

DDS-HF-Sinus-Generator mit BASIC-Tiger[®] und AD7008

Gunther Zielosko

1. Grundlagen

Abstimmbare Sinus-Generatoren oder -Oszillatoren gibt es fast solange, wie es Elektronik überhaupt gibt. Insbesondere im Bereich der HF-Sende -und Empfangstechnik werden sie überall eingesetzt. Begonnen hat alles mit dem LC-Schwingkreis, bei dem eine Komponente zum Verstellen der Frequenz mit speziellen mechanisch veränderbaren Bauteilen ausgerüstet war (Kapazität -> Drehkondensator oder Induktivität -> Variometer). Später wurde die Kapazität mit sogenannten Kapazitätsdioden bereits elektrisch abgestimmt. Der nächste Schritt war der spannungsgesteuerte Oszillator (VCO). All diese Verfahren hatten den Nachteil, daß die erzeugte Frequenz war nicht sehr stabil war. Im Gegensatz zur hohen Stabilität von Quarzoszillatoren mit einer starren Frequenz verursachten Temperaturschwankungen, Betriebsspannung, Bauteiltoleranzen und andere Einflüsse unerwünschte Frequenzänderungen des Generators. Ein Durchbruch wurde mit der PLL-Technik erzielt, bei der die Ausgangsfrequenz eines VCO über einen veränderbaren digitalen Teiler mit einer starren Quarzfrequenz verglichen wird. Weicht die Ausgangsfrequenz in Frequenz oder Phase ab, erfolgt über eine Regelung die Nachstimmung des VCO. Diese PLL-Schaltungen sind in der HF-Technik heute überall anzutreffen.

Aber die Entwicklung geht weiter. Die immer dichter werdenden Funkkanäle fordern eine immer genauere und stabilere Frequenzerzeugung, größere Variationsbereiche und eine feinere Abstimmung. Dies führte zur Entwicklung der DDS-Technik (Digitale Direkte Synthese). Dieses Verfahren arbeitet vollständig digital und bietet demzufolge viele Vorteile gegenüber allen vorherigen Varianten. Allerdings erreicht man mit den verfügbaren Bausteinen zur Zeit noch keine sehr hohen Frequenzen.

Das Prinzip der DDS-Technik ist einfach erklärt. In einem ROM sind die Werte einer Sinusschwingung in digitaler Form gespeichert. Diese Werte werden zeitlich nacheinander ausgelesen und über einen DA-Wandler in diskrete Strom- bzw. Spannungswerte umgeformt und ausgegeben (siehe Bild 3). Das Auslesen erfolgt mit einer stabilen Frequenz, die über einen programmierbaren Teiler direkt aus einem Quarzgenerator abgeleitet wird.

Als BASIC-Tiger[®] -Anwender werden Sie nun fragen, was DDS und BASIC-Tiger[®] miteinander zu tun haben und wozu man das Ganze braucht.

Der DDS-Baustein AD7008, den wir benutzen wollen, muß von außen programmiert werden. Zum Beispiel soll die Ausgangsfrequenz eingestellt werden, ideal wäre dazu eine kleine Tastatur. Die Ausgangsfrequenz f errechnet sich aus der Eingangsfrequenz und einer 32Bit Binärzahl (Teilerverhältnis), das erfordert Fließkomma-Arithmetik. Schön wäre auch die Anzeige der Frequenz mit hoher Genauigkeit. Noch interessanter wird diese Anzeige, wenn z.B. bei einem Empfänger automatisch noch die Zwischenfrequenz berücksichtigt wird. Der BASIC-Tiger[®] stellt komfortabel die Bedien-, Rechen- und Anzeigefunktionen zur Verfügung, der AD7008 macht den hochfrequenten Rest.

Die in dieser Applikation vorgestellte Hard- und Software bildet den Grundstein für Experimente mit Sinussignalen, hier einige mögliche Anwendungen:

- In Frequenz und Amplitude fein einstellbarer Sinusgenerator für die Elektroniker-Werkstatt (NF bis HF)
- Sweepgenerator mit automatisch veränderbaren Frequenzen
- Grundbaustein für Kurzwellen-Empfänger und -Sender
- FSK modulierter Sender

2. Der DDS-Baustein AD7008

Ein klassischer DDS-Baustein ist der AD7008 von Analog Devices, der alle oben dargestellten Funktionen beinhaltet. Ein ausführliches Datenblatt kann man über <http://www.analog.com/pdf/ad7008.pdf> erhalten. Es gibt zwei Varianten des AD7008, die sich in ihrer maximalen Taktfrequenz unterscheiden (20MHz, 50MHz). Die Taktfrequenz wird mit einem programmierbaren Teiler in 2^{32} Schritte eingeteilt, damit ist die Ausgangsfrequenz sehr fein einstellbar. Als Faustregel gilt, daß als Ausgangsfrequenz maximal ein Drittel der Eingangsfrequenz eingestellt werden sollte. Damit kann die 50MHz-Variante des AD7008 eine Ausgangsfrequenz von etwa 16MHz mit gerade noch ausreichender Sinusform erzeugen.

