



Spracherkennung mit dem Z9001



Kommerziell hergestelltes Modul für den
Kleincomputer Z9001
zur
Echtzeitspracherkennung



Spracheingabemodul





Spracheingabemodul



- Entwickelt 1985/1986
- Vorgestellt Leipziger Frühjahrsmesse 1987
gemeinsam mit dem KC87
- Erhältlich 1987
- Produzierte Stückzahl ???



Entwickler



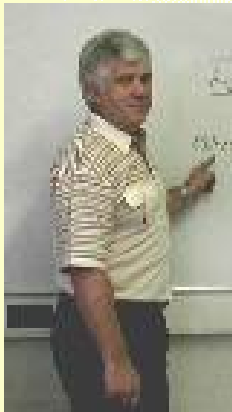
Dr.-Ing. Lothar Seveke

... „Ich war auch Entwickler beim ZFTM (Sprach-Ein- und Ausgabe) und ab und zu mal an den Z9001 und später an Dr. Kleinmichel ausgeliehen. Das Kassetten-Programm (Kansas-City-Format) und vor allem die Entwicklung des Spracheingabe-Moduls sind auf meinem Mist gewachsen“ ...



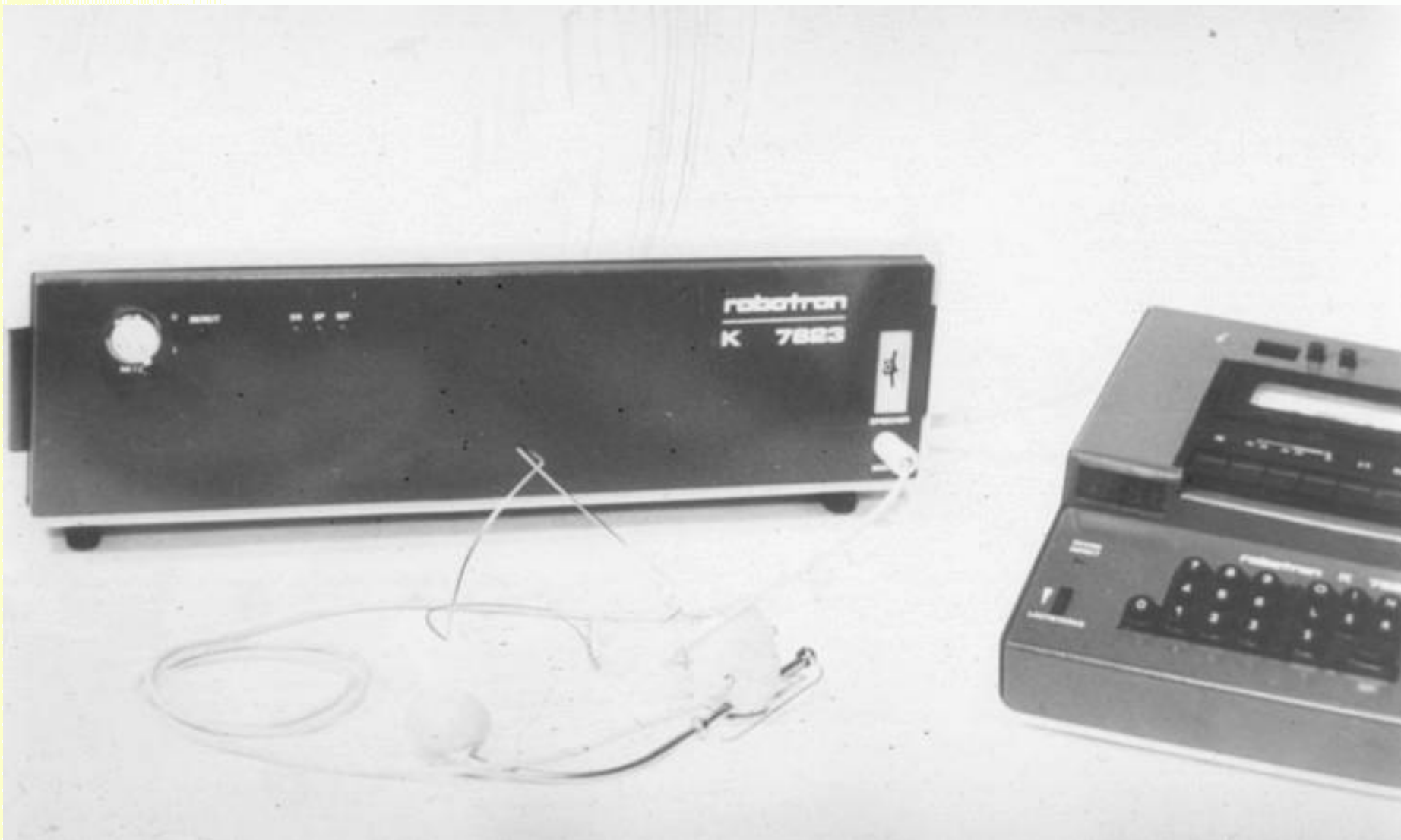
Dr.-Ing. Ulrich Kordon

