```
$prog &HFF , &HEF , &HC9 , &HFF                          ' generated.
'
'<STEUERRECHNER_V0.23_M.BAS>
'
'         ************************************************************
'         *                                                          *
'         *            Programm für Testaufbau Steuerrechner         *
'         *            -------------------------------------         *
'         *           mit vollständiger Synthesizer-Steuerung        *
'         *           ---------------------------------------        *
'         *                                                          *
'         *            Version      0.23_M                         *
'         *                                                          *
'         *            23.05.2010   W. Nitsche                     *
'         *                                                          *
'         *            geschrieben in BASCOM-AVR                   *
'         *                                                          *
'         ************************************************************
'
'
'Achtung: Für dieses Programm muss der Stack folgendermassen initialisiert sein:
'         HW Stack    64
'         Soft Stack  64                   (für lokale Variable in Subroutinen)
'         Frame Size  64                   (für lokale Variable in Subroutinen)
'
'$sim
'$sim ist zum Simulieren da und muß normal auskommentiert sein
'
'Positions-Parameter --- Pospar = 1 --- --------------------------------
'Sub Menue_moni            =  1
'Sub Setup                 =  2
'Sub Menue_freqenz         =  3
'Sub Dac_value             =  4
'Sub Raster_10_4           =  5
'Sub Raster_13_0           =  6
'Sub Ic9_ausgangsteiler    =  7
'Sub Rd_adc7               =  8
'Function M_value          =  9
'Function T1_value         = 10
'Function T2_value         = 11
'Function Tf_value         = 12
'Sub PLL_synthesizer       = 13
'Sub Ic9_data              = 14
'Sub Ic12_data             = 15
'Sub S_data                = 16
'Sub N_data                = 17
'Sub Hmc10_4               = 18
'Sub Hmc13                 = 19
'Sub Calc_hmc698           = 20
'Sub Add_data              = 21
'Sub Dac_data              = 22
'Sub Dac_tx                = 23
'Sub Calc_ad9951           = 24
'Sub Set_ad9951            = 25
'Sub Transmit_dds          = 26
'Sub Transmit_dac          = 27
'Sub Transmit_hc595        = 28
'Sub Drehgeber             = 29
'Sub Freq_aenderung        = 30
'Sub Para_synthe           = 31
'Sub N_frequenz            = 32
'Sub Prog_sythe()          = 33
'Sub Bereichs_variable     = 34
'Sub Frequenz_variable     = 35
'Sub Tx_synthe             = 36
'Sub F_bereich             = 37
'Sub Frequenzanzeige       = 38


'Include Files für ATMega128 einbinden --------------------------------
$regfile = "m128def.dat"

'Initialisierung CPU ---------------------------------------------------
$hwstack = 64
$framesize = 64
$swstack = 64

'Zuweisungen -----------------------------------------------------------
'Testpunkte
Tp1 Alias Porte.2                                        'Zuweisung TP1 = PORTE.2
Tp2 Alias Porte.3                                        'Zuweisung TP2 = PORTE.3
Tp3 Alias Porte.4                                        'Zuweisung TP3 = PORTE.4
Tp4 Alias Porte.5                                        'Zuweisung TP4 = PORTE.5
Tp5 Alias Porte.6                                        'Zuweisung TP5 = PORTE.6
Tp6 Alias Porte.7                                        'Zuweisung TP6 = PORTE.7

'Synthesizer CLOCK
Sdio_dds Alias Porta.0                                   'Zuweisung SDIO DDS-Baustein     = PORTA.0
Sclk_dds Alias Porta.1                                   'Zuweisung Clock DDS-Baustein    = PORTA.1
Ioupdate_dds Alias Porta.2                               'Zuweisung IOUPDATE DDS-Baustein = PORTA.2
Reset_dds Alias Porta.3                                  'Zuweisung Reset DDS-Baustein    = PORTA.3
Sdo_dds Alias Pina.4                                     'Zuweisung SDO DDS-Baustein      = PORTA.4

'Synthesizer PLL
Sync_dac Alias Portb.0                                   'Zuweisung  / SYNC_DYC           = PORTB.0
Sclk_dac Alias Portb.1                                   'Zuweisung SCLK_DAC              = PORTB.1
Sdin_dac Alias Portb.2                                   'Zuweisung SDIN_DAC              = PORTB.2
Lock_loop_2 Alias Portb.3                                'Zuweisung Lock_Loop_2           = PORTB.3
Latch_clk Alias Portb.4                                  'Zuweisung LATCH_CLK             = PORTB.4
Shift_clk Alias Portb.5                                  'Zuweisung SHIFT_CLK             = PORTB.5
Ser_data Alias Portb.6                                   'Zuweisung SER_DATA              = PORTB.6
Reset_src Alias Portb.7                                  'Zuweisung RESET_SRC             = PORTB.7

'Initialisierung Serielle Schnittstellen -------------------------------
$crystal = 16000000                                      'Quarz 16,000MHz
Config Com2 = 9600 , Synchrone = 0 , Parity = None , Stopbits = 1 , Databits = 8 , Clockpol = 0
Config Com1 = 9600 , Synchrone = 0 , Parity = None , Stopbits = 1 , Databits = 8 , Clockpol = 0

'Installiere einen Ringbuffer im Interrupt
Config Serialin1 = Buffered , Size = 20

'Initialisierung I2C-Bus -----------------------------------------------
Config Sda = Portd.5
Config Scl = Portd.4
```

```
I2cinit

'Initialisierung LC-Display im 4 Bit-Pin-Mode und 16 *2 Character ------
Config Lcd = 16 * 2
'Die Complier-Einstellungen werde mit "Config Lcdpin" überschrieben
Config Lcdpin = Pin , Db4 = Portc.4 , Db5 = Portc.5 , Db6 = Portc.6 , Db7 = Portc.7 , E = Portc.3 , Rs = Portc.1

'Initialisierung AD-Converter ----------------------------------------
'Config Adc = Single , Prescaler = Auto                      'Reference = Internal
                                                            'Reference = Internal darf nicht
                                                            'eingeschalten werden, wenn eine
                                                            'externe Spannung angeschlossen ist

'Initialisiere External RAM
Config Xram = Disabled , Waitstatels = 1 , Waitstatehs = 2  'Externen RAM-Zugriff ausschalten


'Initialisierung Port-Konfiguration ----------------------------------
Ddra.0 = 1                                                  'Config SDIO_DDS     = Output
Ddra.1 = 1                                                  'Config SCLK_DDS     = Output
Ddra.2 = 1                                                  'Config IOUPDATE_DDS = Output
Ddra.3 = 1                                                  'Config RESET        = Output
Ddra.4 = 0                                                  'Config SDO_DDS      = Input
Ddra.5 = 0                                                  'Config Pina.5       = Input
Ddra.6 = 0                                                  'Config Pina.6       = Input
Ddra.7 = 0                                                  'Config Pina.7       = Input

Ddrb.0 = 1                                                  'Config Pinb.0 = Output
Ddrb.1 = 1                                                  'Config Pinb.1 = Output ISP-Programmierung
Ddrb.2 = 1                                                  'Config Pinb.2 = Output
Ddrb.3 = 0                                                  'Config Pinb.3 = Input
Ddrb.4 = 1                                                  'Config Pinb.4 = Output
Ddrb.5 = 1                                                  'Config Pinb.5 = Output
Ddrb.6 = 1                                                  'Config Pinb.6 = Output
Ddrb.7 = 1                                                  'Config Pinb.7 = Output

Ddrc.0 = 1                                                  'Config Pinc.0 = Output
Ddrc.2 = 1                                                  'Config Pinc.2 = Output

Ddrd.0 = 0                                                  'Config Pind.0 = Input
Ddrd.1 = 0                                                  'Config Pinc.1 = Input