Der AD7008 bietet eine Reihe von Eigenschaften und Programmiermöglichkeiten, die in diesem Applikationsbericht nur teilweise genutzt werden sollen:

- er arbeitet mit einer einzigen 5V Betriebsspannung
- liefert gleichzeitig eine Sinus- und eine Cosinusschwingung
- erlaubt Frequenz-, Phasen- und Amplitudenmodulation (QAM und SSB möglich)
- parallele oder serielle Programmierung
- digitale Modulation (FSK)
- Power down-Option
- Synchronisierung mit DDS-Takt möglich und abschaltbar

Bild 1 zeigt die Anschlußbelegung des AD7008 in einem 44-poligen PLCC-Gehäuse.

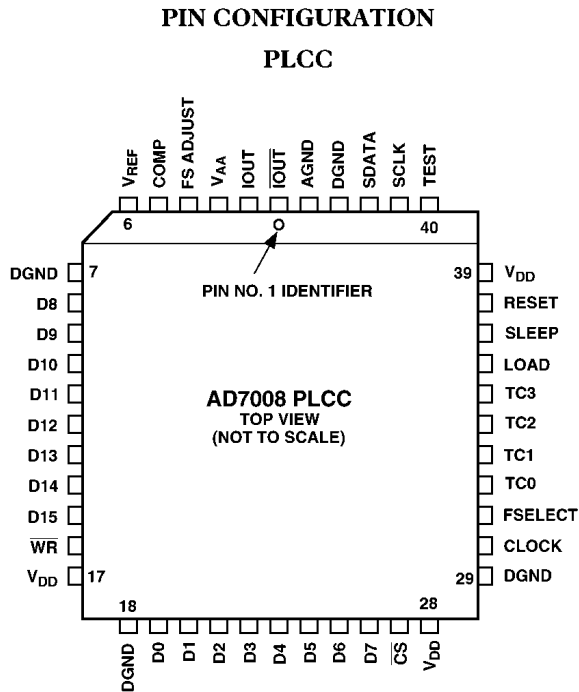


Bild 1 Anschlußbelegung des AD7008

3. Ansteuerung durch den BASIC-Tiger®

Die Beschaltung des AD7008 ist relativ einfach. Ein Quarzgenerator liefert die Taktfrequenz von maximal 50MHz (bzw. 20MHz je nach Variante des AD7008!). Die Ausgänge IOUT zw. IOUT werden mit 50Ω beschaltet und liefern eine Sinus- bzw. Cosinusschwingung. Noch ein paar Kondensatoren, Drosseln und Widerstände - fertig. Eine wichtige Rolle spielen die Filter an den Ausgängen, die die Quarzfrequenz-"Reste" aus der Sinusschwingung entfernen sollen. Wie wirksam sie sind, kann man in Bild 3 sehen. Diese Filter müssen natürlich auf Ihre spezielle Quarzfrequenz abgestimmt sein, die Werte in Bild 2 helfen nur überschlägig.

