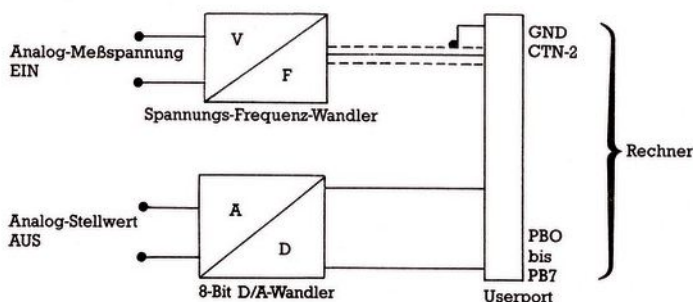


Analoger Meßwert rein — analoger Stellwert raus

In diesem Beitrag wird eine einfache Anordnung für den Commodore 64 beschrieben, bei der ohne Umschalten ein Analogsignal eingegeben und ein Analogstellwert ausgegeben werden kann.

Für einen regeltechnischen Versuch wurde eine Anordnung gebaut, bei der über eine dauernd mit dem Computer verbundene Meßleitung ein analoges Meßsignal aufgenommen, im Computer verarbeitet und über die 8-Bit-Datenleitung des Userports und einen 8-Bit-D/A-Wandler wieder ein analoges Stellsignal als Antwort ausgegeben werden kann (siehe Bild)



Ein- und Ausgangsschaltung

Diese Anordnung dürfte auch für viele Bastelzwecke sehr interessant sein. Für den Analogeingang wurde mit dem IC LM311 ein Spannungs-Frequenzwandler gebaut. Die Frequenz des erzeugten Rechtecksignals wurde über eine eindrige Schirmleitung mit Ader auf PIN6 = CNT-2 des Userports und Schirm auf PIN1 = GND zur Frequenzmessung eingegeben. Diese Anwendung ist prinzipiell in [1] beschrieben, allerdings mit dem kleinen Schönheitsfehler, daß die Sache so nicht funktioniert. Trotzdem ist das erwähnte Buch sehr empfehlenswert wegen seiner Vielfalt und Verständlichkeit. Im folgenden wird zunächst ein funktionsfähiges Programm gezeigt und anschließend werden wesentliche Verbesserungen unter Verwendung einer Assembler-Routine geschildert. Ein Studium der Literatur [1 bis 4] zeigt, daß für eine einfache Frequenzmessung bis hinauf zu etwas mehr als 20 kHz der im CIA 6526 (Complex Interface-Adapter-IC) eingebaute Zähler und die Systemuhr TI als Zeitbasis in Frage kommen. Wer einen Rechteckgenerator zur Hand hat, kann dies auch gleich (TTL-Pegel 5 V einstellen) mal ausprobieren! Geben sie das folgende Basic-Programm ein und schließen Sie den Generator — wie beschrieben — an CNT-2 an.

```

90 REM****BASIC-PROGRAMM FREQUENZMESSUNG*****
100 POKE56590,240
110 POKE56580,255:POKE56581,255
120 T=TI
130 IFT=TITHEN130
140 POKE56590,241
150 IFTI<>T+60THEN150
160 A=PEEK(56581):B=PEEK(56580)
170 PRINT65535-(256*A+B)
180 GOTO100:REM !!DAUERSCHLEIFE!

```

Listing »Frequenzmessung«

Es wurden folgende Adressen verwendet:

Dez. 56590 Hex DD0E Kontrollregister für Zähler 1

Dez. 56580 Hex DD04 Zähler 1 Low-Byte T1L

Dez. 56581 Hex DD05 Zähler 1 High-Byte T1H

In Zeile 100 wird das Kontrollregister für den Zähler 1 so programmiert, daß dieser in der gewünschten Art arbeitet und zunächst im STOP-Zustand ist. In Zeile 110 wird der Zähler mit HB = 255 und LB = 255 geladen; er hat die Eigenschaft, diese zwei Byte nach dem Start herunterzuzählen (Dez. 65535 bis 0). Der Zähler kann nun aber nicht, wie in [1] erwähnt, mit dem Setzen des High-Byte von T1 (T1H) gestartet werden, sondern durch den Befehl Poke 56590, 241, bei dem Bit 0 im Kontrollregister auf 1 = START gesetzt wird. Die TI-Uhr wird bei diesem Programm »im Vorübergehen« gelesen, die beiden IF-Abfragen geben den Start- und den Auslesebefehl für den Zähler.

Im Kontrollregister haben nur Bit 0 und 5 wichtige Funktionen:

11110001 = Dez. 241 → Zähler Start

11110000 = Dez. 240 → Zähler Stop

Die anderen Bits sind als »Dauermaske« aufzufassen.