Ddrd.6 = 0                                                  'Config Pinb.6 = Input Drehschalter DR0
Ddrd.7 = 0                                                  'Config Pinb.7 = Input Drehschalter DR1

'Initialisierung der Ports -------------------------------------------
Dim Porta.0 As Bit                                          'SDIO_DDS     = Output
Dim Porta.1 As Bit                                          'SCLK_DDS     = Output
Dim Porta.2 As Bit                                          'IOUPDATE_DDS = Output
Dim Porta.3 As Bit                                          'RESET_DDS    = Output
Dim Porta.4 As Bit                                          'SDO_DDS      = Input
Dim Porta.5 As Bit                                          'Reserve
Dim Porta.6 As Bit                                          'Reserve
Dim Porta.7 As Bit                                          'Reserve

Dim Portb.0 As Bit                                          '/SYNC_DAC    = Output
Dim Portb.1 As Bit                                          'SCLK_DAC     = Output
Dim Portb.2 As Bit                                          'SDIN_DAC     = Output
Dim Portb.3 As Bit                                          'PB.3         = Input
Dim Portb.4 As Bit                                          'LATCH_CLK    = Output
Dim Portb.5 As Bit                                          'SHIFT_CLK    = Output
Dim Portb.6 As Bit                                          'SER_DATA     = Output
Dim Portb.7 As Bit                                          'RESET_SR     = Output

Dim Portc.0 As Bit                                          'LCD-Display Beleuchtung
Dim Portc.2 As Bit                                          'LCD-Display R/W

Dim Portd.0 As Bit                                          'INT0 Tasten einlesen
Dim Portd.1 As Bit                                          'INT1 Drehschalter
Dim Portd.6 As Bit                                          'PD.6 = Drehschalter DR0
Dim Portd.7 As Bit                                          'PD.7 = Drehschalter DR1

'Initialisierung Testpunkte ------------------------------------------
Ddre.2 = 1                                                  'Config Pine.2 = Output TP1
Ddre.3 = 1                                                  'Config Pine.3 = Output TP2
Ddre.4 = 1                                                  'Config Pine.4 = Output TP3
Ddre.5 = 1                                                  'Config Pine.5 = Output TP4
Ddre.6 = 1                                                  'Config Pine.6 = Output TP5
Ddre.7 = 1                                                  'Config Pine.7 = Output TP6
Dim Porte.2 As Bit                                          'TP1
Dim Porte.3 As Bit                                          'TP2
Dim Porte.4 As Bit                                          'TP3
Dim Porte.5 As Bit                                          'TP4
Dim Porte.6 As Bit                                          'TP5
Dim Porte.7 As Bit                                          'TP6

'Initialisierung globaler Variabler ----------------------------------
Dim Hz1 As Integer                                          'Hilfsvariable 1
Dim Hz3 As Integer                                          'Hilfsvariable 3
Dim Hz4 As Integer                                          'Hilfsvariable 4
Dim Hz5 As Integer                                          'Hilfsvariable 5
Dim Hw1 As Word                                             'Hilfsvariable

Dim Frq_bereich As Integer                                  'Frequenzbereich
Dim Frq_bereich_alt As Integer                              'Frequenzbereich alt

Dim Var1 As Word                                            'Hilfsvariable für Printroutine
Dim Var2 As Word                                            'Hilfsvariable für Printroutine
Dim Posp As Byte                                            'Hilfsvariable für Printroutine
Dim Var1_alt As Word                                        'Hilfsvariable für Printroutine
Dim Var2_alt As Word                                        'Hilfsvariable für Printroutine
Dim Buglog As Byte                                          'Hilfsvariable DEBUG 0 = aus
Dim Ss As Byte                                              'Single Step 0 = aus
Dim Position As Byte                                        'Position Variable
Dim Vari As Long                                            'Variable

Dim Pospar As Byte                                          'Positionsparameter
Dim Wert As Byte                                            'Variable zur Umwandlung einer Variablen
Dim Posipar As String * 5                                   'Positionsparameter als String

Dim Remote As Byte                                          'Character von der Remote-Steuerung
Dim Moni As Byte                                            'Monitorsteuerung
Dim Mode_sw As Byte                                         'Schalterstellung Mode-Schalter
Dim Mode_sw_alt As Byte                                     'alte Schalterstellung Mode-Schalter
```

```
Dim Tasten_1 As Byte                                    'ADC-Value für Tasten_1
Dim Tasten_2 As Byte                                    'ADC-Value für Tasten_2
Dim Tastenfeld As Byte                                  'ADC-Value für Tastenfeld
Dim Tastatur As Byte                                    'Tastenfeld aktiv
Dim Tast_value As Byte                                  'Tastatur-Eingabe

Dim Dac_data As Word                                    'Datenwort für DAC AD5611
Dim Dac_val As Word                                     'DAC-Zwischenspeicher
Dim Zae1_data As Word                                   'DAC-Zwischenspeicher Normalmode
Dim Add_data_hw As Word                                 'Hilfsword für Subroutine Add_data
Dim Data_595 As Word                                    'Datenwort für Shieberegister 74HC595

Dim Tune As Integer                                     'hier werden die Drehungen am Tune-Knopf gezählt
Dim Zaehler As Integer                                  'Zähler für Drehgeber
Dim Zaehler_z As Integer                                'Zwischenspeicher *10 für Drehgeber
Dim Zae1 As Integer                                     'Hilfszähler für Testzwecke
Dim Zae1_alt As Integer                                 'Hilfszähler für Testzwecke
Dim Zae_ic9 As Integer                                  'Hilfszähler für Testzwecke
Dim Zae_ic9_alt As Integer                              'Hilfszähler für Testzwecke
Dim Mm As Byte                                          'Variable M
Dim Zae_ic12 As Integer                                 'Hilfszähler für Testzwecke
Dim Zae_ic12_alt As Integer                             'Hilfszähler für Testzwecke
Dim Zae_m As Integer                                    'Hilfszähler für Testzwecke
Dim Zae_s As Integer                                    'Hilfszähler für Testzwecke
Dim Zae_s_alt As Integer                                'Hilfszähler für Testzwecke
Dim Zae_n As Integer                                    'Hilfszähler für Testzwecke
Dim Zae_n_alt As Integer                                'Hilfszähler für Testzwecke
Dim Zae_hmc As Integer                                  'Hilfszähler für Testzwecke
Dim Zae_hmc_alt As Integer                              'Hilfszähler für Testzwecke
Dim Zae_hmc_alt1 As Integer                             'Hilfszähler für Testzwecke
Dim X1 As Integer                                       'Hilfsvariable
Dim B As Integer                                        'Hilfsvariabele

Dim Dek_pos As Byte                                     'Zähler für Dekaden-Position
Dim X As Byte                                           'Horizontal-Position auf dem LCD-Display
Dim Y As Byte                                           'Vertikal-Position auf dem LCD-Display
Dim Z As Byte , Slave As Byte                           'Speicherzelle für I2C-Bus read
Dim Differenz As Long                                   'gewünschte Frequenzänderung
Dim Nennfrequenz As Long                                'Nennfrequenz ohne Berücksichtigung der ZF usw.