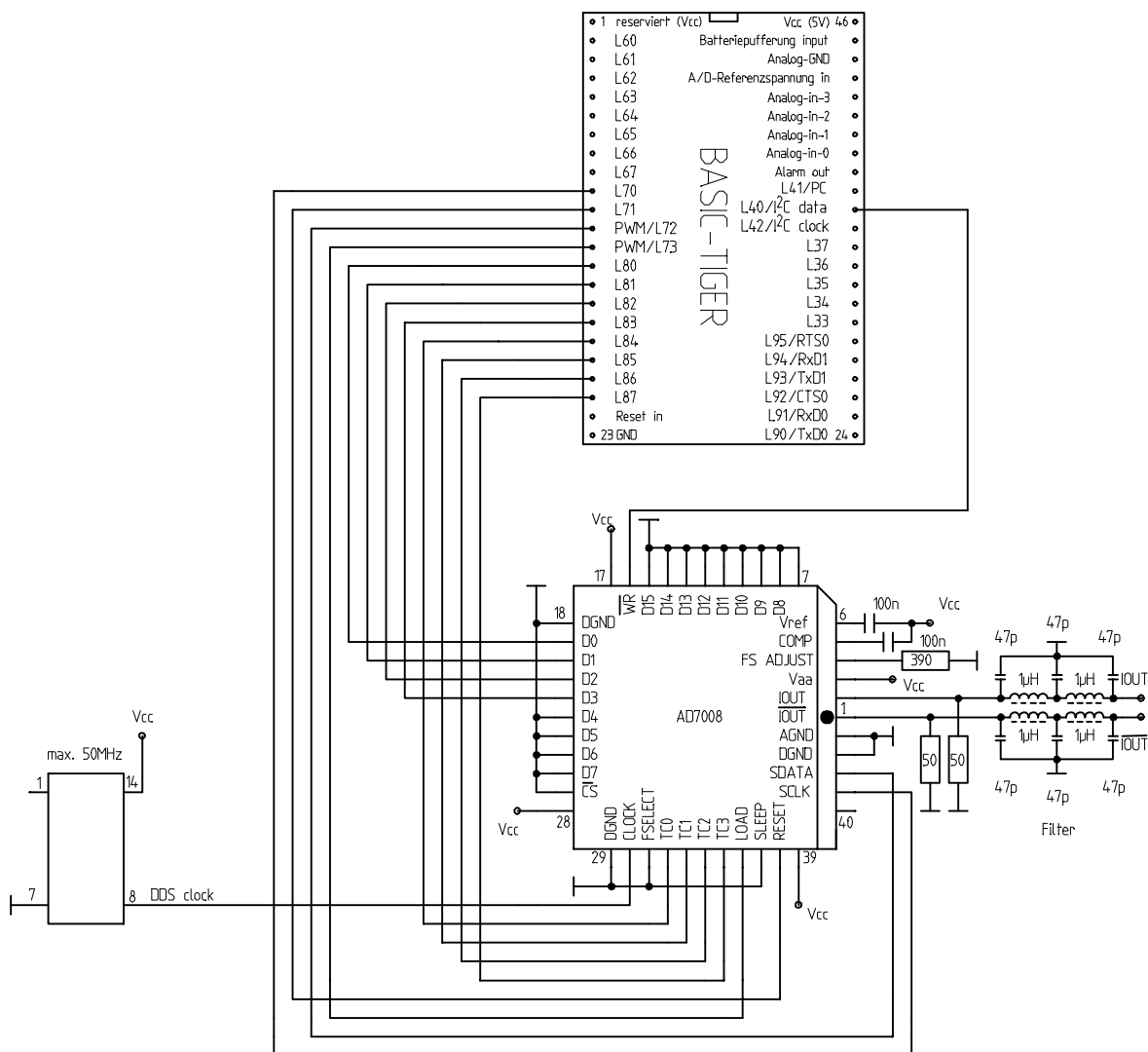


Bild 2 Schaltung des BASIC-Tigers® mit AD7008

Die Programmiermöglichkeiten des AD7008 sind vielseitig, Feinheiten können im Datenblatt nachgelesen werden. Hier sollen nur die wichtigsten Eigenschaften behandelt werden. Zunächst erlaubt der AD7008 eine parallele oder eine serielle Programmierung. In unserem Fall fällt die Wahl auf die serielle Programmierung. Damit kann sichergestellt werden, daß die übliche Beschaltung des BASIC-Tigers® mit alphanumerischem Display, Tastatur, seriellen Schnittstellen usw. ohne Zusatzschaltungen uneingeschränkt benutzbar bleiben. Zur Funktionsauswahl werden einige parallele Ansteuerleitungen gebraucht. Will man auch die alternative Frequenz Fout1 nutzen (z.B. für FSK - eine digitale Modulation, hier wird zur Darstellung von L und 0 zwischen zwei Frequenzen umgeschaltet), muß das Pin FSELECT des AD7008 extern umgeschaltet werden. In der vorliegenden Version liegt es fest auf GND, d.h. es wird immer Fout0 ausgegeben. Die vom Autor gewählte Verschaltung des BASIC-Tigers® mit dem AD7008 (Bild 2) ist auch die Basis für die Beispielprogramme.