... „der Spracheingabemodul für den Z9001 war damals eine Gemeinschaftsentwicklung des Kombinars Robotron und der TU Dresden. Die Arbeiten erfolgten im Auftrag des damaligen VEB Messelektronik Dresden als eigentlicher Hersteller des Z9001 (Themenleiter Z9001: Dr. Gunter Kleinmichel). Wir als TU waren nur für die Hardware des Spracheingabemoduls zuständig, die gesamte Software dazu wurde von Robotron entwickelt und basierte auf dem (kommerziellen) Spracheingabemodul für das K1520-System K7821 (Basis U880/Z80, Einbau mit Bedieneinheit in den Einzelwort-Spracherkenner K7823). Der damalige Bearbeiter bei Robotron war Dr. Seveke, der ursprünglich aus unserer TU-Arbeitsgruppe kam“ ...



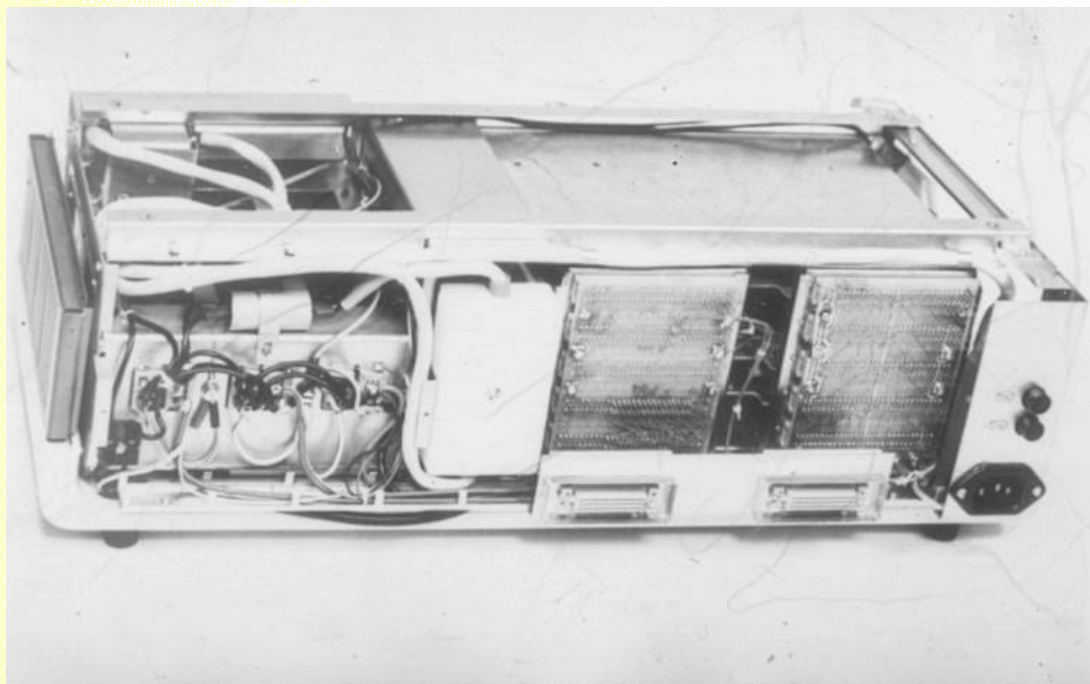


System ESE K7823





System ESE K7823



ese K7823 (oben)
bde K7822 (rechts)