Dim Nennfrequenz_alt As Long                            'Nennfrequenz alt zum Vergleich mit Nennfrequenz
Dim Freqstr As String * 12                              'Frequenzanzeige als String
Dim Adc_val As Word , Channel As Byte                   'Variablenübergabe A/D-Wandler
Dim Adc_val_alt As Word                                 'Wert zwischenspeichern
Dim Gd_val As Byte                                      'Gain und Decreasing kombiniert
Dim Gd_val_old As Byte                                  'Zwischenspeicher für Gd_val
Dim Tx_data As Byte                                     'Datenbyte zur Übertragung an den DDS
Dim Tx_bit As Byte                                      'Datenbyte zur Maskierung des zu übertragenden Bits
Dim Tx_word As Word                                     'Datenword zur Maskierung des zu Übertragenden Bits
Dim Lcd_bit As Byte                                     'Datenbit ausmaskieren für LCD-Display
Dim Dac_aktiv As Byte                                   'Überwacht, ob der DAC aktiv oder in Trestate ist
Dim Adc_mvolt As Word                                   'Spannung PLL-Regelschleife in mVolt

'Variable zur Berechnung des 32-Bit Tuning Word für den DDS -----------
Dim Dds_frequenz As Long                                'DDS-Frequenz
Dim Frequenz As Long                                    'Frequenzübergabge zum Programmieren des DDS-Bausteins
Dim F_tw As Long                                        'Frequenz Tunig Word
Dim Frequenz_1 As Long                                  'DDS-Frequenz
Dim I As Byte                                           'Laufvariable
Dim Lu1 As Byte                                         'Variable aus Lookup-Tabelle Mult:
Dim Lud As Long                                         'Hilfsvariable
Dim Y1 As Long                                          'Hilfsvariable
Dim Z1 As Long                                          'Hilfsvariable
Dim Hw As Word                                          'Hilfsvariable
Dim Lw As Word                                          'Hilfsvariable
Dim Ftw(5) As Byte                                      'Array zur Übertragung der Daten zum DDS-Baustein

'Stringvariable ------------------------------------------------------
Dim Nm As String * 5
Dim Hstring As String * 20                              'Hilfsstring zur Stringbearbeitung

'Initialisierung von I2C-Bus-Adressen ---------------------------------
'Const Pcf8574_exp = &H44                                'PCF8574 I2C-Bus Expander  <======= Test mit I2C-Bus-Expander
Const Pcf8574_ic8 = &H40                                'PCF8574 Bandfilter Preselector IC8
                                                        'LED0 = BF_A =   50kHz bis 500kHz
                                                        'LED1 = BF_B =  500kHz bis 1,8MHz
                                                        'LED2 = BF_C =  1,5MHz bis 2,47MHz
                                                        'LED3 = BF_D = 2,47MHz bis 4,07MHz
                                                        'LED4 = BF_E = 4,07MHz bis 6,7MHz
                                                        'LED5 = BF_F =  6,7MHz bis 11,05 MHz
                                                        'LED6 = BF_G =11,05MHz bis 18,2MHz
                                                        'LED7 = BF_H = 18,2MHz bis 30MHz

Const Pcf8574_ic13 = &H42                               'PCF8574 Gain Preselector IC13
                                                        'LED0 = Gain Eingangsdämpfungsglied
                                                        'LED1 = Gain G1 Verstäerker
                                                        'LED2 = Gain G2 Verstärker
                                                        'LED3 = Gain G3 Verstärker
                                                        'LED4 = Gain G4 Verstärker
                                                        'LED5 = NC_1
                                                        'LED6 = NC_2
                                                        'LED7 = NC_3

'Initialisierung Procedures und Functions -----------------------------
Declare Sub Monitor                                     'Fernsteuerung via PC
Declare Sub Setup                                       'hier kann der setup gemacht werden

Declare Sub Drehgeber                                   'Der Drehgeber wird eingelesen
Declare Sub Freq_aenderung                              'Die Frequenzänderung wird berechnet
Declare Sub N_frequenz                                  'In dieser Subroutine wird die Nennfrequenz berechnet
Declare Sub Frequenzanzeige                             'Frequenzanzeige am LCD-Display
Declare Sub Prog_sythe()                                'Synthesizer Programmieren
Declare Sub F_bereich                                   'Variable für 1 MHz-Frequenzbereich berechnen
Declare Sub Bereichs_variable                           'Variable für 1 MHz-Frequenzbereich anzeigen
Declare Sub Frequenz_variable                           'Variable für den Frequenzbereich anzeigen
Declare Sub Tx_synthe()                                 'Alle Parameter an den Snythesizer senden
Declare Sub Single_step()                               'Warteschleife für Single Step
Declare Sub Rd_pll()                                    'PLL-Regelschleife auslesen und anzeigen

Declare Function M_value(byval Adc_val As Word) As Byte     'In dieser Function wird der Mode-Wert bestimmt
Declare Function T1_value(byval Adc_val As Word) As Byte    'In dieser Function wird der Tastencode der T1-Tastengruppe bestimmt
Declare Function T2_value(byval Adc_val As Word) As Byte    'In dieser Function wird der Tastencode der T2-Tastengruppe bestimmt
```

```
Declare Function Tf_value(byval Adc_val As Word) As Byte     'In dieser Function wird der Tastencode der Tastatur bestimmt

Declare Sub Calc_ad9951                                      'DDS-Baustein Frequenz berechnen
Declare Sub Set_ad9951                                       'DDS-Baustein AD9951 programmieren
Declare Sub Transmit_dds                                     'In dieser Subroutine wird ein Byte an den DDS-Baustein übertragen
Declare Sub Transmit_dac()                                   'Diese Subroutine überträgt Daten an den DAC AD5611
