 5 k Ω | Trimpoti, liegend, 6x6 mm |
| R1-R8 | Widerstand | 100 Ω | ggf. Bauform 0204 |
| R9, R10 | Widerstand | 1 k Ω | |
| R11 | Widerstand | 620 Ω | ggf. Bauform 0204 |
| R12 | Widerstand | 2,2 k Ω | ggf. Bauform 0204 |
| R13, R14 | Widerstand | 2,2 k Ω | |
| R15 | Widerstand | 1 k Ω | |
| R16, R17 | Widerstand | 2,2 k Ω | |
| R18, R19 | Widerstand | 10 k Ω | ggf. Bauform 0204 |
| R20 | Widerstand | 220 k Ω - 330 k Ω | Verstärkung für Lautsprecher |
| R21-R26 | Widerstand | 10 k Ω | |
| R27 | Widerstand | 390 k Ω | Verstärkung für LED |
| R28-R43 | Widerstand | 220 Ω | Bauform 0204, stehend |
| R44 | Widerstand | 220 Ω | |
| X1/LO | Klinkenbuchse | PG203J | Reichelt: EBS 35 BL |
| X2 | D-SUB-Buchse | 25-polige Buchse | Reichelt: D-SUB BU 25US |
| IC-Fassung | IC-Fassung | DIL08 | 1 Stück |
| IC-Fassung | IC-Fassung | DIL14 | 12 Stück |
| IC-Fassung | IC-Fassung | DIL20 | 2 Stück |
| IC-Fassung | IC-Fassung | DIL28-6 | 1 Stück |
| IC-Fassung | IC-Fassung | DIL40 | 1 Stück |
| Platine | Platine | Platine | |

10 Anlage C – Anwendersoftware

Zeitgleich mit der Veröffentlichung der Serienplatinen des Moduls M066 wird ein Softwarepaket publiziert, welches die in diesem Handbuch beschriebenen Programme beinhaltet und darüber hinausgehend weiteres Material zur eigenen Anwendungserstellung.

10.1 Softwarepakete

Das Archiv m066_software_demo_basics.zip (siehe Kapitel "[13.2 Links](#)") enthält zum Zeitpunkt der Drucklegung dieses Handbuchs alle nachfolgend aufgeführten Dateien. Anleitungen und Hinweise sind den entsprechenden Kapiteln dieses Handbuchs zu entnehmen.

| | |
|-----------------|---|
| DEMO.ZIP | zum Angucken und Anhören |
| QUELLEN_DEMO | Ordner mit Arbeitsmaterialien |
| AY.KCC | Demoprogramm M066 |
| MUSxx.KCS | 10 selbstabspielende STC-Musiken |
| ASM.ZIP | für Assembler-Arbeiten |
| QUELLEN_ASM | Ordner mit Arbeitsmaterialien |
| ASMTTEST.KCC | fertiges Demoprogramm mit Musik |
| BASI.ZIP | für Arbeiten mit KC-BASIC |
| QUELLEN_BASI | Ordner mit Arbeitsmaterialien |
| AYPLAY.SSS | Basic-Demoprogramm für CAOS 4.7 |
| AYPLAY42.SSS | Basic-Demoprogramm für CAOS 4.2 |
| AYPLAY42.PDF | Hinweise zur Anwendung unter CAOS 4.2 |
| BASC.ZIP | für Arbeiten mit dem JKCEMU-BASIC-COMPILER |
| QUELLEN_BASC | Ordner mit Arbeitsmaterialien |
| TEST.KCC | fertiges Demoprogramm mit Musik |

Im Text dieses Handbuchs genannte, aber hier nicht explizit aufgelistete Dateien (z.B. AYPLAY60.ASM) befinden sich in dem Ordner mit Arbeitsmaterialien des entsprechenden Softwarepaketes.

10.2 Hinweis zu den verwendeten Arbeitszellen

Alle vorgestellten Beispiele benötigen zahlreiche Arbeitszellen für das Abspielen der Musik. Standardmäßig wurde dafür der Bereich 0080h...00C7h vorgesehen.

Die bisherigen Tests ergaben damit keine Probleme oder Beeinträchtigungen des Abspielens. Der KC85/x mit seiner umfangreichen Ausstattung an Modulen und Software weist mittlerweile aber nur noch sehr wenig freie Ressourcen für Neues auf. Es scheint nahezu unmöglich, alle denkbaren Konstellationen und daran geknüpfte Bedingungen vorherzusehen und zu testen.

Im Einzelfall kann es daher nötig sein, den Arbeitszellenbereich zu verlagern, weil er mit anderen Programmen kollidiert. Ähnliches gilt für die Interrupt-Tabelle (0100h...0107h) des CTC.

Anzeichen für Unverträglichkeiten sind, daß die Musik unter bestimmten Bedingungen (z.B. andere gesteckte Module, parallel laufende Programme) nicht mehr korrekt, d.h. verzerrt, stotternd oder gar nicht abgespielt wird. Der in den Quelltexten mit WORK definierte Beginn der Arbeitszellen ist dann auf einen anderen freien Adreßbereich zu legen.