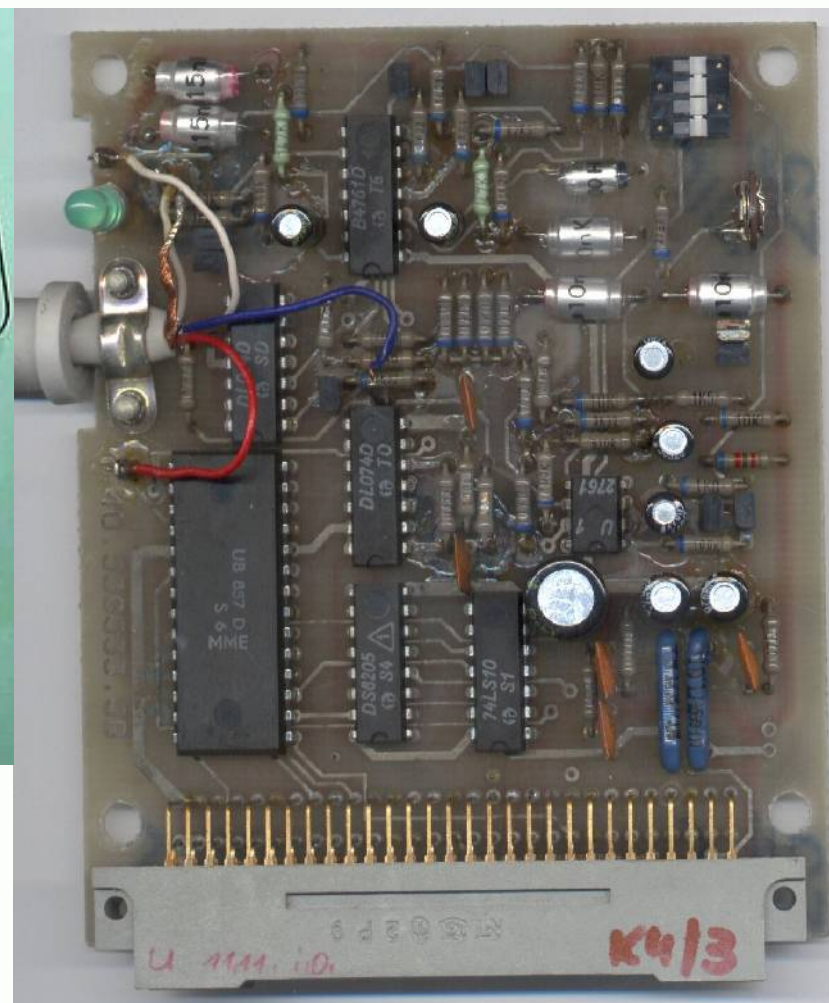




Spracheingabemodul

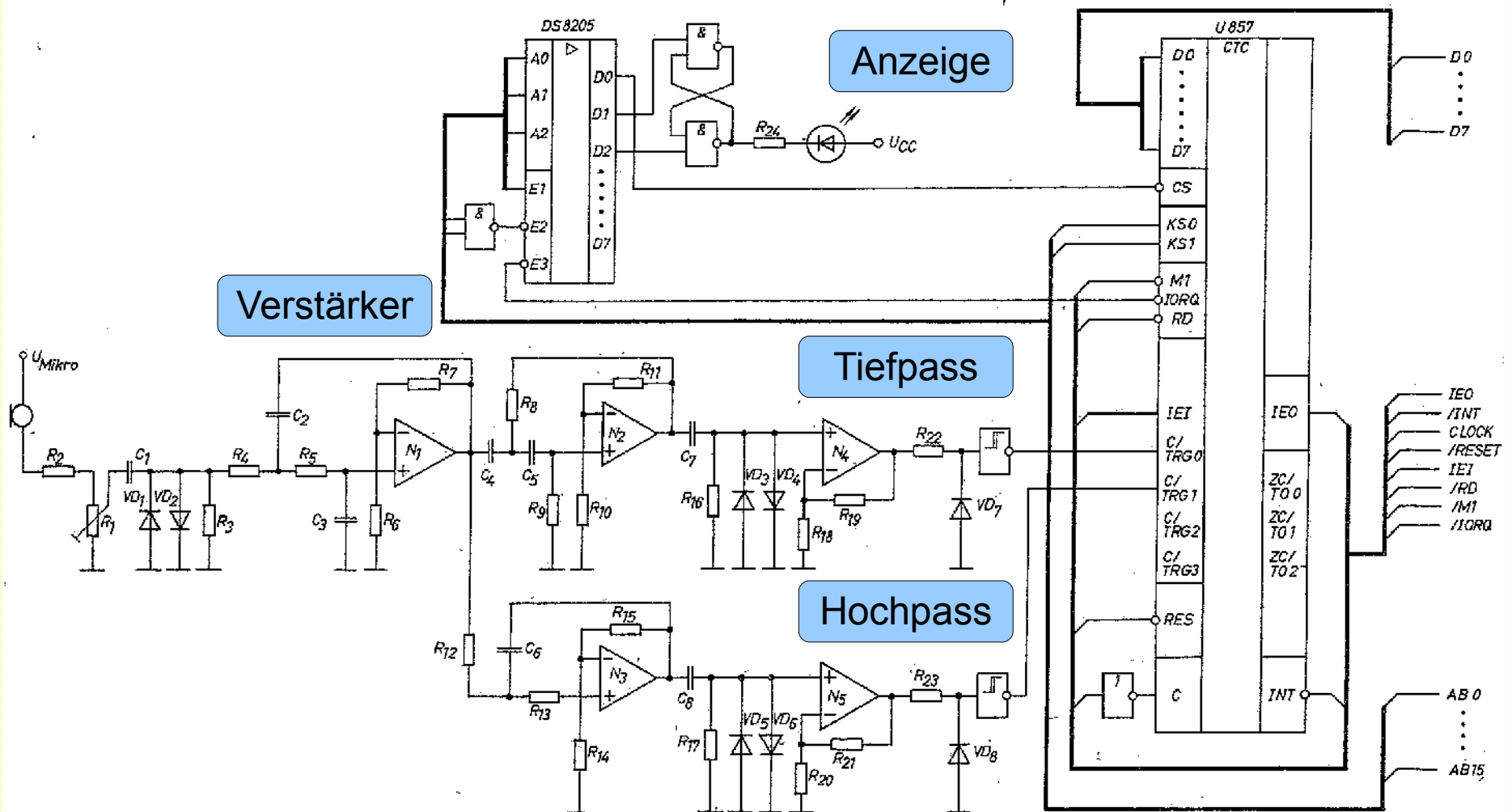


Sprechzeug SP75 (oben)
geöffnetes Modul (rechts)





Schaltplan





Eigenschaften



- Wortschatzgröße 50 Wortklassen
- Max. 200 Referenzmuster
- Wortlänge 0,2 bis 1,8 s
- Reaktionszeit nach Wortende: 0,2 s
- Pause zwischen 2 Worten: $> 0,2$ s
- Erkennungsquote f. geübten Sprecher $> 90\%$
- Rückweisung: $\sim 5\%$
- Fehlererkennung: $\sim 5\%$