Einige Steuerleitungen des AD7008 werden in der gewählten Schaltung direkt, also nicht seriell angesteuert. Dies sind neben RESET, LOAD, und WR die 4 Adreßbits TC0-TC3 des Transfer-Steuer-Registers und die ersten 4 Datenbits D0-D3, die in diesem Falle die 4 Bits des Command-Register darstellen.

Zur seriellen Programmierung des AD7008 sind als wichtigstes zunächst die beiden Anschlüsse SCLK und SDATA zu bedienen. Der erste ist ein Taktpin, der zweite das dazugehörige Datenpin. Diese Anschlüsse werden mit dem Befehl SHIFT_OUT des BASIC-Tigers® geladen (höchstwertiges Bit zuerst, maximal 32 Bit). Diese 32 Bit brauchen wir gerade, um z.B. den Frequenz-Einstellwert in den AD7008 zu bringen, aus der dann das Teilverhältnis und damit die Ausgangsfrequenz abgeleitet wird.

Die Ausgangsfrequenz des AD7008 errechnet sich nach folgender Formel:

$$F_{out} = \frac{\text{Einstellwert} \times \text{Quarzfrequenz}}{2^{32}}$$

oder, wenn man eine bestimmte Ausgangsfrequenz haben möchte:

$$\text{Einstellwert} = \frac{F_{out} \times 2^{32}}{\text{Quarzfrequenz}}$$

Genau diese Berechnung führt der BASIC-Tiger® aus, wenn der AD7008 eine bestimmte Frequenz ausgeben soll (wie gut, daß wir im Tiger-Basic alle Möglichkeiten der Gleitkomma-Arithmetik haben und problemlos mit Realzahlen arbeiten können).

Der Ablauf der Programmierung des AD7008 erfolgt dann nach der Initialisierung (alle Pins bekommen den vorgeschriebenen Ruhepegel, danach RESET) immer so:

- Der Einstellwert wird seriell in ein Auffangregister geschrieben. Dabei tut sich äußerlich noch nichts (keine Änderung des vorherigen Zustandes).
- Mit den 4 Pins TC0-TC3 wird vorgegeben, was am AD7008 neu eingestellt werden soll. Das kann die Frequenz 0, die Frequenz 1 oder eine der Modulationsvarianten Phase oder

IQMOD sein. Die Einzelheiten zeigt die Tabelle 1. Mit einem High-Impuls an LOAD wird der seriell übergebene Wert in das Zielregister eingetragen, damit ist der neue Zustand eingestellt (siehe Beispielprogramm 7008_01.TIG).

TC3	TC2	TC1	TC0	LOAD	Register
X	X	X	X	0	Keine Änderung
1	1	0	0	1	Frequenz 0
1	1	0	1	1	Frequenz 1
1	1	1	0	1	Phase
1	1	1	1	1	IQ-Modulation

Tabelle 1 Übertragung der seriell eingelesenen Daten in das Zielregister

In der Tabelle 1 wurden nur die Funktionen für die serielle Programmierung der Grundfunktionen dargestellt. Das ist zunächst genug, um die beiden Frequenzen zu programmieren, wer möchte, kann dann auch noch verschiedene Modulationsarten einstellen, das Prinzip ist immer ähnlich.

Eine Sonderrolle spielt das Command-Register. Nach dem RESET liegen alle 4 Bits des Commandregisters auf 0, das reicht für den normalen Betrieb meist aus. Das Wort für das Commandregister, das man z.B. zur Ermöglichung der Amplitudensteuerung auf 0010 umladen muß, wird zunächst parallel mit einem kurzen Low-Impuls auf WR über die Datenbits D0 - D3 übernommen. Danach wird dieser "zwischenlagerte" Wert per High-Impuls auf LOAD in das Command-Register übertragen (siehe Programm 7008_01.TIG).

4. Resultate

Der Autor benutzte für die ersten Experimente einen AD7008 (20MHz-Version) mit einem zufällig vorhandenem 15,360MHz Quarzoszillator. Am Ausgang IOOUT wurde die Kurvenform mit Filter (untere Kurve in den Oszillogrammen), am negierten Ausgang das ungefilterte Signal (oben) abgenommen. Es werden die Kurvenformen bei verschiedenen Frequenzen dargestellt, im ersten Bild z.B. 1,7005MHz. Man sieht deutlich die vom inneren DA-Wandler bereitgestellten Spannungssprünge, die durch das Filter aber sehr effektiv "ausgebügelt" werden. Bei kleineren Frequenzen fällt dieser Effekt kaum noch ins Auge (folgende Bilder). Ebenfalls gut zu sehen ist die Phasenverschiebung des Signales an den beiden Ausgängen (Sinus und Cosinus).