Sollte sich in kritischen Fällen gar kein freier Bereich finden lassen, so ist immer noch ein "Einbetten" der Arbeitszellen in das Assembler-Anwenderprogramm möglich.

```

ORG      0300h
         DEFM      7Fh, 7Fh, "NAME", 1      ;PROLOG
         JP        START                    ;SPRUNG ZUM PROGRAMMANFANG
WORK:    DEFS      50h, 0                    ;ARBEITSZELLENBEREICH
         ;80 Bytes Freiraum
START:   ;PROGRAMMBEGINN
         ; ...

```

Listing 11: Arbeitszellen direkt im Assembler-Anwenderprogramm

11 Anlage D – Musiktheorie

11.1 Übersicht der PSG-Register

| Reg.-Nr. | Funktion | Wertebereich | Anmerkungen |
|----------|---------------------------|-----------------------|---|
| 0 | Kanal A Tonhöhe fein | 8-bit (0-255) | einstellbar zwischen 30,5 Hz und 125 kHz abhängig von Taktfrequenz fein=grob=0 → Ton aus |
| 1 | Kanal A Tonhöhe grob | 4-bit (0-15) | |
| 2 | Kanal B Tonhöhe fein | 8-bit (0-255) | |
| 3 | Kanal B Tonhöhe grob | 4-bit (0-15) | |
| 4 | Kanal C Tonhöhe fein | 8-bit (0-255) | |
| 5 | Kanal C Tonhöhe grob | 4-bit (0-15) | |
| 6 | Tonhöhe Rauschen | 5-bit (0-31) | Frequenzspektrum des Rauschens |
| 7 | Mischer | 8-bit low=Freigabe | 7 6 5 4 3 2 1 0 |
| | | | B A C B A C B A |
| | | | IO-Port Rauschen Ton |
| 8 | Kanal A Lautstärke | 4-bit (0-15) | Wenn Bit 4 gesetzt (16), dann ist der Lautstärkewert ohne Einfluß und die Hüllkurve wird benutzt. |
| 9 | Kanal B Lautstärke | 4-bit (0-15) | |
| 10 | Kanal C Lautstärke | 4-bit (0-15) | |
| 11 | Hüllkurve - Dauer fein | 8-bit (0-255) | abhängig von Taktfrequenz! |
| 12 | Hüllkurve - Dauer grob | 8-bit (0-255) | |
| 13 | Hüllkurve - Form | 4-bit (0-15) | 3 2 1 0 |
| | | | 1= dauernd 0= einmalig 1= Anstieg 0= Abfall 1= Umkehr 0= Halten |
| 14 | I/O-Port A | 8-bit (0-255) | Richtung durch OUT-/IN-Operation bestimmt, für Eingabe ist Freigabe im Mischer nötig! |
| 15 | I/O-Port B | 8-bit (0-255) | |

Tabelle 9: Übersicht der PSG-Register

11.2 Registerwerte für einige Notenfrequenzen

Nachfolgend wird eine Übersicht der Registerwerte einiger Notenfrequenzen gegeben, welche bei einem 2-MHz-Takt bzw. einem 1,77-MHz-Takt (■) des PSG Verwendung finden können. Dabei gelten die Spalten zwei bis vier nur für den 2-MHz-Takt.

