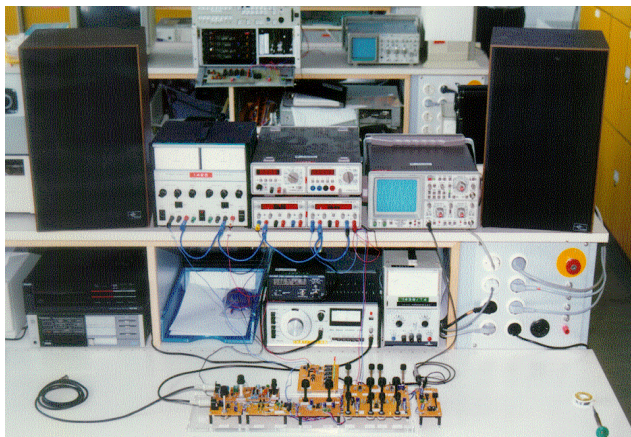


# Audiomischpult

Betreuer: Gottfried Patels

Oliver Singer



## ALLGEMEINES

Ein Mischpult ist die zentrale Einheit eines jeden Studios oder einer Bühnenanlage, und es hat verschiedene Aufgaben zu erfüllen. Die von den einzelnen Tonquellen (z.B. Mikrofonen, Bandgerät, CD-Spieler, DAT-Recorder,...) kommenden Signale werden zuerst auf einen einheitlichen Arbeitspegel gebracht. Diese Tonsignale können dann klanglich beeinflusst, in ihrer Lautstärke aufeinander abgestimmt und rückwirkungsfrei zusammengemischt werden. Es ist auch eine Verteilung auf verschiedene Leitungswege und eine Zumischung von Effekten möglich. Neben der Beeinflussung des Pegels und der Klangfarbe ist auch die Beeinflussung der Richtungszuordnung von grundlegender Bedeutung. Schließlich werden die bearbeiteten Signale auf einem Tonband oder anderen Tonträgern aufgezeichnet, oder aber einer Endverstärkerstufe zugeführt. Damit alle gestalterischen Forderungen erfüllt werden können, sind zum Teil sehr umfangreiche Mischpulte erforderlich.

## PROJEKTBE SCHREIBUNG

Geplant ist ein analoger Tonregietisch, der qualitativ hochwertigen Studiogeräten entspricht.

Für die Erfüllung aller Aufgaben sind einzelne Funktionsgruppen zuständig, die wiederum aus mehreren Baugruppen bestehen.

Das Mischpult ist in Streifentechnik aufgebaut, d.h. die Funktionsgruppen sind mechanisch abgegrenzte, austauschbare Einheiten. Daher ist das Mischpult fast beliebig erweiterbar.

Die einzelnen Funktionsgruppen werden nun näher erklärt:

### 1) Eingangskanäle

Hier werden die von den einzelnen Tonquellen (z.B. Mikrofon, CD-Spieler, Bandgerät, DAT-Recorder) kommenden Signale auf einen einheitlichen Arbeitspegel gebracht. Der symmetrische Mikrophoneingang umfaßt einen genügend großen Regelbereich, um sowohl dynamische, als auch Kondensatormikrophone anschließen zu können. Eine Phantomspeisung für Kondensatormikrophone ist genauso vorgesehen, wie eine Phasenumkehr, mehrere Meßpunkte für die Peak-Detection, ein Insert-Punkt (um Zusatzgeräte einschleifen zu können) und vier AUX-Abzweige. Mit den eingebauten Filtern und dem Equalizer kann das Tonsignal klanglich, und mit dem Kanal-Fader in der Lautstärke beeinflusst werden. Das Panorama-Potentiometer sorgt für die räumliche Verteilung des Signals (zur Erzeugung eines Stereo-Effektes). Eine Zuordnung zu den verschiedenen Summenkanälen ist ebenfalls vorgesehen.