Declare Sub Transmit_hc595()                                 'Diese Subroutine überrägt DAten an das Schieberegister 74HC595
Declare Sub Add_data                                         'Daten der Bedienelemente zum Sendestring zusammensetzen

Declare Sub Tastenfeldeingabe                                'Tastenfeldeingabe
Declare Sub Pll_synthesizer                                  'Testroutinen PLL-Synthesizer
Declare Sub Dac_data                                         'Daten für den DAC aufbereiten
Declare Sub Dac_tx                                           'In dieser Subroutine werden die Daten an den DAC aufbereiten
Declare Sub Ic9_data                                         'in dieser Subroutine wird der Setup des IC9 aufbereitet
Declare Sub Ic12_data                                        'in dieser Subroutine wird der Setup des IC12 aufbereitet
Declare Sub S_data                                           'in dieser Subroutine wird der Setup von S0 und S1 aufbereitet
Declare Sub Z_data                                           'in dieser Subroutine wird der Setup von S0 und S1 aufbereitet
Declare Sub N_data                                           'in dieser Subroutine wird der Setup von S0 und S1 aufbereitet
Declare Sub Print_sm                                         'Testroutine zum Ausdrucken von Parametern über die serielle Schnittstelle
Declare Sub Mode_schalter                                    'Mode-Schalter überwachen und LCD-Display aktualisieren
Declare Sub Displ_lcd_ic9                                    'Zeile 3 für IC9 am LCD-Display darstellen
Declare Sub Displ_lcd_ic12                                   'Zeile 3 für IC12 am LCD-Display darstellen
Declare Sub Displ_lcd_s_data                                 'Zeile 3 für S0 + S1 am LCD-Display darstellen
Declare Sub Displ_lcd_n_data                                 'Zeile 3 für N0 + N5 am LCD-Display darstellen
Declare Sub Displ_lcd_dac_data                               'Zeile 3 für DAC-Data am LCD-Display darstellen
Declare Sub Hmc10_4                                          'Die Parameter für den Baustein HMC698LP5 werden für 10,4 MHz Raster berechnet
Declare Sub Displ_lcd_hmc10_4_data                           'Hilfsroutine für Hmc10_4
Declare Sub Hmc13                                            'Die Parameter für den Baustein HMC698LP5 werden für 13 MHz Raster berechnet
Declare Sub Displ_lcd_hmc13_data                             'Hilfsroutine für Hmc13
Declare Sub Calc_hmc698                                      'Hier werden die Steuervariablen für den HMC698LP5 berechnet
Declare Sub Menue_moni                                       'Monitor-Menue darstellen
Declare Sub Menue_freq                                       'Frequenzeingabe
Declare Sub Menue_freq_start                                 'Menue Frequenzeingabe darstellen
Declare Sub Menue_frequenz                                   'Frequenz einstellen
Declare Sub P_debug                                          'Hilfsroutine für Debug
Declare Sub V_debug                                          'Variablen-Debug
Declare Sub Dac_value                                        'DAC-Value eingeben
Declare Sub Raster_10_4                                      'Raster 10,4 MHz einstellen
Declare Sub Raster_13_0                                      'Raster 13,0 MHz einstellen
Declare Sub Ic9_ausgangsteiler                               'Ausgangsteiler programmieren
Declare Sub Prt_sh()                                         'Daten an Schieberegister drucken
Declare Sub Rd_adc7                                          'ADC-Value auslesen = PLL Regelspannung

Declare Sub Para_synthe()                                    'Frequenz des Synthesizers programmieren


'Interrupts initialisieren ------------------------------------------------
On Int1 Ondr0                                                'Interrupt-Routine für Drehschalter
Config Int1 = Falling                                        'Interrupt 1 bei H/L-Flanke auslösen
Enable Int1                                                  'Externen Interrupt 1 einschalten
Open "com1:" For Binary As #1                                'Serielle Schnittstelle 1 öffnen
Open "com2:" For Binary As #2                                'Serielle Schnittstelle 2 öffnen

Enable Interrupts                                            'Interrupts global zulassen


Goto Hauptprogramm

'Interrupts ---------------------------------------------------------------
Ondr0:                                                       'Int1-Routine überwacht den Drehschalter "Tune"
  If Tp2 = 0 Then Tp2 = 1 Else Tp2 = 0                       'Toggle TP2 für <============================================ Testzwecke
  If Pind.6 = Pind.7 Then                                    'Rechtsdrehung
       Tune = Tune + 1
  Else                                                       'Linksdrehung
       Tune = Tune - 1
  End If
Return


'Hauptprogramm AVX-EVAL-Board ---------------------------------------- @@@
Hauptprogramm:

   Print #2 , "Steuerrechner Version 0.23_M --- Inbetriebnahme Synthesizer vollstaendig"
   Print #2 ,
   Print #1 , "Steuerrechner Version 0.23 M --- Inbetriebnahme Synthesizer vollstaendig"

   'Init Preselector
   Gd_val = 3                                                'Gain = 1 Dämpfung = ein
   Gd_val = Gd_val Xor 255                                   'negative Logik
   I2cstart                                                  'generiere Start
   I2cwbyte Pcf8574_ic13                                     'send slave Adresse
   I2cwbyte Gd_val
   I2cstop

   'Beleuchtung LCD-Display ein
   Portc.0 = 1
   'LCD-Display R/W auf write stellen
   Portc.2 = 0

   'AD-Wandler starten
   Start Adc

   'Variable setzen
   Nennfrequenz = 45030000                                   'Nennfrequenz nach Start
   Nennfrequenz_alt = 0                                      'Nennfrequenz alt rücksetzen
   Frq_bereich_alt = 0                                       'Frequenzbereich initialisieren

   Hz3 = 0                                                   'Schalterprellen auf 0 setzen

   'Testvariable rücksetzen
   Var1_alt = 0
   Var2_alt = 0
   Zae1_alt = 99
   Mode_sw_alt = 255

   'Tastatureingabe aus
   Tastatur = 0

   'InitLcd durchführen.
