

Hardware

Sicherungen

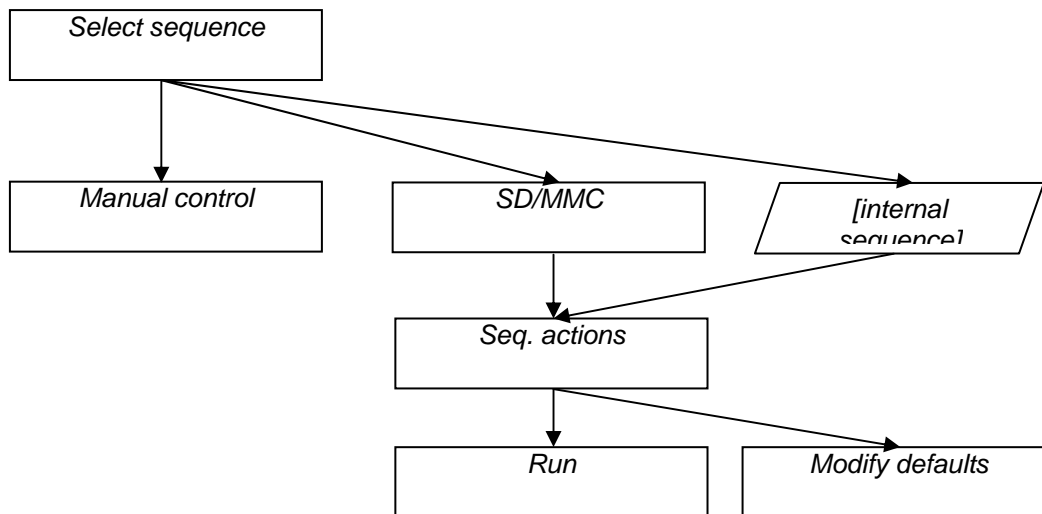
- Hauptsicherung für Last- und Steuerteil (Sicherungshalter unter Netzanschluss): T8A
- Nebensicherung für Steuerteil: T200mA

Anschlüsse

- Netzspannungsversorgung über Kaltgerätekabel.
- Die Gesamte Stromaufnahme der an die 230V Kaltgerätedosen angeschlossenen Verbraucher darf 8A nicht überschreiten.
- Die XLR-Stecker dürfen jeweils mit maximal 8A/48V (DC oder AC) beaufschlagt werden. (Das Parallelschalten mehrerer Kanäle ist wegen der nicht garantierbaren Stromaufteilung zu Unterlassen.)
- An der 9-poligen D-Sub Buchse sind die RS232 und ICSP (In Circuit Serial Programming) Anschlüsse herausgeführt. Da die Pinbelegung daher **nicht** der standard RS232 Belegung entspricht, darf keinesfalls ein RS232 Kabel angeschlossen werden. (Die D-Sub Buchse dient ausschließlich für Firmware updates.)



Bedienung Menüstruktur



Mit OK wird ein selektiertes Untermenü betreten, mit ESC wird es wieder Verlassen. Die Links- und Rechtstasten wechseln durch die Menüpunkte, mit den Auf-/Abtasten werden selektierte Werte verändert. (Am Besten ist es, man spielt ein wenig damit herum, das Menü sollte selbsterklärend sein.)

Hinweise zu SD/MMC

- Solange SD/MMC ausgewählt ist, sollte eine Speicherkarte weder eingesteckt noch entfernt werden. Wird der Menüpunkt SD/MMC ausgewählt ohne dass sich eine Speicherkarte im Slot befindet, so kann es unter Umständen länger dauern, bis die Aktion durch ein Timeout abgebrochen wird (≥ 30 s).
- Es können nur "normale" (keine SDHC) Speicherkarten verwendet werden.
- Im SD/MMC-Menü sieht man den Inhalt der Speicherkarte. Verzeichnisse sind rechts unten mit "DIR" gekennzeichnet und können durch drücken von OK betreten werden. Ein Verzeichnis wird durch Selektieren und Auswählen des ".." Eintrages verlassen, wodurch man wieder in das übergeordnete Verzeichnis gelangt.
- Wird eine Datei, welche kein passender Bytecode für den Sequencer ist, ausgewählt, so führt das zu einem "Internal error", welcher sich nur durch einen Reset (Ein- und Ausschalten) beheben lässt.

Programmierung

Der Sequencer wird in einer assemblerähnlichen, rein linearen (keine Verzweigungen und Sprungbefehle), Sprache programmiert. Der Quelltext kann mit einem beliebigen Texteditor (Notepad, vim, ... aber nicht Word) geschrieben werden. Danach wird dieser durch den Assembler in einen platzsparenden Bytecode assembliert, der vom Sequencer interpretiert wird.