### 2) Mastermodul

In dieser Funktionsgruppe erfolgt eine rückwirkungsfreie Summierung der Eingangskanäle, und die Beeinflussung der Lautstärke des Gesamtsignals. Der Ausgang des Mischpultes liegt ebenfalls hier.

### 3) Subgruppen

Bei diesen Einheiten handelt es sich um einfache Zwischensummen. Man kann hier die Signale betreffender Eingangskanäle zusammenfassen und dem Mischpult entnehmen (z.B. für Mehrspuraufnahmen), oder aber an das Mastermodul weiterleiten.

### 4) Abhörmodul

Diese Funktionsgruppe enthält im wesentlichen einen Kopfhörerverstärker und einen Wahlschalter, mit dem man die verschiedenen Tonsignale an den unterschiedlichsten Stellen des Pultes abhören kann.

### 5) AUX-Mastermodule

Die abgezweigten Signale der einzelnen Eingangskanäle werden hier summiert, und können dem Mischpult entnommen werden (um z.B. Effektgeräte anzusteuern).

### 6) Meter-Brücke

Hier befinden sich die Aussteuerungsanzeigen zur Pegelüberwachung der Eingangskanäle, der Subgruppen und des Mastermoduls. □

## Real software engineers don't ...

- \* Real Software Engineers Don't Read Dumps
- \* Real Software Engineers don't read dumps. They never generate them, and on the rare occasions that they come across them, they are vaguely amused.
- \* Real Software Engineers don't comment their code. The identifiers are so mnemonic they don't have to.
- \* Real Software Engineers don't write applications programs, they implement algorithms. If someone has an application that the algorithm might help with, that's nice. Don't ask them to write the user interface, though.
- \* Real Software Engineers eat quiche.
- \* If it doesn't have recursive function calls, Real Software Engineers don't program in it.
- \* Real Software Engineers don't program in assembler. They become queasy at the very thought.
- \* Real Software Engineers don't debug programs, they verify correctness. This process doesn't necessarily involve executing anything on a computer, except perhaps a Correctness Verification Aid package.
- \* Real Software Engineers like C's structured constructs, but they are suspicious of it because they have heard that it lets you get "close to the machine."
- \* Real Software Engineers play tennis. In general, they don't like any sport that involves getting hot and sweaty and gross when out of range of a shower. (Thus moun-

tain climbing is Right Out). They will occasionally wear their tennis togs to work, but only on very sunny days.

- \* Real Software Engineers admire PASCAL for its discipline and Spartan purity, but they find it difficult to actually program in. They don't tell this to their friends, because they are afraid it means they are somehow Unworthy.
- \* Real Software Engineers work from 9 to 5, because that is the way the job is described in the formal spec. Working late would feel like using an undocumented external procedure.
- \* Real Software Engineers write in languages that have not actually been implemented for any machine, and for which only the formal spec (in BNF) is available. This keeps them from having to take any machine dependencies into account. Machine dependencies make Real Software Engineers very uneasy.
- \* Real Software Engineers don't write in ADA, because the standards bodies have not quite decided on a formal spec yet.
- \* Real Software Engineers like writing their own compilers, preferably in PROLOG (they also like writing them in unimplemented languages, but it turns out to be difficult to actually RUN these).
- \* Real Software Engineers regret the existence of COBOL, FORTRAN and BASIC: PL/1 is getting there, but it is

not nearly disciplined enough; far too much built-in function.

- \* Real Software Engineers aren't too happy about the existence of users, either. Users always seem to have the wrong idea about what the implementation and verification of algorithms is all about.
- \* Real Software Engineers don't like the idea of some inexplicable and greasy hardware several aisles away that may stop working at any moment. They have a great distrust of hardware people, and wish that systems could be virtual at ALL levels. They would like personal computers (you know no one's going to trip over something and kill your DFA in mid-transit), except that they need 8 megabytes to run their Correctness Verification Aid packages.
- \* Real Software Engineers think better while playing WFF 'N' PROOF.