| Note | Soll-Freq. | Ist-Freq. | Diff. in % | Fein | | Grob | |
|------------------|------------|-----------|------------|------|-----|------|---|
| c ⁵ | 4186.010 | 4166.667 | -0.46 | 30 | 26 | 0 | 0 |
| h ⁴ | 3951.070 | 3906.250 | -1.13 | 32 | 28 | 0 | 0 |
| ais ⁴ | 3729.310 | 3676.471 | -1.42 | 34 | 30 | 0 | 0 |
| a ⁴ | 3520.000 | 3472.222 | -1.36 | 36 | 31 | 0 | 0 |
| gis ⁴ | 3322.440 | 3289.474 | -0.99 | 38 | 33 | 0 | 0 |
| g ⁴ | 3135.960 | 3125.000 | -0.35 | 40 | 35 | 0 | 0 |
| fis ⁴ | 2959.960 | 2976.190 | +0.55 | 42 | 37 | 0 | 0 |
| f ⁴ | 2793.830 | 2777.778 | -0.57 | 45 | 40 | 0 | 0 |
| e ⁴ | 2637.020 | 2659.574 | +0.86 | 47 | 42 | 0 | 0 |
| dis ⁴ | 2489.020 | 2500.000 | +0.44 | 50 | 44 | 0 | 0 |
| d ⁴ | 2349.320 | 2358.491 | +0.39 | 53 | 47 | 0 | 0 |
| cis ⁴ | 2217.460 | 2232.143 | +0.66 | 56 | 50 | 0 | 0 |
| c ⁴ | 2093.000 | 2083.333 | -0.46 | 60 | 53 | 0 | 0 |
| h ³ | 1975.530 | 1984.127 | +0.44 | 63 | 56 | 0 | 0 |
| ais ³ | 1864.660 | 1865.672 | +0.05 | 67 | 59 | 0 | 0 |
| a ³ | 1760.000 | 1760.563 | +0.03 | 71 | 63 | 0 | 0 |
| gis ³ | 1661.220 | 1666.667 | +0.33 | 75 | 67 | 0 | 0 |
| g ³ | 1567.980 | 1562.500 | -0.35 | 80 | 71 | 0 | 0 |
| fis ³ | 1479.980 | 1488.095 | +0.55 | 84 | 75 | 0 | 0 |
| f ³ | 1396.910 | 1404.494 | +0.54 | 89 | 79 | 0 | 0 |
| e ³ | 1318.510 | 1315.789 | -0.21 | 95 | 84 | 0 | 0 |
| dis ³ | 1244.510 | 1250.000 | +0.44 | 100 | 89 | 0 | 0 |
| d ³ | 1174.660 | 1179.245 | +0.39 | 106 | 94 | 0 | 0 |
| cis ³ | 1108.730 | 1106.195 | -0.23 | 113 | 100 | 0 | 0 |
| c ³ | 1046.500 | 1050.420 | +0.37 | 119 | 106 | 0 | 0 |
| h ² | 987.767 | 984.252 | -0.36 | 127 | 112 | 0 | 0 |
| ais ² | 932.328 | 932.836 | +0.05 | 134 | 119 | 0 | 0 |

| Note | Soll-Freq. | Ist-Freq. | Diff. in % | Fein | | Grob | |
|------------------|------------|-----------|------------|------|-----|------|---|
| a ² | 880.000 | 880.282 | +0.03 | 142 | 126 | 0 | 0 |
| gis ² | 830.609 | 833.333 | +0.33 | 150 | 133 | 0 | 0 |
| g ² | 783.991 | 786.164 | +0.28 | 159 | 141 | 0 | 0 |
| fis ² | 739.989 | 739.645 | -0.05 | 169 | 149 | 0 | 0 |
| f ² | 698.456 | 698.324 | -0.02 | 179 | 158 | 0 | 0 |
| e ² | 659.255 | 657.895 | -0.21 | 190 | 168 | 0 | 0 |
| dis ² | 622.254 | 621.891 | -0.06 | 201 | 178 | 0 | 0 |
| d ² | 587.330 | 586.854 | -0.08 | 213 | 188 | 0 | 0 |
| cis ² | 554.365 | 555.556 | +0.21 | 225 | 200 | 0 | 0 |
| c ² | 523.251 | 523.013 | -0.05 | 239 | 211 | 0 | 0 |
| h ¹ | 493.883 | 494.071 | +0.04 | 253 | 224 | 0 | 0 |
| ais ¹ | 466.164 | 466.418 | +0.05 | 12 | 237 | 1 | 0 |
| a ¹ | 440.000 | 440.141 | +0.03 | 28 | 251 | 1 | 0 |
| gis ¹ | 415.305 | 415.282 | -0.01 | 45 | 10 | 1 | 1 |
| g ¹ | 391.995 | 391.850 | -0.04 | 63 | 26 | 1 | 1 |
| fis ¹ | 369.994 | 369.822 | -0.05 | 82 | 43 | 1 | 1 |
| f ¹ | 349.228 | 349.162 | -0.02 | 102 | 61 | 1 | 1 |
| e ¹ | 329.628 | 329.815 | +0.06 | 123 | 80 | 1 | 1 |
| dis ¹ | 311.127 | 310.945 | -0.06 | 146 | 100 | 1 | 1 |
| d ¹ | 293.665 | 293.427 | -0.08 | 170 | 121 | 1 | 1 |
| cis ¹ | 277.183 | 277.162 | -0.01 | 195 | 143 | 1 | 1 |
| c ¹ | 261.626 | 261.506 | -0.05 | 222 | 167 | 1 | 1 |
| h | 246.942 | 247.036 | +0.04 | 250 | 192 | 1 | 1 |
| ais | 233.082 | 233.209 | +0.05 | 24 | 219 | 2 | 1 |
| a | 220.000 | 220.070 | +0.03 | 56 | 247 | 2 | 1 |
| gis | 207.652 | 207.641 | -0.01 | 90 | 21 | 2 | 2 |
| g | 195.998 | 195.925 | -0.04 | 126 | 52 | 2 | 2 |
| fis | 184.997 | 184.911 | -0.05 | 164 | 86 | 2 | 2 |
| f | 174.614 | 174.581 | -0.02 | 204 | 122 | 2 | 2 |
| e | 164.814 | 164.908 | +0.06 | 246 | 159 | 2 | 2 |
| dis | 155.563 | 155.473 | -0.06 | 36 | 199 | 3 | 2 |
| d | 146.832 | 146.886 | +0.04 | 83 | 241 | 3 | 2 |