Spracherkennung

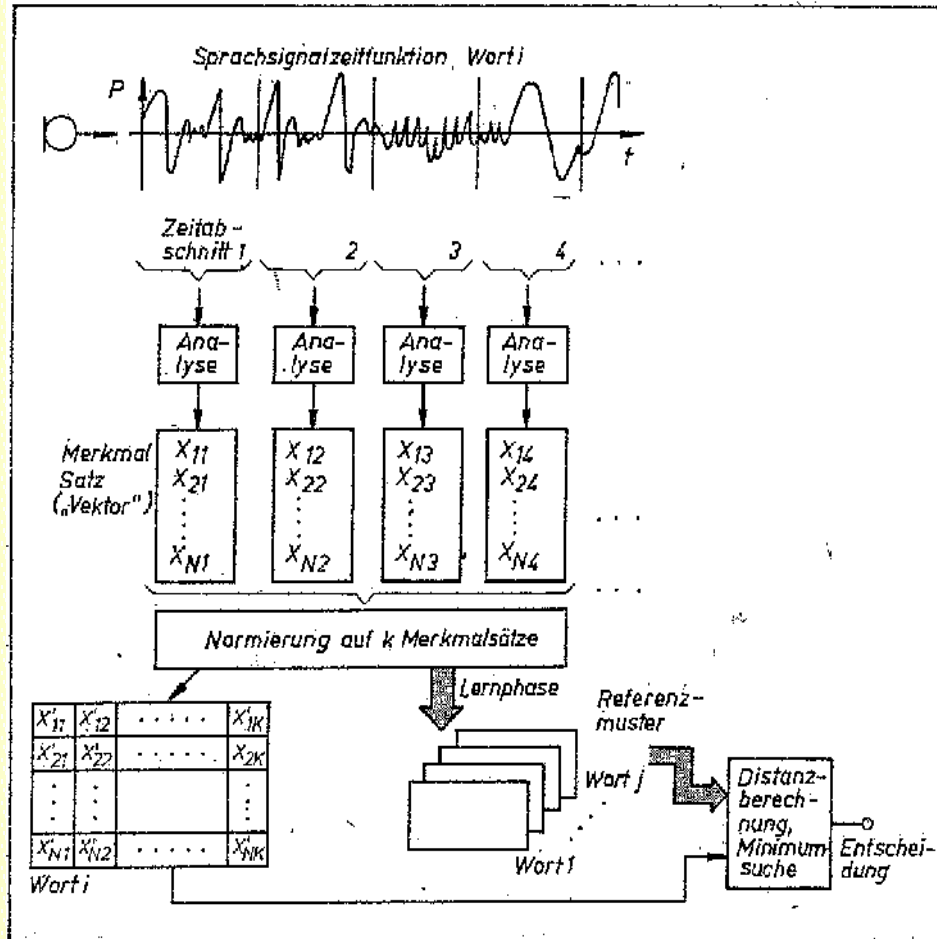


Bild 6: Prinzip einfacher Spracherkennung

1. Umwandlung des Audiosignals in eine Folge von Merkmalsvektoren
2. Normierung
3. Vergleich mit vorher ermittelten Referenzmustern