   Initlcd                                                   'LCD-Reset per Instruktion (HD4478-PowerOn-Reset reicht oft nicht)
   Cls                                                       'Anzeige löschen
   Cursor On                                                 'LCD-Cursor einschalten
```

```
    Locate 4 , 1 : Lcd "   Steuerrechner "                'Begrüßung auf dem LCD-Display
    Print #2 , "Start BAVARIX"
    Buglog = 1                                            'Logbook an
    Moni = 0
    Remote = 255                                          'Frequenzeinstellung aktivieren
    Ss = 0                                                'Single Step aus
 Do
    Select Case Moni
    Case 0
         Call Menue_moni                                  'Monitor-Menue darstellen
    Case 1
         Print #2 , "ERR 1"
    Case 2
         Print #2 , "ERR 2"
    Case 3
         Call Menue_frequenz                              'Frequenz einstellen
    Case 4                                                'DAC-Wert eingeben
         Call Dac_value                                   'Wert für DAC eingeben
         Moni = 0
    Case 5                                                '10,4 MHz Raster
         Call Raster_10_4                                 'Raster 10,4 MHz einstellen
         Moni = 0
    Case 6                                                '13,0 MHz Raster
         Call Raster_13_0                                 'Raster 13,0 MHz einstellen
         Moni = 0
    Case 7                                                'IC9 Ausgnagsteiler
         Call Ic9_ausgangsteiler                          'IC9 Ausgnagsteiler setzen
         Moni = 0
    Case 8
         Call Rd_adc7                                     'ADC7 auslesen und anzeigen
         Moni = 0
    Case 27                                               'ESC gedrückt
         Call Setup                                       'Setup ausführen
         Moni = 0
    Case Else
       Print #2 , "ERR 3"
    End Select

Loop

'-----------------------------------------------------------------------
Sub Menue_moni
  'Diese Subroutine stellt die Auswahl des Monitors dar.

    Pospar = 1 : Call P_debug                             '#1 Positionsparameter
    Print #2 ,
    Print #2 , "--- Monitor-Programm BAVARIX--- "
    Print #2 , "1 = Frequenzeinstellung        4 = 13,0 MHz Raster"
    Print #2 , "2 = DAC-Value eingeben         5 = IC9 Ausgangsteiler"
    Print #2 , "3 = 10,4 MHz Raster            6 = Analogspannung PLL-Regelschleife"
    Print #2 , "ESC = Setup"

    If Remote <> 255 Then
       Remote = 0                                         'Remote initialisieren
       While Remote = 0                                   'Solange, bis eine Taste gedrückt
            Remote = Inkey(#2)                            'Serielle Schnittstelle vom PC einlesen
       Wend
    End If
    Select Case Remote
       Case "1"                                           'Frequenzeinstellung
          Moni = 3
       Case "2"                                           'DAC-Value eingeben
          Moni = 4
       Case "3"                                           '10,4 MHz Raster
          Moni = 5
       Case "4"                                           '13,0 MHz Raster
          Moni = 6
       Case "5"                                           'IC9 Ausgangsteiler
          Moni = 7
       Case "6"                                           'Analogspannung PLL Regelschleife
          Moni = 8
       Case 27                                            'ESC gedrückt = Setup
          Moni = 27
       Case 255                                           'Startbedingung
          Moni = 3                                        'Frequenzeinstellung
    End Select
    Remote = 0                                            'Remote initialisieren

End Sub


'-----------------------------------------------------------------------
Sub Setup
  'In dieser Subroutine können verschiedenen Setups durchgeführt werden.

    Pospar = 2 : Call P_debug                             '#2 Positionsparameter
    Print #2 ,
    Print #2 , "       --- Setup ---"
    Print #2 , "1 = Log on    3 = Single Step on"
    Print #2 , "2 = Log off   4 = Single Step off"
    Print #2 , "CR = Exit"
    Print #2 ,
    While Remote <> 13                                    'solange, bis "e" gedrückt
       Remote = 0
       While Remote = 0                                   'Solange, bis eine Taste gedrückt
            Remote = Inkey(#2)                            'Serielle Schnittstelle vom PC einlesen
       Wend
       Select Case Remote
       Case "1"                                           'Log on
          Print #2 , "Log on"
          Buglog = 1
       Case "2"                                           'Log off
          Print #2 , "Log off"
          Buglog = 0
       Case "3"                                           'Single Step on
          Print #2 , "Single Step on"
          Ss = 1
       Case "4"                                           'Single Step off
          Print #2 , "Single Step off"
          Ss = 0
       Case "e"                                           'Exit
```

```
         'keine Funktion
      Case Else
         Print #2 , "keine Funktion"
      End Select
   Wend

End Sub


'--------------------------------------------------------------------
Sub Menue_frequenz
  'In diesem Menue kann man die Frequenz eingestellen.
  '@@@@
   Pospar = 3 : Call P_debug                              '#3 Positionsparameter
   Print #2 ,
   Print #2 , "       --- Frequenzeinstellung ---"
   Print #2 , "1 = <        2 = >        Ende mit CR"
   Locate Y , X : Lcd ""                                  'Cursor setzen
   Call Freq_aenderung                                    'Cursor-Position übergeben
   Frequenz = Nennfrequenz                                'Frequenzübergabe
   'Call Calc_ad9951                                       'DDS-Baustein programmieren
   Call Frequenzanzeige                                   'Frequenz am LCD-Display anzeigen
   Remote = 0                                             'Remote initialisieren
   While Remote = 0                                       'Solange, bis eine Taste gedrückt
        Remote = Inkey(#2)                                'Serielle Schnittstelle vom PC einlesen
        Cursor On Blink                                   'LCD-Cursor einschalten
        Locate Y , X : Lcd ""                             'Cursor setzen
        If Tune <> 0 Then
           Call Drehgeber                                 'Drehgeber eingelesen
           Call Freq_aenderung                            'Frequenzänderung berechnet
           Call N_frequenz                                'Nennfrequenz berechnen
           If Nennfrequenz <> Nennfrequenz_alt Then       'hat sich die Frequenz geändert?
               Frequenz = Nennfrequenz                    'Frequenzübergabe
               Call Para_synthe                           'Syntehsizer parametrieren
               Print #2 ,
               Print #2 , "1 = <        2 = >        Ende mit CR"
               Nennfrequenz_alt = Nennfrequenz            'letzte Frequenz am Display merken
           End If
        End If
   Wend

   Select Case Remote                                     'Eingabe auswerten
   Case "1"
        Dek_pos = Dek_pos + 1                             'Tastenposition inkrementieren
        If Dek_pos > 8 Then
           Dek_pos = 8
        End If
        Call Freq_aenderung                               'Cursor-Position richtig berechnen
        Remote = 0                                        'Zurück zur Frequenzeingabe
        Moni = 3                                          'Zurück zur Frequenzeingabe
   Case "2"
        If Dek_pos > 0 Then
            Dek_pos = Dek_pos - 1                         'Tastenposition dekrementieren
        End If
        Call Freq_aenderung                               'Cursor-Position richtig berechnen
        Remote = 0                                        'Zurück zur Frequenzeingabe
        Moni = 3                                          'Zurück zur Frequenzeingabe
   Case 13
        Moni = 0                                          'Frequenzeingabe beenden
   Case Else
        'Do nothing
   End Select

End Sub


'--------------------------------------------------------------------
Sub Dac_value
  'In dieser Subroutine wird der Wert für den DAC eingegeben.