Das Basicprogramm zeigt jedoch leider bei der praktischen Anwendung Anzeigeschwankungen von etwa 2 Prozent und manchmal weit darüber. Schuld daran ist sicher die Programmiersprache Basic, welche zum Beispiel aufgrund von zu langsamen Bearbeitungszeiten des Basic-Interpreters in den IF-Abfragen ziemliche Fehler in der Zeitbasis zu verursachen scheint. Gelegentlich kommt die ganze Messung aber auch noch stärker »aus dem Tritt«. Geht man dazu über, die Uhr TI nicht »im Vorübergehen« abzulesen, dann kann man schon eine etwas bessere Genauigkeit erreichen.

```

90 REM****BASIC-PROGRAMM2 FREQUENZMESSUNG*****
100 POKE56590,240
110 POKE56580,255:POKE56581,255
120 POKE160,0:POKE161,0
130 POKE162,0:POKE56590,241
150 IFTI<>T+60THEN150
160 A=PEEK(56581):B=PEEK(56580)
170 PRINT65535-(256*A+B)
180 GOTO100:REM !!DAUERSCHLEIFE!

```

Listing »Frequenzmessung« 2

In diesem Programm wird in Zeile 120/130 durch EinPOKEN von 0 in die drei TI-Register 160, 161, 162 die Uhr TI jedesmal neu gestartet, wobei Register 162 des »schnellsten Rädchens« zuletzt genullt wird. Hierdurch entfällt die erste IF-Schleife des Starts. Benützt wird im Prinzip nur Register 162, in welchem die Uhr von 0 bis 255 in Schritten von 1/60 s (Jiffies) läuft. Die beiden anderen Register werden nur zur Sicherheit genullt.

Ausgehend von diesem Programm wurde das folgende Assembler-Programm geschrieben, das im Prinzip den gleichen Ablauf hat, bei dem aber die Startbefehle für Zähler und Uhr TI sowie der Stop-Befehl für den Zähler und vor allem die Zeitabfrage und das Ablesen »blitzschnell« ablaufen:

```

DD0E      100 CRA      =      $DD0E
DD04      110 T1L     =      $DD04
DD05      120 T1H     =      $DD05
00A2      130 TIM     =      $A2
00A0      131 TY1    =      $A0
00A1      132 TY2    =      $A1
C000      140          *=      $C000
C000      A9 F0      150      LDA      #$F0
C002      8D 0E DD   160      STA      CRA
C005      A9 00      220      LDA      #$00
C007      85 A0      221      STA      *TY1
C009      85 A1      222      STA      *TY2
C00B      85 A2      230      STA      *TIM
C00D      A9 F1      240      LDA      #$F1
C00F      8D 0E DD   250      STA      CRA
C012      A5 A2      260      MA      LDA      *TIM
C014      C9 3C      270          CMP      #$3C
C016      D0 FA      280          BNE      MA
C018      AE 05 DD   290          LDX      T1H
C01B      AC 04 DD   300          LDY      T1L
C01E      60          320          RTS
          330          .EN
    
```

Listing »Frequenzmessung Maschinensprache«

```

100 FOR I = 49152 TO 49182
110 READ X:POKEI,X:S=S+X:NEXT
120 DATA169,240,141, 14,221,169, 0,133,160,133,161,133
130 DATA162,169,241,141, 14,221,165,162,201, 60,208,250
140 DATA174, 5,221,172, 4,221, 96
150 IFS<> 4561 THEN PRINT"FEHLER IN DATAS":END
160 PRINT"OK"
    
```

Listing »Basiclader«

Nach einmaligem RUN-Lauf kann dieses Programm immer mit SYS 49152 aufgerufen werden, der Basiclader kann dann gelöscht werden (vorher natürlich speichern!). Das Programm arbeitet einen Durchlauf des Meßzyklus ab und kehrt dann nach Basic zurück, zuvor hat es jedoch HB und LB des Zählers, T1H und T1L, nach Adresse Dez 781 und 782 abgelegt. Aus diesem »Briefkasten« können die Werte mit PEEK geholt werden. Als Beispiel ein Miniprogramm zur dauernden Frequenzanzeige in Hz:

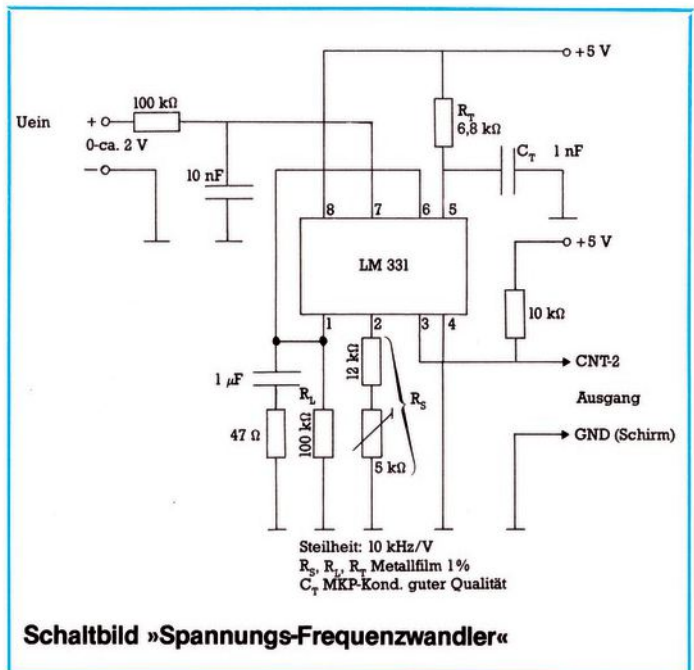
```

100 SYS49152
110 A=PEEK(781):B=PEEK(782)
120 PRINT65535-(256*A+B)
130 GOTO100:REM !!DAUERSCHLEIFE!
    
```

Listing »Frequenzanzeige«

Mit diesem Programm wurde eine Genauigkeit von deutlich besser als 1 Prozent erhalten. Versuche zeigten, daß selbst im praktischen Betrieb mit einem Spannungs-Frequenzwandler die Meßdauer ohne Verschlechterung des Ergebnisses auf $30 \times 1/60 \text{ s} + 0,5 \text{ s}$ verkürzt werden kann. Dies kann man im Basiclader dadurch bewerkstelligen, daß in Zeile 130 die Zahl 60 in 30 geändert wird.

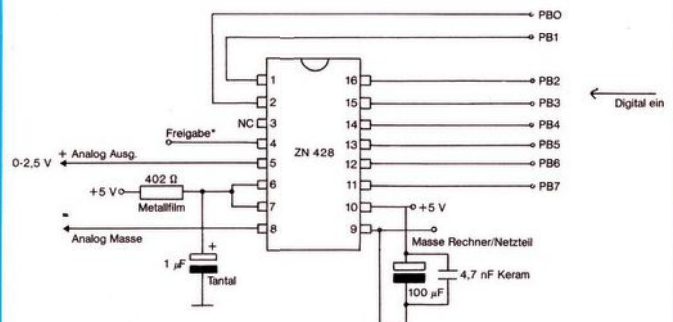
Bei der praktischen Anwendung wurde entsprechend [1] beziehungsweise nach Angaben des Herstellers [5] mit dem IC LM 331 folgender Spannungs-Frequenzwandler gebaut:



Schaltbild »Spannungs-Frequenzwandler«

Die Schaltung funktioniert sehr gut und recht genau! Für die »Stellenwertausgabe« wurde mit dem 8-Bit-D-A-Wandler ZN428 folgende Schaltung aufgebaut:

Schaltbild »Stellwertausgabe«



*: Bei »Freigabe« Low oder Open: Wert wird laufend »durchgeschoben«. Bei »Freigabe« High: Wandler gesperrt, alter Wert wird noch angezeigt

Die Ausgabe über den Userport erfolgt mit Hilfe folgender Befehle:

Poke 56579, 255 (Datenleitungen PBO bis PB7 auf Ausgabe)

Poke 56577, x (x = Dez.-Wert der Ausgabe)

Der Wandler erzeugt dann eine zu x proportionale Spannung.

Dem Nachteil einer V/f-Wandler-Zeit von $\approx 0,5 \text{ s}$ steht der Vorteil gegenüber, daß keine Ein-Ausgabeleitungs-Umschaltung erforderlich ist, die Signale stehen stetig am Port zur Verfügung. Eine mehrkanalige Ausführung des Eingangs durch Multiplexen ist denkbar. Beim Ausgang ist dies sicher etwas schwieriger.

(Dr. Ernst Breit/rg)

Literatur

[1] Anderson A., Kullbjör A.: »Messen, steuern, regeln mit dem VC 20 und Commodore 64«, Haller-Verlag, Saarbrücken

[2] Angerhausen, Brückmann, Englisch, Gerits, »64 intern«, Data-Becker, Düsseldorf

[3] Floegel E., »Hardware-Erweiterungen für Commodore 64«, Hofacker-Verlag, Holzkirchen

[4] Brückmann, »Der Commodore 64 und der Rest der Welt«, Data-Becker, Düsseldorf

[5] National-Semiconductor, Daten-Blatt zu LM 131/231/331