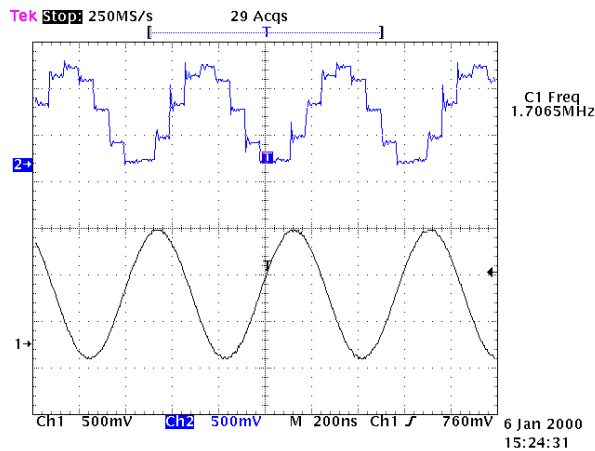


Bild 3 1,7MHz ungefiltert und gefiltert

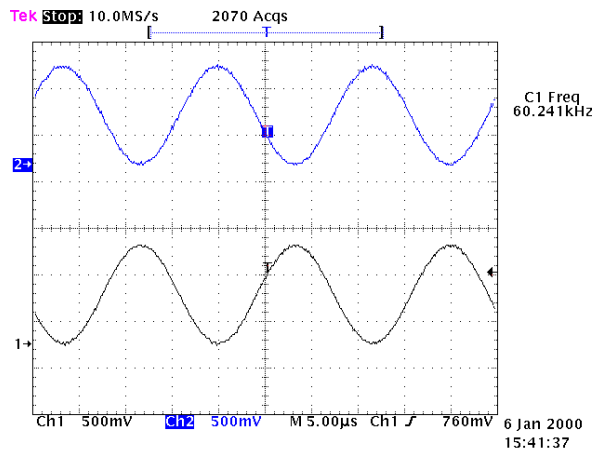


Bild 4 Signale bei 60kHz

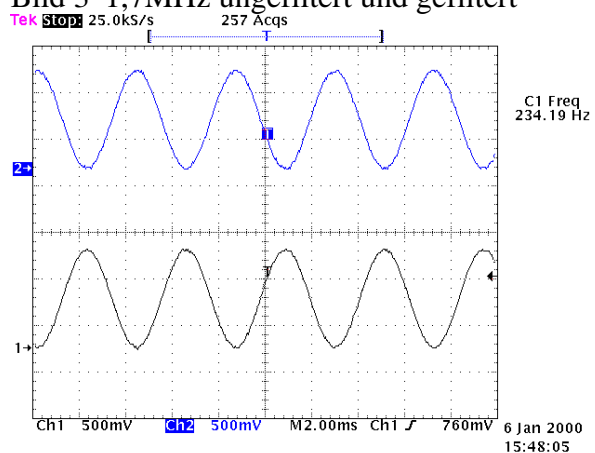


Bild 5 bei 234Hz

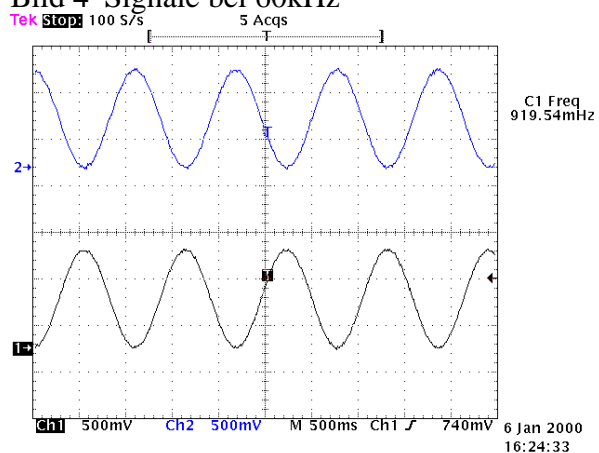


Bild 6 und bei weniger als 1Hz (!)

Die Bilder 7 und 8 zeigen die Funktion der Amplitudensteuerung. Zu beachten ist bei allen Anwendungen des Sinus- bzw. Cosinussignales, daß es immer einen Gleichspannungsanteil (Offset) gibt, wer nur die Wechselfspannung braucht, muß einen Koppelkondensator vorsehen.