Das NDG

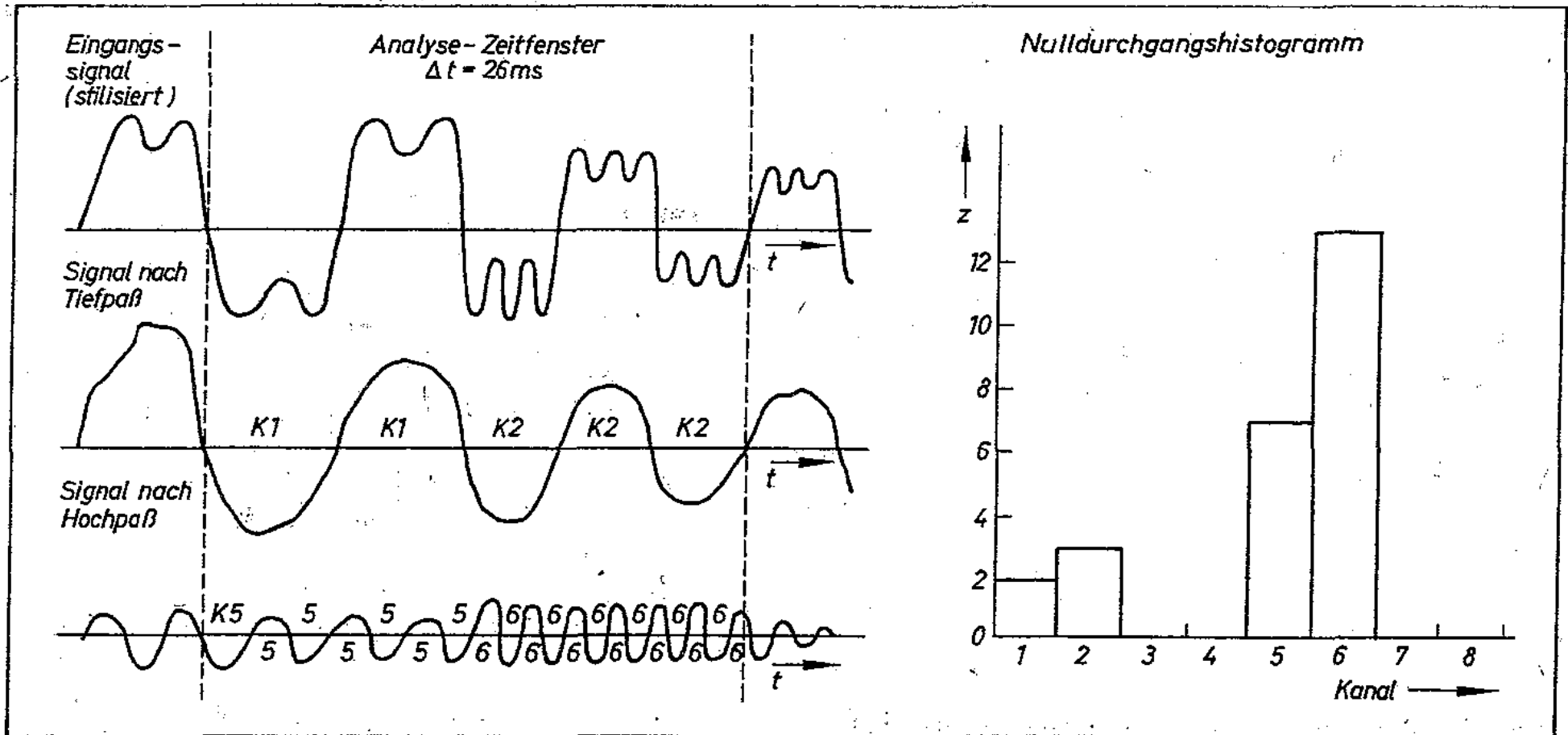