| Note | Soll-Freq. | Ist-Freq. | Diff. in % | Fein | | Grob | |
|------------------|------------|-----------|------------|------|-----|------|----|
| cis | 138.591 | 138.581 | -0.01 | 134 | 30 | 3 | 3 |
| c | 130.813 | 130.753 | -0.05 | 188 | 78 | 3 | 3 |
| H | 123.471 | 123.518 | +0.04 | 244 | 128 | 3 | 3 |
| Ais | 116.541 | 116.496 | -0.04 | 49 | 181 | 4 | 3 |
| A | 110.000 | 110.035 | +0.03 | 112 | 238 | 4 | 3 |
| Gis | 103.826 | 103.821 | -0.01 | 180 | 41 | 4 | 4 |
| G | 97.999 | 97.962 | -0.04 | 252 | 105 | 4 | 4 |
| Fis | 92.499 | 92.524 | +0.03 | 71 | 172 | 5 | 4 |
| F | 87.307 | 87.291 | -0.02 | 152 | 243 | 5 | 4 |
| E | 82.407 | 82.399 | -0.01 | 237 | 62 | 5 | 5 |
| Dis | 77.782 | 77.785 | +0.00 | 71 | 142 | 6 | 5 |
| D | 73.416 | 73.400 | -0.02 | 167 | 227 | 6 | 5 |
| Cis | 69.296 | 69.290 | -0.01 | 12 | 60 | 7 | 6 |
| C | 65.406 | 65.411 | +0.01 | 119 | 155 | 7 | 6 |
| H ₁ | 61.735 | 61.728 | -0.01 | 233 | 0 | 7 | 7 |
| Ais ₁ | 58.271 | 58.275 | +0.01 | 97 | 106 | 8 | 7 |
| A ₁ | 55.000 | 54.993 | -0.01 | 225 | 219 | 8 | 7 |
| Gis ₁ | 51.913 | 51.910 | -0.01 | 104 | 83 | 9 | 8 |
| G ₁ | 48.999 | 49.000 | +0.00 | 247 | 210 | 9 | 8 |
| Fis ₁ | 46.249 | 46.245 | -0.01 | 143 | 88 | 10 | 9 |
| F ₁ | 43.654 | 43.660 | +0.02 | 47 | 230 | 11 | 9 |
| E ₁ | 41.203 | 41.200 | -0.01 | 218 | 125 | 11 | 10 |
| Dis ₁ | 38.891 | 38.892 | +0.00 | 142 | 28 | 12 | 11 |
| D ₁ | 36.708 | 36.711 | +0.01 | 77 | 198 | 13 | 11 |
| Cis ₁ | 34.648 | 34.645 | -0.01 | 24 | 121 | 14 | 12 |
| C ₁ | 32.703 | 32.705 | +0.01 | 238 | 55 | 14 | 13 |
| H ₂ | 30.868 | 30.864 | -0.01 | 210 | 0 | 15 | 14 |

Bei einem mit 2 MHz getakteten PSG ist es nicht möglich, die beiden Noten

| | | | | | | | |
|------------------|--------|--------|-------|-----|-----|----|----|
| Ais ₂ | 29.135 | 29.138 | +0.01 | 194 | 213 | 16 | 14 |
| A ₂ | 27.500 | 27.503 | +0.01 | 193 | 183 | 17 | 15 |

zu spielen, da nur 4 Bit für den Grobwert vorhanden sind.

