

Herwig Feichtinger

Tonerkennung per Software

Insbesondere bei Anwendungen von Einplatinen-Computern wie dem in mc 1981, Heft 2, vorgestellten „EMUF“ ist die Erkennung bestimmter Töne ein häufig vorkommendes Problem – etwa bei Modems, Funkfernseh-Konvertern oder Fernsteuerungen.

```

0000 ORL  =%E9F0      :AIM-45-
0000 NUMA  =%EA46      :UNTERPRG.
0000 PB    =%A800      :I/O-PORT
0000      *=%10C
010C      4C000C JMP TEST  :START=F1
010F      *=%0
0000 SUM  *=%*+1      :ERGEBNIS
0001 CNT  *=%*+1      :PERIODEN
0002 SPL1 *=%*+1      :BIT-
0003 SPL2 *=%*C00     :MUSTER
0C00
0C00      :AIM-TESTPROGRAMM
0C00 TEST A91E LDA #30   :DAUER:
0C02      9501 STA CNT   :60PERIODEN
0C04      A900 LDA #0    :MIT NULL
0C06      9500 STA SUM   :GEHT'S LOS
0C08 TEST0 201A0C JSR RD  :
0C08      C601 DEC CNT   :WIEDER
0C0D      00F0 BNE TEST0 :2PERIODEN
0C0F      20F0E9 JSR CRLF
0C12      A500 LDA SUM   :ZAEHLSTAND
0C14      2046EA JSR NUMA :AUSGEBEN
0C17      4C000C JMP TEST
0C1A
0C1A      :BITMUSTER LESEN
0C1A RD   A010 LDY #16   :2X8BITS
0C1C      78 SEI
0C1D RD1  A20A LDX #10   :1750HZ
0C1F RD0  CA DEX
0C20      D0FD BNE RD0
0C22      AD00AB LDA PB  :PB?EING.
0C25      0A ASL A
0C26      6602 ROR SPL1  :MUSTER
0C28      6603 ROR SPL2  :SPEICHERN
0C2A      8B DEY
0C2B      D0F0 BNE RD1
0C2D      58 CLI
0C2E      :MUSTER VERGLEICHEN
0C2E      A502 LDA SPL1  :MUSTER 1
0C30      F010 BEQ ERR
0C32      C9FF CMP #FF   :KEIN SIG.
0C34      F00C BEQ ERR
0C36      4503 EOR SPL2  :MUSTER 2
0C38      A007 LDY #7    :8 BITS
0C3A CHK1 6A ROR A       :GLEICHE
0C3B      B002 BCS CHK2  :BITS
0C3D      E600 INC SUM   :ZAEHLEN
0C3F CHK2 8B DEY
0C40      10F0 BPL CHK1  :Z-FLG=1:
0C42 ERR 60 RTS        :KEIN SIG.
0C43      .END
    
```

Bild 1. Mit dem 1-MHz-CPU-Takt als Referenz wertet dieses 6502-Programm die Frequenz 1750 Hz mit einer Bandbreite von etwa ± 100 Hz noch bei einem Signal-Rausch-Abstand von rund 3 dB aus! Der obere Teil namens TEST dient zur Demonstration auf dem AIM-65/PC-100, der eigentliche Auswerteteil RD ist voll relokatablel

Um unnötigen Hardware-Aufwand zu umgehen, ist man normalerweise bestrebt, dem Computer das niederfrequente Eingangssignal mit TTL-Pegel an einem I/O-Port zur Verfügung zu stellen und es der CPU zu überlassen, festzustellen, ob es sich um die „gesuchte“ Frequenz handelt.

Für diesen Zweck wurden bereits mehrere Verfahren entwickelt [1, 2, 3]; auch die in vielen Mikrocomputern vorhandene Betriebssoftware für das Kassetten-Interface arbeitet meist nach einem dieser Verfahren. Dabei wird entweder der zeitliche Abstand zwischen zwei Nulldurchgängen oder die Zahl der Nulldurchgänge pro Zeiteinheit gemessen. Ein anderes, nicht minder interessantes Verfahren arbeitet auch bei gestörten, verrauschten Eingangssignalen noch ausreichend zuverlässig und trägt den Namen „Autokorrelation“ [4]. Ein Programm für die Prozessorfamilie 65XX, das auf diesem Prinzip aufbaut, zeigt Bild 1.

Dabei wird das anliegende Signal mit der achtfachen erwarteten Frequenz abgetastet, d.h. auf eine Periode der Sollfrequenz fallen acht Abtastwerte (Samples). Jeder Abtastwert kann, da das Signal als TTL-Pegel am I/O-Port anliegt, 0 oder 1 sein. Die ersten acht Werte füllen das Byte SPL1, die nächsten acht SPL2 (Bild 2).

In SPL1 und SPL2 findet sich dann also je eine digitalisierte Periode des Eingangssignals – sofern diese der Sollfrequenz entspricht. Ist das der Fall, so sind beide Bytes im Idealfall gleich, und zwar unabhängig von der gerade vorhandenen Phasenlagen des Signals;

Die Sollfrequenz kann – bedingt durch Zählschleifen Prozessor-Geschwindigkeitsgrenzen – zwischen 96 Hz und 4,63 kHz liegen; frequenzbestimmend ist das Byte an der Adresse 0C1E, hier dezimal 10 für 1750 Hz.