   Pospar = 4 : Call P_debug                              '#4 Positionsparameter
   Print #2 ,
   Print #2 , "--- DAC-Value eingeben ---"
   Print #2 , "Trestate OFF = +     Trestate ON = -      Ende mit CR"

   Hz4 = 0                                                'Subroutine initialisieren
   While Remote <> 13                                     'Solange, bis die Taste CR gedrückt wird
      Remote = Inkey(#2)                                  'Serielle Schnittstelle vom PC einlesen

      If Remote = "+" Then                                'DAC aktiv
         Dac_aktiv = 1
      Remote = 13                                         'beenden
      End If

      If Remote = "-" Then                                'DAC in Trstate
         Dac_aktiv = 0
         Remote = 13                                      'beenden
      End If

      If Remote > 47 Then                                 'Testen, ob Charakter gültig
         If Remote < 58 Then
            Wert = Val(remote)                            'Ascii-Zeichen in nuimmerischen Wert umwandeln
            Print #2 , Wert ;                             'Eingabezeichen darstellen
            Hz4 = Hz4 + 1                                 'Anzahl gültiger Zahlen zählen
            Select Case Hz4                               'Eingabe zusammensetzen
               Case 1
                  Dac_val = Wert
               Case 2
                  Dac_val = Dac_val * 10
                  Dac_val = Dac_val + Wert
               Case 3
                  Dac_val = Dac_val * 10
                  Dac_val = Dac_val + Wert
               Case 4
                  Dac_val = Dac_val * 10
                  Dac_val = Dac_val + Wert
            End Select
         End If
      End If
      If Dac_val > 1023 Then                              'Eingabe beenden
         Dac_val = 1023                                   'Eingabe auf gültiges Maximum begrenzen
         Remote = 13
```

```
      End If
      If Hz4 = 4 Then                                   'Eingabe beenden
         Remote = 13
      End If
   Wend

    If Dac_aktiv = 0 Then
       Dac_data = 49152                                 'DAC in Tristate schalten
       Call Transmit_dac                                'Daten an den DAC übertragen
       Call Add_data                                    'Daten zur Steuerung übertragen
       Locate 4 , 1 : Lcd "&"                           'DAC-Status darstellen (DAC = tristate)
    Else
       Dac_data = Dac_val * 16                          'Daten übernehmen + Shift 4 x left zur korrekten Positionierung
       Call Transmit_dac                                'Daten an den DAC übertragen
       Call Add_data                                    'Daten zur Steuerung übertragen
       Locate 4 , 1 : Lcd "*"                           'DAC-Status darstellen (DAC = aktiv)
    End If

End Sub


'-----------------------------------------------------------------------
Sub Raster_10_4
    'In dieser Subroutine werden die Parametrierbits für den Baustein
    'HMC698LP5 für ein Raster mit 10,4 MHz berechnet und die Daten zur
    'Programmierung dieses Bausteins weitergeleitet.
    'Output:    der Baustein HMC698LP5 wird programmiert


   Pospar = 5 : Call P_debug                            '#5 Positionsparameter
   Print #2 ,
   Print #2 , "--- 10,4 MHz Raster ---"
   Print #2 , "Setup eingeben: (50 ... 65)  Ende mit CR"
   Hz4 = 0                                              'Subroutine initialisieren
   While Remote <> 13                                   'Solange, bis die Taste CR gedrückt wird
      Remote = Inkey(#2)                                'Serielle Schnittstelle vom PC einlesen
      If Remote > 47 Then                               'Testen, ob Charakter gültig
         If Remote < 58 Then
            Wert = Val(remote)                          'Ascii-Zeichen in nuimmerischen Wert umwandeln
            Print #2 , Wert ;                           'Eingabezeichen darstellen
            Hz4 = Hz4 + 1                               'Anzahl gültiger Zahlen zählen
            Select Case Hz4                             'Eingabe zusammensetzen
               Case 1
                  Zae_hmc = Wert
               Case 2
                  Zae_hmc = Zae_hmc * 10
                  Zae_hmc = Zae_hmc + Wert
            End Select
         End If
      End If
      If Zae_hmc > 65 Then                              'Eingabe beenden
         Zae_hmc = 65                                   'Eingabe auf gültiges Maximum begrenzen
         Remote = 13
      End If
      If Hz4 = 2 Then                                   'Eingabe beenden
         Remote = 13
      End If
   Wend
   If Zae_hmc < 50 Then
      Zae_hmc = 50
   End If
   Zae_ic12 = 2                                         'IC12 setzen (10,4 MHz-Raszer)
   Call Calc_hmc698                                     'Daten für den HMC698 berechnen

End Sub


'-----------------------------------------------------------------------
Sub Raster_13_0
    'In dieser Subroutine werden die Parametrierbits für den Baustein
    'HMC698LP5 für ein Raster mit 13,0 MHz berechnet und die Daten zur
    'Programmierung dieses Bausteins weitergeleitet.
    'Output:    der Baustein HMC698LP5 wird programmiert

   Pospar = 6 : Call P_debug                            '#6 Positionsparameter
   Print #2 ,
   Print #2 , "--- 13,0 MHz Raster ---"
   Print #2 , "Setup eingeben: (40 ... 53)  Ende mit CR"
   Hz4 = 0                                              'Subroutine initialisieren
   While Remote <> 13                                   'Solange, bis die Taste CR gedrückt wird
      Remote = Inkey(#2)                                'Serielle Schnittstelle vom PC einlesen
      If Remote > 47 Then                               'Testen, ob Charakter gültig
         If Remote < 58 Then
            Wert = Val(remote)                          'Ascii-Zeichen in nuimmerischen Wert umwandeln
            Print #2 , Wert ;                           'Eingabezeichen darstellen
            Hz4 = Hz4 + 1                               'Anzahl gültiger Zahlen zählen
            Select Case Hz4                             'Eingabe zusammensetzen
               Case 1
                  Zae_hmc = Wert
               Case 2
                  Zae_hmc = Zae_hmc * 10
                  Zae_hmc = Zae_hmc + Wert
            End Select
         End If
      End If
      If Zae_hmc > 53 Then                              'Eingabe beenden
         Zae_hmc = 53                                   'Eingabe auf gültiges Maximum begrenzen
         Remote = 13
      End If
      If Hz4 = 2 Then                                   'Eingabe beenden
         Remote = 13
      End If
   Wend
   If Zae_hmc < 40 Then
      Zae_hmc = 40
   End If

   Call Calc_hmc698                                     'Daten für den HMC698 berechnen

End Sub
```

```
'--------------------------------------------------------------------
Sub Ic9_ausgangsteiler
    'Diese Subroutine bereitet die Daten für IC9 (Ausgangsteiler) auf dem Synthesizer-Board
    'PLL vor.