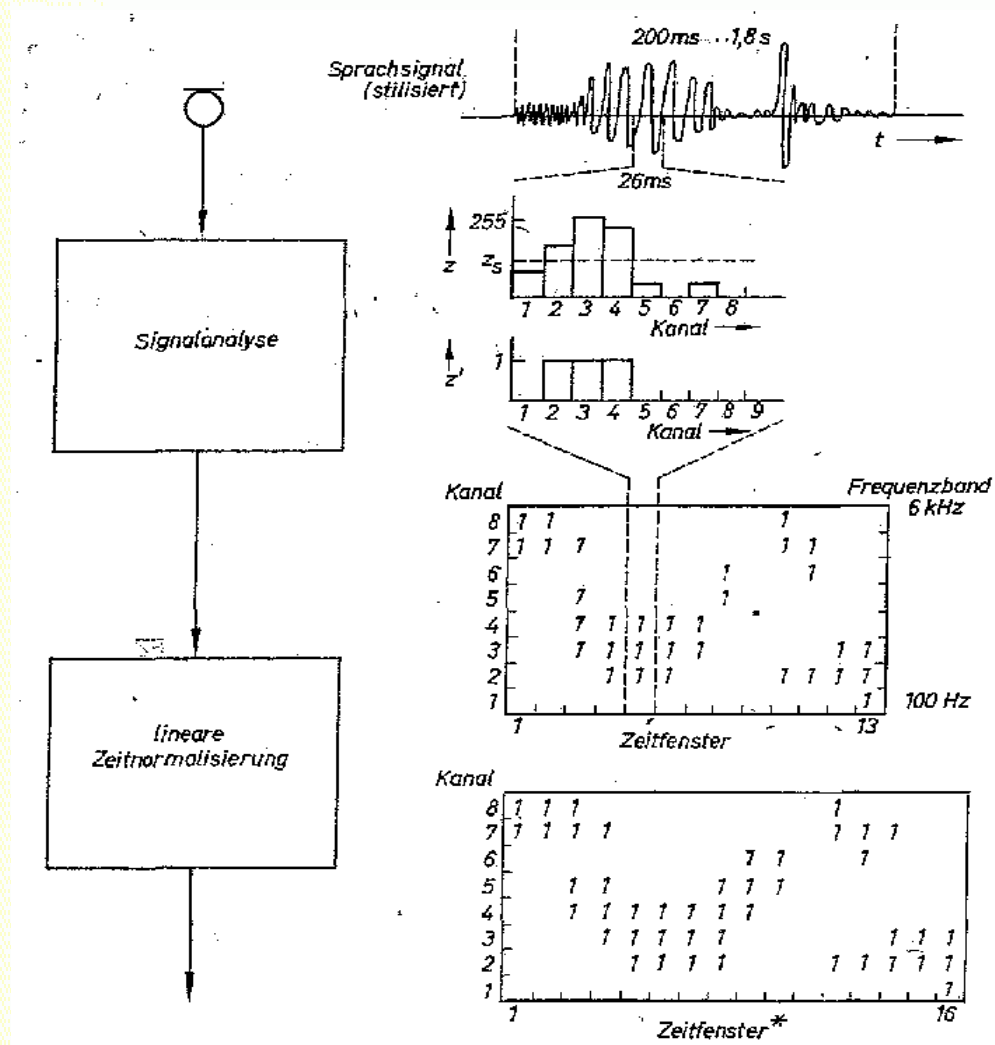