Relevant für die richtige Erkennung der Sollfrequenz ist also lediglich, daß korrespondierende (gleichwertige) Bits in SPL1 und SPL2 gleich sind. Bei verrauschtem oder gestörtem Signal wird

das natürlich trotz korrekter Frequenz nicht für jedes Bit stimmen. Deshalb stellt das Programm in der Schleife CHK1 fest, wieviele gleichwertige Bits übereinstimmen, und erhöht für jedes die Zählvariable SUM. Eine „perfekte“ Übereinstimmung wäre natürlich auch gegeben, wenn gar kein Eingangssignal anliegt, d.h. wenn der Eingang konstant auf Null oder Eins liegt; das Programm erkennt dies daran, daß SPL1 entweder 00 oder hex FF ist und verzichtet dann auf das Weiterzählen von SUM.

Je größer schließlich der Wert von SUM ist, desto sicherer wurde die Sollfrequenz erkannt. Der maximal erreichbare SUM-Wert ist 8 pro RD-Durchlauf, in unserem Fall beim AIM-Testprogramm also hex F0. Eine vernünftige Entscheidungsschwelle, ob der Ton nun anliegt oder nicht, ist das Überschreiten von etwa zwei Dritteln dieses Maximalwertes, hier z.B. hex A0. Dies beeinflusst natürlich auch die Bandbreite.

Im Beispielpogramm TEST dient der Kassettenport des AIM-65 bzw. PC-100 als Signaleingang; um sicherzustellen, daß er als Eingang geschaltet ist, sollte man vor dem Programmstart (F1) die Reset-Taste drücken. Auf dem AIM-Display erscheint als Maß für die Übereinstimmung mit der Sollfrequenz eine Hex-Zahl zwischen 00 und F0 – nämlich die über 60 Perioden aufsummierte Zahl gleicher Bits in SPL1 und SPL2. Neben seiner Störuneempfindlichkeit hat das Autokorrelations-Programm auch den

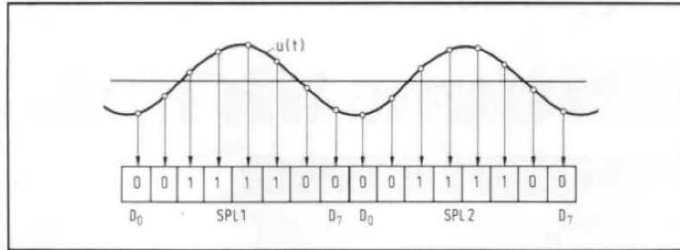


Bild 2. Aus zwei Perioden der Sollfrequenz werden 16 Abtastwerte gewonnen. Liegt die richtige Frequenz an, so sind die ersten acht gleich den letzten acht Bits

```

10 INPUT "SOLLFREQ./HZ" : F
20 K=INT((1E4/F-176)/48+.5)
21 F=1E6/(48*K+176)
30 IF K>0 AND K<257 GOTO 50
40 PRINT " NICHT REALISIERBAR" : END
50 PRINT " F="INT(F+.5)"HZ" : K="K
RUN
SOLLFREQ./HZ? 1750
F= 1736 HZ : K= 10
    
```

Bild 3. Auf dem PC-100 entwickeltes Basic-Programm zum Errechnen der frequenzbestimmenden Konstante K an der Adresse 0C1E sowie der tatsächlichen Mittenfrequenz

Vorteil, daß seine Laufzeit von anliegenden Eingangssignal unabhängig ist, was bei [2] und [3] nicht der Fall ist.

Bild 3 zeigt ein kleines Basic-Programm, das die Konstante K an der Adresse 0C1E errechnet, die die Mittenfrequenz

bestimmt, sowie die sich tatsächlich aus K ergebende Mittenfrequenz. Im nächsten Heft werden wir über ein verwandtes Verfahren, die Kreuzkorrelation, berichten.

Literatur

- [1] Nf-Zähler, in: Anwendungsbeispiele für den Mikroprozessor 6502, RPB 173, Franzis-Verlag, München.
- [2] FSK-Demodulation per Software. Sonderheft „Programme für Kleincomputer und Taschenrechner“, Franzis-Verlag, München.
- [3] Ein Software-Tondecoder. Sonderheft „Programme für Kleincomputer und Taschenrechner“, Franzis-Verlag, München.
- [4] Korrelationsanalyse, in: Taschenbuch Elektronik, Bd. 2. Carl-Hanser-Verlag, München/Wien.

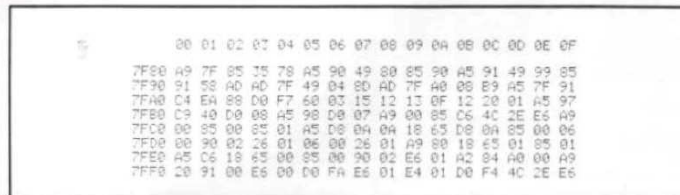
CBM-Cursor-Utility

Das im Bild als Hex-Dump aufgelistete Maschinenprogramm wurde für den CBM 3032 (32 KByte, kommerzielle Tastatur) geschrieben. Es bietet zusätzlichen Komfort bei den Cursor-Funktionen, der besonders beim Redigieren von Programmen nützlich ist: Durch das Betätigen der ESC-Taste wird der Cursor an den Anfang der laufenden Zeile gesetzt; Shift-ESC löscht den Bildschirm ab der Cursor-Position. Nach dem Laden des Programms wird es mit SYS 32640 aktiviert und meldet „CURSOR E“; das Ausschalten erfolgt auf die gleiche Weise, wobei „CURSOR A“ ausgegeben wird. Das Programm liegt am oberen Ende des

32-KByte-RAM-Bereiches und ist automatisch gegen Zerstörung durch Basic-Programme geschützt. Michael Greth

Literatur

ROM und RAM in PET und CBM. Franzis-Sonderheft „Mikrocomputer-Anwendungen“.



Mit diesem kleinen Maschinenprogramm bewirkt ein Druck auf die ESC-Taste, daß der Cursor zum linken Zeilenrand geht; Shift-ESC löscht den Rest der Zeile rechts vom Cursor