Eine Quelltext-Datei enthält immer zwei Abschnitte:

- Header: In diesem werden allgemeine Einstellungen, die für die ganze Sequenz gültig sind, getroffen. Dazu gehören die Ausführungsgeschwindigkeit, Kanalzuteilung,
- Sequenz: Die Sequenz selbst. Sie bestimmt, wann welche Kanäle geschaltet werden.

Zeitbasis

Bevor wir zu der Beschreibung der einzelnen Befehle kommen, soll die Generierung der internen Zeitbasis kurz erläutert werden.

Die Grundzeiteinheit des Sequencers beträgt 10ms (=1/100s). Da aber ein Wartezyklus maximal ca. 16000 Zeiteinheiten lang sein kann, könnte ein Wartezyklus maximal ca. 160s lang sein. Darüber hinaus könnte auch die Ausführungsgeschwindigkeit der Sequenz nicht geändert werden während eine Zeitauflösung von 10ms nur in den seltensten Fällen notwendig ist.

Deshalb wird aus der Grundzeiteinheit eine langsamere Zeiteinheit erzeugt. Aus wie vielen Basistakten ein Sequenztakt besteht wird durch `subtick_default` festgelegt. Wird z.B. `subtick_default` auf 100 gesetzt, so beträgt der Sequenztakt 1s. Durch Änderung des Sequenztaktes kann nun auch die Ausführungsgeschwindigkeit komfortabel variiert werden, da die Berechnung aller Wartezyklen auf Sequenztakten basiert.

Tracks/Channels

Der Sequencer verfügt intern über 8 "virtuelle Spuren" (Tracks) die in der Programmierung mit den Zahlen 0 bis 7 bezeichnet werden. Diese Tracks können einem beliebigen der 8 Ausgangskanäle (Channels) zugewiesen werden. Dadurch können z.B. leicht 2 Kanäle vertauscht, etc. werden. Außerdem kann eine Zufallszuteilung der Tracks zu den Channels eingestellt werden, sodass bei jedem erneuten Sequenzstart die entsprechenden Kanäle zufällig zugeteilt werden.

Headerbefehle

Theoretisch ist es nicht zwingend notwendig, dass im Header Einstellungen getroffen werden. Alle Einstellungen die im Header nicht festgelegt sind, werden vom Assembler mit (mehr oder weniger) sinnvollen Voreinstellungen belegt. Da diese aber auch zwischen Assembler-Versionen abweichen können und weil es übersichtlicher ist, wird empfohlen das Meiste selbst festzulegen.

Einige Einstellungen des Headers können auch nach der Assemblierung auf dem Sequencer temporär (für den aktuellen Durchlauf der Sequenz) im "Modify defaults"-Menü modifiziert werden. Diese geänderten Einstellungen werden vom Sequencer aber nicht gespeichert, sodass sie bei einer erneuten Auswahl der Sequenz wieder auf die im Bytecode festgelegten Standardeinstellungen zurückgesetzt werden.

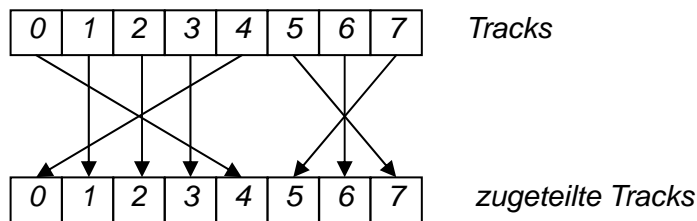
Liste von gültigen Headerbefehlen (ein * kennzeichnet, dass diese Einstellung auf dem Sequencer nachträglich geändert werden kann):

description

Eine Kurzbeschreibung der Sequenz. Sie darf maximal 16 Zeichen lang sein und keine Leerzeichen enthalten. Diese Beschreibung wird bei der Ausführung der Sequenz in der ersten Zeile des Displays angezeigt.

track_mix*

Alle nach track_mix eingetragenen Channels werden zufällig zugeteilt. track_mix 0457 bewirkt z.B. (Die genaue Zuteilung der Tracks 0, 4, 5 und 7 ist zufällig.)