BASIC-Programm zur Berechnung der o.a. Tabelle

```

10 REM NOTENTABELLE
20 CLS
30 LPRINT"N          SOLL-f      IST-f          %      FEIN GROB"
40 LPRINT"-----"
50 K=125000
60 FOR I=1 TO 88
70   READ N$,F1
80   PRINTN$,F1
90   LPRINTN$;" ";
100  TP=INT(0.5+K/F1) : G=INT(TP/256) : F=TP-256*G
130  LPRINTUSING "#####.###";F1;
140  F2=K/(G*256+F)
150  LPRINTTAB(17);
160  LPRINT USING "#####.###";F2;
170  LPRINTTAB(27);
180  PZ=100*((F2/F1)-1)
190  LPRINTUSING "+#.##";PZ; : LPRINTTAB(33);
210  LPRINTUSING "###";F; : LPRINTTAB(37);
230  LPRINTG
240  NEXT
250  DATA "c5      ",4186.01,"h4      ",3951.07,"ais4  ",3729.31
260  DATA "a4      ",3520.00,"gis4   ",3322.44,"g4     ",3135.96
270  DATA "fis4     ",2959.96,"f4     ",2793.83,"e4     ",2637.02
280  DATA "dis4     ",2489.02,"d4     ",2349.32,"cis4   ",2217.46
290  DATA "c4      ",2093.00,"h3     ",1975.53,"ais3   ",1864.66
300  DATA "a3      ",1760.00,"gis3   ",1661.22,"g3     ",1567.98
310  DATA "fis3     ",1479.98,"f3     ",1396.91,"e3     ",1318.51
320  DATA "dis3     ",1244.51,"d3     ",1174.66,"cis3   ",1108.73
330  DATA "c3      ",1046.50,"h2     ",987.767,"ais2   ",932.328
340  DATA "a2      ",880.000,"gis2   ",830.609,"g2     ",783.991
350  DATA "fis2     ",739.989,"f2     ",698.456,"e2     ",659.255
360  DATA "dis2     ",622.254,"d2     ",587.330,"cis2   ",554.365
370  DATA "c2      ",523.251,"h1     ",493.883,"ais1   ",466.164
380  DATA "a1      ",440.000,"gis1   ",415.305,"g1     ",391.995
390  DATA "fis1     ",369.994,"f1     ",349.228,"e1     ",329.628
400  DATA "dis1     ",311.127,"d1     ",293.665,"cis1   ",277.183
410  DATA "c1      ",261.626,"h      ",246.942,"ais    ",233.082
420  DATA "a      ",220.000,"gis    ",207.652,"h      ",195.998
430  DATA "fis     ",184.997,"f      ",174.614,"e      ",164.814
440  DATA "dis     ",155.563,"d      ",146.832,"cis    ",138.591
450  DATA "c      ",130.813,"H      ",123.471,"Ais   ",116.541
460  DATA "A      ",110.000,"Gis   ",103.826,"G      ",97.9989
470  DATA "Fis    ",92.4986,"F      ",87.3071,"E      ",82.4069
480  DATA "Dis    ",77.7817,"D      ",73.4162,"Cis   ",69.2957
490  DATA "C      ",65.4064,"H,,    ",61.7354,"Ais,, ",58.2705
500  DATA "A,,    ",55.0000,"Gis,, ",51.9130,"G,,    ",48.9995
510  DATA "Fis,,  ",46.2493,"F,,    ",43.6535,"E,,    ",41.2034
520  DATA "Dis,,  ",38.8909,"D,,    ",36.7081,"Cis,, ",34.6479
530  DATA "C,,    ",32.7032,"H,,,,  ",30.8677,"Ais,,,",29.13531
540  DATA "A,,,,  ",27.5000

```

12 Anlage E – Soundtracker-Dateiformat STC

Um eine Musik mit allen nötigen Informationen als Datei abzubilden, entstanden (je nach Rechner- und Tonanlage) diverse Formate. Hier soll das in den Demos benutzte STC-Format erläutert werden. Dabei handelt es sich um eine "compilierte" Form, d.h. die Register-Programmierwerte für den PSG (z.B. AY-3-8910) werden nicht direkt sondern komprimiert abgelegt. Das spart natürlich Platz (kürzere Datei für die Musik), erfordert aber ein "decompilieren" durch den Player. Übliche Musik in diesem Format hat eine Länge von ca. 2-4 KB.

12.1 Eckdaten und Grenzen des STC-Formats

Für den weiter oben bereits beschriebenen "Soundtracker" sind folgende Kennwerte definiert:

- Kanäle: 3
- Oktaven: 8
- Positionen: 1-256
- Anzahl Pattern: 1-32
- Positionen im Pattern: 1-64
- Samples: 15 (1-F, 0 hat Sonderfunktion)
- Ornamente: 15 (1-F, 0 hat Sonderfunktion)

12.2 Begriffe

Aus der Theorie der "Soundtracker" sind folgende Begriffe zu definieren

| | |
|----------|---|
| PATTERN | Ein PATTERN (Muster) ist die Abfolge von max. 64 Einträgen (Schritten), bestehend aus Noten, Samples und Effekten als "Notenliste". |
| POSITION | Mehrere Patterns bilden die Musik. Die POSITION enthält das aktuell gespielte Pattern in der Musik. Wird ein Pattern mehrfach benutzt (wiederholt), so erscheint es auch mehrfach in der Liste. |
| SAMPLE | Ein SAMPLE ist die Abfolge von Teiltönen (Segmenten), die beim Aufruf einer Note abgespielt wird. Es bestimmt den Grundklang (= "Instrument") der Note. Jedes Sample hat eine Nummer, über die es ausgewählt wird. In den 32 Elementen eines Samples ist der zeitliche Verlauf von Lautstärke, Rauschen und Tonhöhenänderung definiert. |
| ORNAMENT | Ornamente sind Zusatzeffekte, die (unabhängig vom Sample selbst) einen Tonhöhenversatz abbilden. Für jedes der 32 Elemente kann eine Verschiebung der Tonhöhe in Halbtönen gegenüber dem Normalwert angegeben werden. |

Tabelle 10: Begriffsdefinitionen für "Soundtracker"