    'Output:    Bitmuster übertragen an IC9

   Pospar = 7 : Call P_debug                                '#7 Positionsparameter
   Print #2 ,
   Print #2 , "--- IC9 Ausgangsteiler ---"
   Print #2 , "Setup eingeben: (0 ... 3)  Ende mit CR"
   Hz4 = 0                                                  'Subroutine initialisieren
   While Remote <> 13                                       'Solange, bis die Taste CR gedrückt wird
      Remote = Inkey(#2)                                    'Serielle Schnittstelle vom PC einlesen
      If Remote > 47 Then                                   'Testen, ob Charakter gültig
         If Remote < 58 Then
            Wert = Val(remote)                              'Ascii-Zeichen in nuimmerischen Wert umwandeln
            Print #2 , Wert ;                               'Eingabezeichen darstellen
            Hz4 = Hz4 + 1                                   'Anzahl gültiger Zahlen zählen
            Select Case Hz4                                 'Eingabe zusammensetzen
               Case 1
                  Zae_ic9 = Wert
            End Select
         End If
      End If
      If Zae_ic9 > 3 Then                                   'Eingabe beenden
         Zae_ic9 = 3                                        'Eingabe auf gültiges Maximum begrenzen
         Remote = 13
      End If
      If Hz4 = 1 Then                                       'Eingabe beenden
         Remote = 13
      End If
   Wend
   If Zae_ic9 < 0 Then
      Zae_ic9 = 0
   End If
   Call Add_data                                            'Daten übertragen
End Sub


'--------------------------------------------------------------------
Sub Rd_adc7
   'Diese Routine liest des ADC7 ein und stellt das Ergebnis im Monitorprogramm dar.

   Pospar = 8 : Call P_debug                                '#8 Positionsparameter
   Print #2 ,
   Print #2 , "--- PLL-Regelschleife ---"
   Print #2 , " Analogwert:           DAC:            Ende = CR"
   Adc_val_alt = 2000                                       'Routine initialisieren
   While Remote <> 13                                       'Solange, bis die Taste CR gedrückt wird
      Remote = Inkey(#2)                                    'Serielle Schnittstelle vom PC einlesen
      Channel = 7                                           'ADC 7 Analogspannung PLL-Regelschleife einlesen
      Adc_val = Getadc(channel)                             'ADC initialisiseren
      Adc_val = Getadc(channel)                             'ADC starten und Analogwert digitalisieren
      If Adc_val_alt <> Adc_val Then
         If Adc_val <> 0 Then
            Adc_mvolt = 48 * Adc_val
            Adc_mvolt = Adc_mvolt \ 10
            Print #2 , Adc_mvolt ; : Print #2 , " mV = " ;  'ADC-Value in mV
            Print #2 , Adc_val ; : Print #2 , "  ---  " ;   'ADC-Value als 12 Bit-Wert ausgeben
            Print #2 , "DAC-Value = " ; : Print #2 , Dac_val       'DAC-Value ausgeben
            Adc_val_alt = Adc_val                           'alten Wert übergeben
         End If
      End If
      If Lock_loop_2 = 1 Then
         Locate 4 , 2 : Lcd "L"                             'Lockbit Lock Loop 2 darstellen lock
      Else
         Locate 4 , 2 : Lcd "U"                             'Lockbit Lock Loop 2 darstellen unlock
      End If
   Wend

End Sub


'--------------------------------------------------------------------
Function M_value(byval Adc_val As Word) As Byte
'  Diese Function bekommt den AD-Wandler-Wert vom Mode-Schalter
'  und setzt ihn in einen Mode-Wert um.
'  Input:   Adc_val  0 bis 1023
'  Output:  M_value  Mode 0 bis Mode 11

  Pospar = 9 : Call P_debug                                 '#9 Positionsparameter
  Select Case Adc_val
    Case 71 To 121
      M_value = 1                                           'Mode 1  (96)
    Case 167 To 217
      M_value = 2                                           'Mode 2  (192)
    Case 230 To 280
      M_value = 3                                           'Mode 3  (255)
    Case 327 To 377
      M_value = 4                                           'Mode 4  (352)
    Case 423 To 473
      M_value = 5                                           'Mode 5  (448)
    Case 486 To 536
      M_value = 6                                           'Mode 6  (511)
    Case 566 To 616
      M_value = 7                                           'Mode 7  (591)
    Case 743 To 793
      M_value = 8                                           'Mode 8  (768)
    Case 743 To 793
      M_value = 8                                           'Mode 9  (768)
    Case 871 To 921
      M_value = 10                                          'Mode 10 (896)
    Case 935 To 985
      M_value = 11                                          'Mode 11 (960)
    Case Else
      M_value = 255                                         'ungültiger Wert
  End Select

End Function
```

```
'--------------------------------------------------------------------
Function T1_value(byval Adc_val As Word) As Byte
'  Diese Function bekommt den AD-Wandler-Wert von der Tastengruppe
'  Frequency und Memory und setzt ihn in T1_value um.
'  Input:   Adc_val  0 bis 1023
'  Output:  T1_value  1 bis 4

  Pospar = 10 : Call P_debug                              '#10 Positionsparameter
  Select Case Adc_val
    Case 180 To 213                                       'Tastengruppe T1 Save  (199)
       If Hz3 = 50 Then                                   'abwarten, bis Schalterprellen beendet
         T1_value = 4                                     'setze Tastencode
         Hz3 = 0                                          'Zähler Schalterprellen auf 0 setzen
       Else
         Hz3 = Hz3 + 1                                    'Schalterprellen abwarten
       End If
    Case 370 To 410                                       'Tastengruppe T1 Load  (399)
      T1_value = 3
      Hz3 = 0                                             'Zähler Schalterprellen auf 0 setzen
    Case 540 To 600                                       'Tastengruppe T1 >  (591)
      If T1_value = 255 Then
          If Dek_pos > 0 Then
              Dek_pos = Dek_pos - 1                       'Tastenposition dekrementieren
          End If
      End If
      T1_value = 2
    Case 740 To 827                                       'Tastengruppe T1 <  (768)
      If T1_value = 255 Then
          Dek_pos = Dek_pos + 1                           'Tastenposition inkrementieren
          If Dek_pos > 8 Then
             Dek_pos = 8
          End If
      End If
      T1_value = 1
    Case Else
      T1_value = 255                                      'ungültiger Wert
  End Select
  Adc_val = 0                                             'ADC_VAL rücksetzen

End Function


'--------------------------------------------------------------------
Function T2_value(byval Adc_val As Word) As Byte
'  Diese Function bekommt den AD-Wandler-Wert von der Tastengruppe
'  F1 bis F4 und setzt ihn in einen T2_value um.