Channels

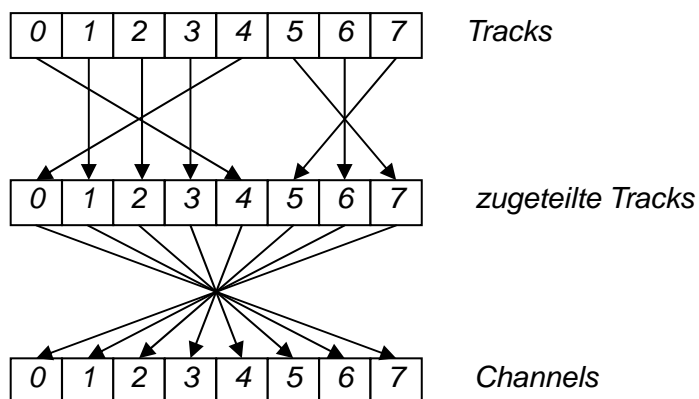
track_mapping*

Mit Hilfe dieser Einstellung können die zugeteilten Tracks auf einen beliebigen Ausgangskanal gelegt werden. Es müssen immer die 8 Spurzahlen angegeben werden. Jede Spurzahl kann mehr als einmal vorkommen, sodass ein Track auf mehrere Channels gelegt werden kann. Die Stelle in der Zahlenreihe entspricht der Spur und die Zahl selbst dem Kanal.

track_mapping 01234567 bewirkt z.B. dass Track-0 auf Channel-0 ausgegeben wird, Track-1 auf Channel-1 usw.

track_mapping 76543210 hingegen würde die Reihenfolge genau umdrehen. Track-0 würde auf Channel-7 ausgegeben, Track-1 auf Channel-6, Track-2 auf Channel-5 usw.

Die Zuteilung dieses Mappings erfolgt nach der zufälligen Verwürfelung durch track_mix. Z.B. bewirkt track_mix 0457 in Kombination mit track_mapping 76543210 (Wieder ist die genaue Zuteilung der Tracks 0, 4, 5 und 7 zufällig):

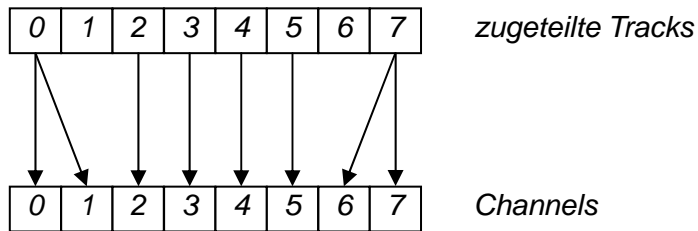


Bsp: Ergebnis von track_mapping 00234577 (Der zugeteilte Track 0 wird auf Channel 0 und 1 ausgegeben während der Track 7 auch Channel 6 und 7 ausgegeben wird. Die Tracks 1 und 6 werden überhaupt nicht ausgegeben, da sie im Mapping nicht vorkommen.)

Wörtlich entspricht track_mapping 00234577 folgendem:

Lege Track 0 auf Channel 0, Lege Track 0 auf Channel 1, Lege Track 2 auf Channel 3, Lege Track 3 auf Channel4, usw.

Wie oben beschrieben, ist der Channel, auf den der Track gelegt wird also nur implizit (d.h. er wird nicht direkt angegeben) festgelegt. Der Zielchannel entspricht wie oben beschrieben der Stelle in der Zahlenfolge.



subtick_default*

Gibt an, aus wie vielen Grundzeiteinheiten ein Sequenztakt besteht.

track_randomness*

Gibt den Zufallsstrebereich für die Wartezyklen an (also wie "zufällig" sich die Spuren verhalten). Der Wertebereich beträgt 0 bis 100 (Einheit: %), die Zufallsverteilung ist Gleichverteilung.

Wird z.B. ein Wartezyklus von 10 Einheiten programmiert und ist track_randomness auf 50% eingestellt, so wird die Wartezeit zufällig zwischen 5 und 15 Einheiten gewählt (also $10 \pm 50\%$).

start_delay*

Gibt die Anzahl der Sequenztakten an, bevor die eigentliche Sequenz startet.

start_randomness*

Gibt den Zufallsstrebereich für die Startzeit an. (Details: Siehe track_randomness).

Folgendes ergibt z.B. eine Startverzögerung der Sequenz von 5 Sequenztakten, wobei die Verzögerung zwischen 4 und 6 Sequenztakten zufällig gewählt wird (entspricht 20%):

```
start_delay 5
start_randomness 20
```

@sequence

Kennzeichnet das Ende des Headers und ist der Einzige Befehl, der in jedem Quelltext vorkommen muss, da er Header- und Sequenzteil trennt.