12.3 Dateiaufbau

Der prinzipielle Aufbau einer jeden "Soundtracker"-Datei für den PSG gestaltet sich wie in der folgenden Tabelle dargestellt.

| | | | |
|-----------------------------|--------------|---|--------------------------|
| Kopfdaten | BYTE [1] | Tempo | |
| | WORD [1] | Zeiger auf Positionstabelle | Offsets zum Dateianfang. |
| | WORD [1] | Zeiger auf Ornamenttabelle | |
| | WORD [1] | Zeiger auf Patterntabelle | |
| | BYTE [18] | Kennung STC-Modul 'SONG BY ST COMPILE' | |
| | WORD [1] | Musiklänge | |
| Samples 1...16 x | BYTE [1] | Sample-Nummer | |
| | BYTE [3]*32 | Daten | |
| | BYTE [2] | Wiederholinfos | |
| Positionstabelle | BYTE [1] | Pattern-Anzahl p in der Musik (inkl. mehrfach benutzte) | |
| | BYTE [2] * p | Pattern-Nummer sowie Tonlage des jeweiligen Pattern | |
| Ornamenttabelle 1...16 x | BYTE [1] | Ornament-Nummer | |
| | BYTE [32] | Daten | |
| Patterntabelle 1... 32 x | BYTE [1] | Pattern-Nummer | |
| | WORD [3] | Zeiger auf Daten der Kanäle A, B, C | |
| | BYTE [1] | #FF als Ende-Kennung der Tabelle | |
| Patterndaten | BYTE [x] | #FF als Ende-Kennung jedes Patterns | |

Tabelle 11: Aufbau der "Soundtracker"-Dateien im STC-Dateiformat

12.3.1 Samples

Die Samples sind wie folgt definiert.

| | | |
|---------------------|--------------|----------------|
| Samples 1...16 x | BYTE [1] | Sample-Nummer |
| | 32x BYTE [3] | Daten |
| | BYTE [2] | Wiederholinfos |

Tabelle 12: Samples-Block in der "Soundtracker"-Datei

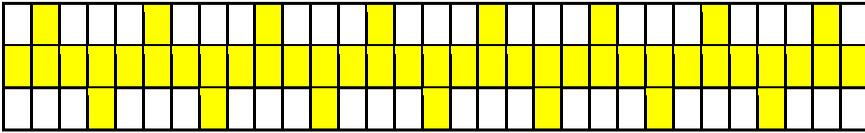


Abbildung 18: Beispiel "Ornament 7" aus "Jingle"

```
DEFB #00,#0C,#00,#F4,#00,#0C,#00,#F4,#00,#0C,#00,#F4,#00,#0C,#00,#F4
DEFB #00,#0C,#00,#F4,#00,#0C,#00,#F4,#00,#0C,#00,#F4,#00,#0C,#00,#F4
```

Listing 11: Daten das für "Ornament 7" aus "Jingle" in Assemblerkode

12.3.3 Pattern

Die im Pattern enthaltenen Binärwerte lassen sich wie folgt einteilen

| | |
|---------|--|
| #00-#5F | Notenwert in Halbtönen (00 = C ₁ = tiefster Ton) |
| #60-#6F | Samplenummer |
| #70-#7F | Ornamentnummer |
| #80 | Kommando "R" = Pause |
| #81 | Leere Note, beendet nur Schritt |
| #82 | Sperren von Hüllkurve und Ornament |
| #83-#8E | Hüllkurvenform (3 - 14, vgl. AY3-Register 13), nächstes Byte ist Hüllkurvendauer |
| #8F-#A0 | nicht benutzt (reserviert) |
| #A1-#FE | Geschwindigkeit (Pause zwischen den Noten) |
| #FF | Ende des Patterns |

Tabelle 15: Einteilung der Pattern-Binärwerte

Der ursprünglich als Kennung einer STC-Musik gedachte Standardtext "SONG BY ST COMPILE" kann auch durch eine beliebige andere Zeichenkette (z.B. Musikname) gleicher Länge ersetzt werden. Tracker und Player benutzen ihn nicht als Testkriterium einer STC-Datei.