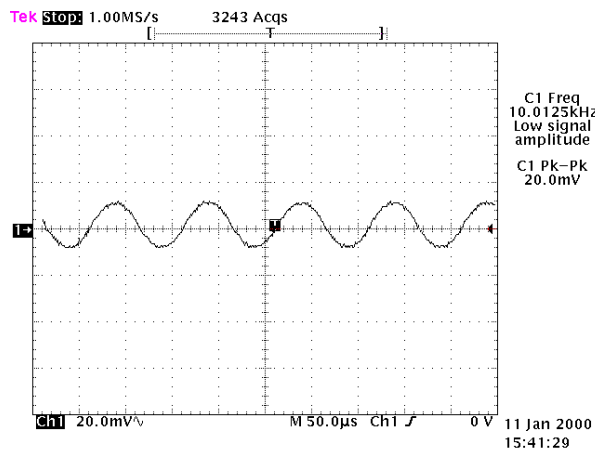


Bild 7 10kHz mit 20mV Amplitude

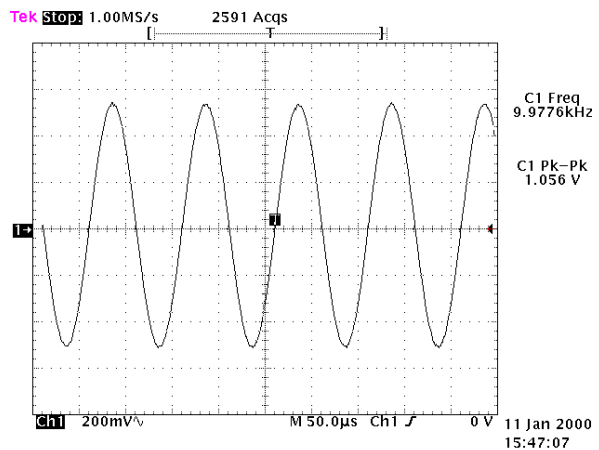


Bild 8 und mit 1V Amplitude

5. Software

Die Funktionen, die Steuermöglichkeiten und die Einsatzfälle des AD7008 sind so vielfältig, daß ein Applikationsbericht wie dieser nur einen kleinen Teil der Fragen lösen kann. Das gilt auch für die Software. Es werden zwei kurze Programme vorgestellt, die weitgehend automatisch die Frequenz (7008_01.TIG) und die Amplitude (7008_02.TIG) in weiten Grenzen steuern. Auf Bedienelemente, komplizierte Menü-Funktionen oder grafische Darstellungen auf dem Display wurde bewußt verzichtet, um die Programme übersichtlich zu halten.

```
-----  
' Name: 7008_01.TIG  
' Das Programm erzeugt mit dem AD7008 eine variable Sinus-Frequenz Fout0.  
' Am Anfang werden alle Steuereingaenge initialisiert und ein RESET erzeugt.  
' Danach werden die Daten fuer die zu erzeugende Sinusfrequenz eingestellt  
' (Quarzfrequenz, Anfangsfrequenz). Die Frequenzausgabe beginnt mit 0,1Hz  
' und laeuft bis 5MHz, immer in Schritten von 10%. Zwischen den eingestell-  
' ten Werten wird der vorherige Wert 5 Sekunden gehalten.  
'  
' Verdrahtung:  
' AD7008 | Pin | BASIC-Tiger®  
' WR      | 16 | L40  
' SCLK    | 41 | L70  
' RESET   | 38 | L71  
' SDATA   | 42 | L72  
' LOAD    | 36 | L73  
' D0      | 19 | L80  
' D1      | 20 | L81  
' D2      | 21 | L82  
' D3      | 22 | L83  
' TC0     | 32 | L84  
' TC1     | 33 | L85  
' TC2     | 34 | L86  
' TC3     | 35 | L87  
-----
```

```
TASK MAIN                                ' Beginn Task MAIN  
INSTALL DEVICE #1, "LCD1.TDD"            ' LCD-Display installieren  
  
LONG D                                    ' Variablen definieren  
REAL Fout0                                ' Ausgabewert für Frequenz  
REAL Fquarz                                ' Ausgangsfrequenz Fout0  
REAL Faktor                                ' Quarzfrequenz  
                                           ' 2 hoch 32  
  
DIR_PORT 8,0                              ' Ausgangspins festlegen  
DIR_PIN 7,0,0                              ' alle Pins Port 8 sind Ausgang  
DIR_PIN 7,1,0                              ' Pin 70 ist Ausgang  
DIR_PIN 7,2,0                              ' Pin 71 ist Ausgang  
DIR_PIN 7,3,0                              ' Pin 72 ist Ausgang  
DIR_PIN 7,3,0                              ' Pin 73 ist Ausgang  
DIR_PIN 4,0,0                              ' Pin 40 ist Ausgang  
  