Bild 2: Bildung des Nulldurchgangshistogramms



Bildung der Muster



← Zeitfenster

← Merkmalsvektor

← 1-Bit-Quantisierung

← Muster (Folge von 8-Bit-Merkmalsvektoren)

← Normiertes Muster (16 Byte)



Beispiele



<p style="text-align: center;">zwei</p> <pre> 8 I **** I I * ** I I **** I I * **** I I **** ** I I **** ** I I * ** I I *** **** I 1 I 1234567890123456 s w a e </pre>	<p style="text-align: center;">sieben</p> <pre> 8 I **** I I * **** * I I **** **** I I **** I I **** I I **** ** I I **** ** I I **** ** I I **** ** I 1 I 1234567890123456 z i b m </pre>	<p style="text-align: center;">Kanal eins</p> <pre> 8 I I I * I I ** I I ** I I ** I I ** I I ** I I ** I I ** I 1 I 1234567890123456 a e n s </pre>
<p style="text-align: center;">vier</p> <pre> 8 I **** I I **** I I *** **** I I * **** I I * I I *** I I **** ** I I **** ** I 1 I 1234567890123456 f i e </pre>	<p style="text-align: center;">acht</p> <pre> 8 I I I ** I I ** * I I **** I I **** I I **** I I * I I * I 1 I 1234567890123456 a x - t </pre>	<p style="text-align: center;">Kanal fünf</p> <pre> 8 I ** I I *** ** I I **** *** I I * * * I I I I I I I I I 1 I 1234567890123456 f y n f </pre>



Erkennen der Wortklasse



Bitweiser Vergleich mit bereits gespeicherten Referenzmustern (Hammingdistanz = Anzahl der verschiedenen Bits)

Referenzmuster mit geringster Distanz =
erkanntes Wort

Distanz zu groß \Rightarrow Abweisung

Verbesserung: Wortklassen von
Referenzmustern



weitere Aufgaben



- Wortsignaldetektion (Anfang, Ende)
- Clusterbildung
- automatisches Verwerfen falscher Muster
- Erkennung bereits während der Signalanalyse



Software

33-1524 (SCPT v 7.8)

LO 0000 LWA
 CP 0000 PAUSE
 JP 0000 A. (AZUST)

ZUSATZ
 2. ZUSATZ

KEIN WORTENDE
 IN DER PROG. WORTENDE
 IN VOR. WORTENDE ODER SICHEREM WORTENDE

ANALYSE
 ZUSATZ
 ZUSATZ
 ZUSATZ

ALLE CLUSTER BETRACHTEN
 LAUFENDE NUMMER DER
 VERKANNTE KLASSE IN
 TABELLE DER ANZEIGE
 ANZEIGE SYMBOL DER

CALL BERLIN
 JR
 ZUSATZ

0355*	1A
035A*	FF 00*
035B*	3B 09
035C*	45
035D*	6F
035E*	78
035F*	86
0360*	8A
0361*	77
0362*	3E F0
0363*	45
0364*	6F
0365*	08
0366*	09
0367*	0A
0368*	0B
0369*	0C
036A*	0D
036B*	0E
036C*	0F
036D*	10
036E*	11
036F*	12
0370*	13
0371*	14
0372*	15
0373*	16
0374*	17
0375*	18
0376*	19
0377*	1A

LO A. (DE)
 CP KREN
 JP C. (PI)

PAUSE: PUSH HL
 PUSH DE
 PUSH RC

1)-BYTE-VEKTOR UND KOMPONENTENSUMME ERMITTELN
 LD A, (SWHL)
 LD C, A
 LD E, 0
 LD H, 0
 LD A, M
 LD H, 0
 CP C
 RR D

FC: HL-SCHWELWERT
 IB: KOMPONENTENZAHLER
 IAKTUELLEN VEKTORPUFFER LOESCHEN
 ID: 1)-BYTE-VEKTOR, HL-SCHWELLE UEBERSCHR. => 01