Sequenzbefehle

Sequenzbefehle bewirken im Gegensatz zu den Headerbefehlen die Zustandsänderungen der Tracks und somit der Ausgänge.

on

Schaltet die angegebenen Tracks ein. Bsp: on 017 schaltet die Tracks 0, 1 und 7 ein.

off

Schaltet die angegebenen Tracks aus. Bsp: off 35 schaltet die Tracks 3 und 5 aus.

toggle

Toggelt die angegebenen Tracks (war der Track vorher ausgeschaltet, so wird er eingeschaltet und umgekehrt). Bsp: toggle 0123 toggelt die ersten 4 Tracks.

wait

Wartet eine angegebene Anzahl an Sequenztakten. Bsp: wait 15 wartet 15 Sequenztakten

beep

Gibt einen Signalton aus.

repeat

Bewirkt, dass die Sequenzausführung abgebrochen und neu von vorne (inkl. Startverzögerung) begonnen wird. Dieser Befehl macht also nur Sinn, wenn er der letzte ist (statt eos).

display

Als Parameter kann eine bis zu 16 Zeichen lange Zeichenkette (die aber keine Leerzeichen enthalten darf) angegeben werden. Diese wird bei der Ausführung der Sequenz bei Erreichen des entsprechenden display-Befehls in der 2. Zeile des Displays angezeigt.

pause

Beim Erreichen des pause-Befehls wird die Ausführung der Sequenz angehalten. Durch Drücken der OK-Taste kann die Ausführung fortgesetzt werden, durch Drücken von ESC wird die Ausführung abgebrochen.

eos

(End Of Stream) Optionaler Befehl der am Ende der Sequenz stehen kann. Wird er weggelassen, so fügt ihn der Assembler automatisch an. Steht er in der Mitte einer Sequenz, so werden alle Befehle nach dem eos-Befehl ignoriert. (Somit kann eos dazu benutzt werden den hinteren Teil einer Sequenz abzuschalten.)

Kommentare und Dateinamen

In einen Quelltext können beliebig viele Leerzeilen eingefügt werden, sie werden vom Assembler ignoriert. Gleiches gilt für Leerzeichen und Tabulatoren.

Ein Kommentar beginnt mit einem Hash (#). Dieses kann entweder am Anfang einer Zeile stehen, wodurch die ganze Zeile zu einer Kommentarzeile erklärt wird, oder nach einem Befehl, wodurch einzelne Befehle kommentiert werden können.

Der Dateiname (inklusive Endung) ist sowohl für die Quelltexte als auch für die Bytecode-Dateien vollkommen frei wählbar. Für die Länge der Dateinamen wird allerdings empfohlen, sich an die alte MSDOS 8.3 Konvention zu halten (maximal 8 Zeichen vor dem '.' und maximal 3 Zeichen nach dem '.' in Dateinamen), da lange Dateinamen vom Sequencer nicht korrekt angezeigt werden können.

Assemblerlauf

Nachdem die Sequenz im Texteditor fertig verfasst wurde, muss sie in ein für den Sequencer ausführbares Format gebracht werden, was durch den Assembler erledigt wird.

Die Assemblierung kann entweder auf der Kommandozeile durchgeführt werden, indem folgendes ausgeführt wird:

```
java seqasm in.seq out.bin
```

Wobei *in.seq* der Quelltext und *out.bin* die zu erzeugende Bytecodedatei ist.

Alternativ kann die miniIDE benutzt werden, die z.B. durch einen Doppelklick auf miniIDE.bat gestartet werden kann.

Mit File=>Open input file muss zuerst eine Quelltextdatei ausgewählt werden. Danach wird mit File=>Set output file die Ausgabedatei festgelegt. Anschließend kann mit einem Klick auf den ASM-Button die Datei assembliert werden. Eventuelle Fehlermeldungen werden in dem mittigen Textfenster angezeigt.

Nachdem der Quelltext ohne Fehler in den Bytecode übersetzt wurde, kann dieser auf eine SD/MMC Speicherkarte kopiert und im Sequencer ausgeführt werden.

Beispiel: C.6A.5 Police Pistol 1 (PP1) aus BDMP-Handbuch

```
start_delay 5      # 10 Sekunden Startverzögerung
subtick_default 200 # Sequenztakt 2 Sekunden
track_randomness 0 # Keine zufälligen Zeitintervalle (0% randomness)
description C.6A.5
```

```
@sequence # Beginn der Sequenz
display Part_1
on 0
wait 60      # 120 Sekunden warten (1/100s*200*60=120s)
off 0
```

```
display Part_2
pause
```

```
on 0      # 1: Track 0 on
wait 1
off 0     # Track 0 off
wait 1
on 0     # 2
wait 1
off 0
wait 1
```

Sequencer Handbuch

```
on 0      # 3
wait 1
off 0
wait 1
on 0      # 4
wait 1
off 0
wait 1
on 0      # 5
wait 1
off 0
wait 1
on 0      # 6
wait 1
off 0
```

```
display Part_3
pause
```

```
on 0      # 1
wait 1
off 0
wait 1
on 0      # 2
wait 1
off 0
wait 1
on 0      # 3
wait 1
off 0
```