                                           ' Initialisierungswerte  
OUT 7,00000001b,0                          ' Pin 70 (SCLK) Ruhepegel 0  
OUT 7,00001000b,0                          ' Pin 73 (LOAD) Ruhepegel 0
```

```

OUT 7,00000100b,0      ' Pin 72 (SDATA) Ruhepegel 0
OUT 7,00000010b,0      ' Pin 71 (RESET) Ruhepegel 0
OUT 4,00000001b,255    ' Pin 16 (WR) Ruhepegel 1

                                ' RESET auslösen
OUT 7,00000010b,255    ' RESET-Eingang kurz auf High
OUT 7,00000010b,0      ' RESET-Eingang wieder auf Low
WAIT_DURATION 10        ' etwas warten

Anfang:                  ' Sprungmarke "Anfang"
                                ' Vorgabewerte für Frequenzeinstellung
    Fout0 = 0.1           ' Vorgabe der Anfangsfrequenz (0,1Hz)
    Fquarz = 15360000     ' Quarzfrequenz (konstant)
    Faktor = 4294967295   ' 2 hoch 32 als Dezimalzahl (konstant)

Weiter:                  ' Sprungmarke Weiter
                                ' Ausgabe auf LCD-Display
USING "NF<8><1> <1>V3.3.3.3.3.3.3.3,0.0.0.0.0.0.0" ' Ausgabeformat Frequenz
PRINT USING #1, "<1>F0 =" ;Fout0; "Hz"          ' Ausgabe der Ausgangsfrequenz
PRINT USING #1, "FQ =" ;Fquarz; "Hz"           ' Ausgabe der Quarzfrequenz auf LCD

                                ' Ausgabefrequenz errechnen
    D = (Fout0/Fquarz)*Faktor                  ' Formel D = Fout0/Fquarz*Faktor
    CALL F_LOAD_0 (D)                          ' Lade Frequenzregister 0

    WAIT_DURATION 5000                         ' 5 Sekunden warten

    Fout0 = Fout0 * 1.1                       ' Frequenz um 10% erhoehen

    IF Fout0 > 5000000 THEN                    ' wenn Frequenz > 5MHz, zu
        GOTO Anfang                          ' Sprungmarke "Anfang" gehen
    ENDIF

    GOTO Weiter                                ' ansonsten Frequenz weiter erhoehen
                                                ' zu Sprungmarke "Weiter" gehen

END                                            ' Ende Task MAIN

```

```

-----
SUB F_LOAD_0 (D)
    SHIFT_OUT 7, 2, 0, D, -32
    OUT 8,255,11000000b
    OUT 7,00001000b,255
    OUT 7,00001000b,0
END
-----

```

```

-----
' Name: 7008_02.TIG
' Das Programm erzeugt mit dem AD7008 eine feste Sinus-Frequenz Fout0.
' Am Anfang werden alle Steuereingaenge initialisiert und ein RESET erzeugt.
' Danach werden die Daten fuer die zu erzeugende Sinusfrequenz eingestellt
' (Quarzfrequenz, Anfangsfrequenz). Der letzte Programmteil laesst die
' Amplitude der Sinusschwingung kontinuierlich groeßer und wieder kleiner
' werden

```

Verdrahtung:

AD7008	Pin	BASIC-Tiger®
WR	16	L40
SCLK	41	L70
RESET	38	L71
SDATA	42	L72
LOAD	36	L73
D0	19	L80
D1	20	L81
D2	21	L82
D3	22	L83

```
' TC0      | 32 | L84
' TC1      | 33 | L85
' TC2      | 34 | L86
' TC3      | 35 | L87
```

```
-----
TASK MAIN                                ' Beginn Task MAIN
INSTALL DEVICE #1, "LCD1.TDD"           ' LCD-Display installieren

                                           ' Variablen definieren
LONG D                                  ' Ausgabewert für Frequenz
LONG A                                  ' Ausgabewert fuer Amplitude
REAL Fout0                              ' Ausgangsfrequenz Fout0
REAL Fquarz                             ' Quarzfrequenz
REAL Faktor                              ' 2 hoch 32

                                           ' Ausgangspins festlegen
DIR_PORT 8,0                            ' alle Pins Port 8 sind Ausgang
DIR_PIN 7,0,0                            ' Pin 70 ist Ausgang
DIR_PIN 7,1,0                            ' Pin 71 ist Ausgang
DIR_PIN 7,2,0                            ' Pin 72 ist Ausgang
DIR_PIN 7,3,0                            ' Pin 73 ist Ausgang
DIR_PIN 4,0,0                            ' Pin 40 ist Ausgang

                                           ' Initialisierungswerte
OUT 7,00000001b,0                       ' Pin 70 (SCLK) Ruhepegel 0
OUT 7,00001000b,0                       ' Pin 73 (LOAD) Ruhepegel 0
OUT 7,00000100b,0                       ' Pin 72 (SDATA) Ruhepegel 0
OUT 7,00000010b,0                       ' Pin 71 (RESET) Ruhepegel 0
OUT 4,00000001b,255                     ' Pin 16 (WR) Ruhepegel 1

                                           ' RESET auslösen
OUT 7,00000010b,255                     ' RESET-Eingang kurz auf High
OUT 7,00000010b,0                       ' RESET-Eingang wieder auf Low
WAIT_DURATION 10                         ' etwas warten

                                           ' Vorgabewerte für Frequenzeinstellung
Fout0 = 10000                            ' Vorgabe der Anfangsfrequenz (10kHz)
Fquarz = 15360000                        ' Quarzfrequenz (konstant)
Faktor = 4294967295                      ' 2 hoch 32 als Dezimalzahl (konstant)

                                           ' Ausgabefrequenz errechnen
D = (Fout0/Fquarz)*Faktor                ' Formel D = Fout0/Fquarz*Faktor
CALL F_LOAD_0 (D)                       ' Lade Frequenzregister 0

CALL A_MOD                               ' Ermoegliche Amplitudenmodulation

Anfang:                                  ' Sprungmarke "Anfang"

FOR A = 1 TO 1023                         ' Amplitude laeuft von 1 (Min)
                                           ' auf 512 (Max) hoch und geht bis
                                           ' 1023 (Min) wieder runter
                                           ' 20-Bit IQMOD-Register geht bis 1048576
CALL A_LOAD (A)                          ' In IQMOD-Register laden

PRINT #1, "<1>A ="; A                    ' Amplitudenwert ausgeben
WAIT_DURATION 50                          ' etwas warten
NEXT

GOTO Anfang                              ' wieder neu beginnen

END                                        ' Ende Task MAIN

-----
SUB F_LOAD_0 (D)                          ' Unterprogramm Laden Frequenz 0
SHIFT_OUT 7, 2, 0, D, -32                ' 32 Bits werden ausgegeben, MSB zuerst
OUT 8,255,11000000b                      ' Ziel-Register Frequenz 0 (hoechste 4
                                           ' 4 Bit (1100) -> FREQ0-Register
```

```
        OUT 7,00001000b,255      ' LOAD-Eingang kurz auf High
        OUT 7,00001000b,0        ' LOAD-Eingang wieder auf Low
END
'-----
'
SUB A_MOD                          ' Unterprogramm Amplituden-Modu-
                                  ' lation einstellen

        OUT 8,255,00000100b      ' Paralleleingabe der Werte
                                  ' für Command-Register
        OUT 4,00000001b,0        ' Amplituden-Mod. moeglich (CR2=1)
        OUT 4,00000001b,255     ' WR-Eingang kurz auf Low
                                  ' WR-Eingang wieder auf High

        OUT 8,255,00000000b      ' in Command-Register (hoechste 4
                                  ' 4 Bit) uebernehmen
                                  ' (0000) -> COMMAND-Register
        OUT 7,00001000b,255     ' LOAD-Eingang kurz auf High
        OUT 7,00001000b,0        ' LOAD-Eingang wieder auf Low

END
'-----
'
SUB A_LOAD (A)                     ' Unterprogramm Laden Amplitude
        SHIFT_OUT 7, 2, 0, A, -20 ' 20 Bits werden ausgegeben, MSB zuerst
        OUT 8,255,11110000b      ' Ziel-Register IQMOD (hoechste 4
                                  ' 4 Bit (1111) -> IQMOD-Reister
        OUT 7,00001000b,255     ' LOAD-Eingang kurz auf High
        OUT 7,00001000b,0        ' LOAD-Eingang wieder auf Low

END
'-----
```