PAUSEDETEKTION
 PAUSE

LAUSPEICHERN EINES A-BYTE-VEKTORS AUS DEM VEKTORPUFFER
 LAUS 1)-BYTE-VEKTOR AUF DER AKTUELLEN ADRESSE IM SIGNALSPEICHER
 HL-REG. I ADRESSE DES AKT. VEKTORPUFFERS
 XPFEL: REG. I SIGNALSPEICHER
 XPFEL: ENDE DES SIGNALSPEICHERS
 XA: AKTUELLE ADRESSE IN XPFEL
 VANZ: ZAHL DER VEKTOREN BIS PROG. WORTENDE
 SW1: MINIMALE KO.-SUMME FUER SIGNALANFANGSERKENNUNG
 T1: MINDESTDAUER FUER SIGNALANFANGSERKENNUNG
 SW2: MAXIMALE KO.-SUMME FUER PAUSEERKENNUNG
 T2: MAXIMALE DAUER FUER PAUSEERKENNUNG
 SW3: MINIMALE KO.-SUMME FUER KNACKERKENNUNG
 T3: MAXIMALE DAUER FUER KNACKERKENNUNG
 TT1: AKTUELLE SIGNALDAUER NACH SIGNALANFANGSERK., T1=T1...0
 TT2: AKTUELLE PAUSEDAUER NACH PAUSEERKENNUNG, T2=T2...-13
 TT3: AKTUELLE KNACKDAUER NACH KNACKERK., T3=T3...0
 AZUST: BIT 0 GESETZT: PROG. WORTENDE
 BIT 1 GESETZT: SICHERER WORTANFANG
 BIT 2 GESETZT: PROG. WORTENDE
 BIT 3 GESETZT: SICHERES WORTENDE
 BIT 4 GESETZT: XPFEL-UEBERLAUF



Signalanalyse

- CTC-Kanal 0 und 3 arbeiten als Zähler mit Zählwert 1 und werfen beim Nulldurchgang von Tief- und Hochpass einen Interrupt (TP-NDG, HP-NDG)
- CTC-Kanal 1 arbeitet als Zeitgeber (Analyse-Zeitfenster), durch Abfrage des Zählwerts erhält man die Position innerhalb des Zeitfensters.
- Die NDG werten diese Position als Intervallklasse aus



Software-Umsetzung



- Die Interrupt-Routine des Zeitfensters sammelt die Intervallklassen von HP und TP in 8 Byte zusammen
- Die Pausendetektion (PAUSE) sucht Wortanfang und -Ende, der 8-Byte-Vektor wird 1-bit-quantisiert (Schwellwert SWHL)
- Die 8 Bit werden in den Signalspeicher XFELD hinzugefügt.

Zeitliche Normierung

- ZEIN normiert das Feld XFELD zu einem 16 Byte-Feld YFELD (ähnlich Bild stauchen/strecken)



Software-Umsetzung



Weitere Routinen

- VGLEI: minimale Hamming-Distanz berechnen
- KNORM: Clusterbewertung normieren
- CLST: Cluster streichen
- CLUST: Clusterbildung

das alles waren Routinen des
Spracherkenners SE 9001, © PHONEI 1986

Was fehlt nun noch?



Rahmenprogramm



```
KC 85/1 Emulator
Emulator Ansicht Optionen Hilfe

Musteranzahl im Wortschatz: 0
Eingestellter Teilwortschatz: 1

A Wortschatz laden
B Lernen
C Wortschatz abspeichern
D Teilwortschatz aendern
E Klassen umbenennen
F Klassen loeschen
G Wortschatz loeschen
STOP Programmende

Kennbuchstabe:■

50 fps
```

```
10 TW=1:POKE 80,TW :!Teilwortschatz einstellen
20 ! (nur bei Bedarf)
30 CALL DEEK(76) :!Aufruf des Erkennerprogramms A
40 ER=PEEK(81) :!Erkennungsergebnis
50 IF ER>127 THEN GOTO xxx :!Fehlerbehandlung
```

- Lernen
- Nachlernen
- Klassifizieren
- Laden/Speichern

Erweiterung

- 5 Teilwortschätze mit je max. 50 Klassen; insg. max 200 Referenzmuster

Nutzung von BASIC

- über definierte Schnittstelle: **CALL RECOG** ⇒ Ergebnis Klasse (0..50)



homecomputer.ddd-de.vu



Danke!