12.3.4 Beispiel "JINGLE.STC"

Der nachfolgende Quellcode in Assembler stellt die vollständige Musik "Jingle" im STC-Format dar.

```

;OFFSET 0: Kopfdaten
DEFB #06 ;Tempwert (Globale Verzögerung in Interrupts)
DEFW #00E1 ;Zeiger auf Positionstabelle
DEFW #00E4 ;Zeiger auf Ornament-Tabelle
DEFB #0126 ;Zeiger auf Pattern-Tabelle

DEFM „,SONG BY ST COMPILE“,0 ;Kennung STC-Modul (kann auch
abweichen!)

DEFW #018D ;Song-(Datei-)Länge in Bytes

;Es folgen die Samples: 99 Bytes (max. 16 x)
DEFB #06 ;Sample-Nummer, dann 3x32 Byte Daten
DEFB #0F,#80,#00,#0E,#80,#00,#0D,#80,#00,#0C,#80,#00,#0C,#80,#00,#0C
DEFB #80,#00,#0C,#80,#00,#0B,#80,#00,#0B,#80,#00,#0B,#80,#00,#0B,#80
DEFB #00,#0A,#80,#00,#0A,#80,#00,#0A,#80,#00,#0A,#80,#00,#09,#80,#00
DEFB #09,#80,#00,#09,#80,#00,#09,#80,#00,#08,#80,#00,#08,#80,#00,#08
DEFB #80,#00,#08,#80,#00,#07,#80,#00,#07,#80,#00,#07,#80,#00,#07,#80
DEFB #00,#06,#80,#00,#06,#80,#00,#05,#80,#00,#03,#80,#00,#01,#80,#00
DEFB #00 ;Wiederhol-Position
DEFB #00 ;Wiederhol-Länge

DEFB #0E ;Sample-Nummer
DEFB #0A,#20,#00,#0D,#00,#01,#0F,#80,#02,#0E,#80,#02,#0D,#80,#01,#0C
DEFB #A0,#00,#0B,#A0,#01,#0A,#A0,#02,#0A,#A0,#02,#0A,#A0,#01,#0A,#80
DEFB #00,#0A,#80,#01,#0A,#80,#02,#0A,#80,#02,#0A,#80,#01,#0A,#A0,#00
DEFB #0A,#A0,#01,#0A,#A0,#02,#0A,#A0,#02,#0A,#A0,#01,#0A,#80,#00,#0A
DEFB #80,#01,#0A,#80,#02,#0A,#80,#02,#0A,#80,#01,#0A,#80,#00,#0A,#A0
DEFB #01,#0A,#A0,#02,#0A,#A0,#02,#0A,#A0,#01,#0A,#80,#00,#0A,#80,#00
DEFB #18 ;Wiederhol-Position
DEFB #07 ;Wiederhol-Länge

;Offset 00E1: Positionstabelle
DEFB #00 ;Anzahl Pattern im Song (-1)
DEFW #01 ;Pattern-Nummer für Position 1
DEFB #00 ;Tonlage des Pattern für diese Position

;Offset 00E4: Ornament-Tabelle, jeweils 33 Byte pro Ornament
DEFB #00 ;Ornament-Nummer, dann 32 Byte Daten
DEFB #00,#00,#00,#00,#00,#00,#00,#00,#00,#00,#00,#00,#00,#00,#00,#00
DEFB #00,#00,#00,#00,#00,#00,#00,#00,#00,#00,#00,#00,#00,#00,#00
DEFB #07 ;Ornament-Nummer
DEFB #00,#0C,#00,#F4,#00,#0C,#00,#F4,#00,#0C,#00,#F4,#00,#0C,#00,#F4
DEFB #00,#0C,#00,#F4,#00,#0C,#00,#F4,#00,#0C,#00,#F4,#00,#0C,#00,#F4

;OFFSET 0126: Patterntabelle, jeweils 7 Byte pro Pattern
DEFB #01 ;Pattern-Nummer
DEFW #012E ;Offset Pattern Kanal A
DEFW #014C ;Offset Pattern Kanal B
DEFW #0171 ;Offset Pattern Kanal C
DEFB #FF ;Ende Patterntabelle

```

```
;Offset 012E: Pattern Kanal A
DEFB #A2,#66,#77,#40,#A1,#41,#40,#A2,#3E,#A1,#40,#3E,#A2,#3C,#A1,#3E
DEFB #3C,#A2,#3B,#A1,#3C,#3B,#AA,#39,#A2,#70,#28,#AF,#28,#FF

;Offset 014C: Pattern Kanal B
DEFB #A2,#6E,#28,#A1,#29,#28,#A2,#26,#A1,#28,#26,#A2,#24,#A1,#26,#24
DEFB #A2,#23,#A1,#24,#23,#A2,#21,#A1,#27,#28,#29,#2A,#2B,#2C,#A2,#2D
DEFB #66,#1C,#AF,#15,#FF

;OFFSET 0171: Pattern Kanal C
DEFB #A2,#66,#34,#A1,#35,#34,#A2,#32,#A1,#34,#32,#A2,#30,#A1,#32
DEFB #30,#A2,#2F,#A1,#30,#2F,#AA,#2D,#A2,#2C,#AF,#2D,#FF
```

Listing 12: Vollständige Daten für "Jingle" im STC-Format in Assemblercode

13 Anlage F – Quellenangaben und Links

13.1 Quellen

Notenfrequenzen

https://de.wikipedia.org/wiki/Frequenzen_der_gleichstufigen_Stimmung

STC-Dateiformatbeschreibung

<http://www.avray.ru/формат-файла-stc-sound-tracker/>

AY-Player

МУЗЫКАЛЬНЫЙ ДВЕРНОЙ ЗВОНОК НА ЗВУКОВОМ СОПРОЦЕССОРЕ AY8910, ЖУРНАЛ 'РАДИО', 1998, №6, С. 42+43

<http://www.radioman-portal.ru/magazin/radio/1998/6.php>

Die darin enthaltene Abspielroutine stammt offenbar direkt vom ZX81 (mit AY-3-8910-Adapter), da sie fast identisch mit der unter folgendem Link zu Erreichenden ist (zxstplay.zip).

<http://www.sinclairzxworld.com/viewtopic.php?t=528&start=100>

13.2 Links

ASide-Assembler

<http://www.theeg.de/aside/>

Emulator JKCEMU

<http://www.jens-mueller.org/jkcemu/>

Neuentwicklung des M066

<http://kc-und-atari.idylle-am-waldrand.com/kc85/m066.html>

Bericht zur Entwicklung eines Soundmoduls – 1. KC-Club-Treffen

http://www.kcclub.de/index.php?option=com_content&view=article&id=76&Itemid=85

Softwarepaket M066 – Demonstration und Grundlagen

http://kc-und-atari.idylle-am-waldrand.com/download/kc855/m066_software_demo_basics.zip

oder

<http://susowa.homeftp.net/> -> Download -> CAOS-Software -> Musik und Demos

Softwarepaket M066 – STC-Player und KC-Tracker

http://kc-und-atari.idylle-am-waldrand.com/download/kc855/m066_software_stcplayer_kctracker.zip

oder

<http://susowa.homeftp.net/> -> Download -> CAOS-Software -> Musik und Demos

AY-Player und Sammlung von STC-Dateien

<http://www.avray.ru/avr-ay-player/>

http://www.avray.ru/music_collections/

14 Anlage G – KC85/x-Systeme im Test

Neben den schon beschriebenen funktionalen Tests des Klangmoduls wurden auch KC85/x-Systemkompatibilitätstests durchgeführt, um die Funktionsfähigkeit des M066 in diesen verschiedenen Systemen nachzuweisen.

Natürlich können nicht alle Systemgegebenheiten und Anwendungsfälle berücksichtigt werden, dafür ist die Gerätlandschaft rund um den KC85/x zu vielfältig.

Nachfolgend werden die Testsysteme aufgelistet.

14.1 KC85/2 (CAOS 2.2, 16kB Grund-RAM)

- Laden per Kassette/KC-Recorder (M052 geht nicht unter CAOS 2.2 bzw. am KC85/2 nur mit Modul M006=CAOS3.1)
 - AY-Demo und KCS (soweit Ende <4000h) funktionieren
 - Besonderheiten: das (normale) Pixelflimmern bei BS-Zugriffen während des Abspiels

14.2 KC85/3 (CAOS 3.1, 16kB Grund-RAM)

- Laden per USB mit M052 (zuvor "SWITCH 2 0" nicht vergessen)
 - AY-Demo und KCS (soweit Ende <4000h) funktionieren
 - Besonderheiten: das (normale) Pixelflimmern bei BS-Zugriffen während des Abspiels

14.3 KC85/4 (CAOS 4.2)

- Laden per USB mit M052
 - AY-Demo und KCS funktionieren

14.4 KC85/5 (CAOS 4.7)

- Laden per USB mit M052 (für CAOS 4.7 ist neue Firmware 2.4 oder 2.5 nötig)
 - AY-Demo und KCS funktionieren
 - KC-Tracker funktioniert
 - USB-Player funktioniert

14.5 M066 im KC85/4-Turm (CAOS 4.2)

Das Testsystem bestand aus den Geräten D001, D002, D004 und Diskettenlaufwerk. Es wurden die Module M003, M021, M025, M061, M051 und M052 eingesetzt bzw. eingeschaltet.

- Laden per USB mit M052 (%ULOAD bzw. %uload bei neuer Firmware 2.5)
 - AY-Demo und KCS funktionieren
- Laden per Diskette (CAOS-Betriebsart, %FLOAD)
 - AY-Demo und KCS funktionieren
 - Hinweis: Arbeitszellen eigener AY-Player dürfen 0000h bis 011Fh nicht benutzen, sonst wird FLOAD überschrieben
- Generelle Ergebnisse:
 - keine Probleme, egal welcher Steckplatz, welche Modulreihenfolge, welches Modul ein- oder ausgeschaltet ist
 - D004 eingeschaltet/ausgeschaltet bzw. mit/ohne CAOS-Diskette: kein Einfluß auf den Klang
 - KC-Tracker und USB-Player sind nicht lauffähig, da hierzu CAOS 4.7 erforderlich ist
- CP/M wurde nicht getestet, da die zum Zeitpunkt der Drucklegung dieses Handbuchs vorliegende M066-Software nur für die CAOS-Betriebsart entwickelt ist







mikroelektronik

RFT



KC-CLUB

Alte Computer by SRN
<http://www.kc-und-atari.de>