'  Input:   Adc_val  0 bis 1023
'  Output:  T2_value  1 bis 4

  Pospar = 11 : Call P_debug                              '#11 Positionsparameter
  Select Case Adc_val
    Case 180 To 213
      T2_value = 4                                        'Tastengruppe T2 F4
      Nennfrequenz = 10000000                             'Nennfrequenz in Hz, wenn F4 gedrückt
    Case 370 To 410
      T2_value = 3                                        'Tastengruppe T2 F3
      Nennfrequenz = 15000000                             'Nennfrequenz in Hz, wenn F3 gedrückt
    Case 540 To 600
      T2_value = 2                                        'Tastengruppe T2 F2
      Nennfrequenz = 20000000                             'Nennfrequenz in Hz, wenn F2 gedrückt
    Case 740 To 827
      T2_value = 1                                        'Tastengruppe T2 F1
      Wait 1                                              'CPU 1 Sekunde warten lassen
      'Nennfrequenz = 25000000                             'Nennfrequenz in Hz, wenn F1 gedrückt
    Case Else
      T2_value = 255                                      'ungültiger Wert
  End Select

  If T2_value <> 255 Then
    Posp = 2
  End If

  Adc_val = 0                                             'ADC_VAL rücksetzen

End Function


'--------------------------------------------------------------------
Function Tf_value(byval Adc_val As Word) As Byte
'  Diese Function bekommt den AD-Wandler-Wert vom Tastenfeld und setzt
'  ihn in einen Tastenwert Tast_value um.
'  Input:   Adc_val  0 bis 1023
'  Output:  Tast_value  0 bis 9 + * + #

  Pospar = 12 : Call P_debug                              '#12 Positionsparameter
  Select Case Adc_val
    Case 362 To 412
      Tast_value = 1                                      '1    (387)
    Case 743 To 771
      Tast_value = 2                                      '2    (768)
    Case 995 To 1045
      Tast_value = 3                                      '3    (1020)
    Case 293 To 343
      Tast_value = 4                                      '4    (318)
    Case 610 To 660
      Tast_value = 5                                      '5    (632)
    Case 935 To 985
      Tast_value = 6                                      '6    (960)
    Case 223 To 273
      Tast_value = 7                                      '7    (248)
    Case 518 To 568
      Tast_value = 8                                      '8    (543)
    Case 871 To 921
      Tast_value = 9                                      '9    (896)
    Case 455 To 505
      Tast_value = 0                                      '0    (480)
    Case 134 To 184
      Tast_value = 11                                     '*    (159)
    Case 772 To 800
      Tast_value = 12                                     '#    (775)
    Case Else
      Tast_value = 255                                    'ungültiger Wert
  End Select
```

```
  Adc_val = 0                                          'ADC_VAL rücksetzen

End Function


'---------------------------------------------------------------------
Sub Print_sm
  'In dieser Subroutine werden Testparameter auf das LCD-Display geschrieben
  'Input:   Var1 = Wordvariable 1 zum Printen
  '         Var2 = Wordvariable 2 zum Printen

     If Var1 <> Var1_alt Then
        Locate 2 , 1 : Lcd "1="
        Locate 2 , 4 : Lcd "     "
        Locate 2 , 4 : Lcd Var1
        Var1_alt = Var1                                'alten Wert zwischenspeichern
     End If

     If Var2 <> Var2_alt Then
        Locate 2 , 9 : Lcd "2="
        Locate 2 , 12 : Lcd "     "
        Locate 2 , 12 : Lcd Var2
        Var2_alt = Var2                                'alten Wert zwischenspeichern
     End If

End Sub


'---------------------------------------------------------------------
Sub P_debug
  'In dieser Subroutine werden Testparameter auf die serielle Schnittstelle COM1
  'geschrieben und zum PC gesendet. Abhängig vom Steuerflag "Buglog" kann diese
  'Funktion ein oder ausgeschaltet werden.
  'Input:   Pospar   = Positionskennung Subroutine
  '         Buglog = Schalter on/off Debug
     If Buglog <> 0 Then
         Posipar = Str(pospar)
         Print #1 , " #" ; : Print #1 , Posipar
     End If

End Sub


'---------------------------------------------------------------------
Sub V_debug
  'In dieser Subroutine werden Variable auf die serielle Schnittstelle COM1
  'geschrieben und zum PC gesendet. Abhängig vom Steuerflag "Buglog" kann diese
  'Funktion ein oder ausgeschaltet werden.
  'Input:   Position   = Positionskennung Variable
  '         Vari       = Variable
  '         Buglog = Schalter on/off Debug

     If Buglog <> 0 Then
         Posipar = Str(position)
         Print #1 , "Pos = " ; : Print #1 , Posipar ;
         Print #1 , "  Par = " ; : Print #1 , Vari ;
         If Vari < 256 Then
            Hz5 = Vari
            Print #1 , " = Character ( " ; : Print #1 , Chr(hz5);
            Print #1 , " )"
         Else
            Print #1 ,
         End If
     End If

End Sub


'---------------------------------------------------------------------
Sub Pll_synthesizer
   'In dieser Subroutine werden Testdaten für den PLL-Synthesizer in Abhängigkeit
   'des Mode-Schalters erzeugt und an die jeweiligen ICs geschickt.
   'Input:     Mode_sw = Stellung Mode-Schalter
   Pospar = 13 : Call P_debug                          '#13 Positionsparameter
   Select Case Mode_sw
     Case 20                                           'Mode 20 not used  DAC * 1
       Call Drehgeber
       Zae1 = Zae1 + Zaehler                           'letzte Drehbewegung übernehmen
       Zaehler = 0
       Call Dac_data
     Case 5                                            'Mode 5  DAC * 10
       Call Drehgeber
       Zaehler_z = Zaehler * 10                        'x 10
       Zae1 = Zae1 + Zaehler_z                         'letzte Drehbewegung übernehmen
       Zaehler = 0
       Call Dac_data
     Case 4                                            'Mode 4  DAC * 100
       Call Drehgeber
       Zaehler_z = Zaehler * 100                       'x 100
       Zae1 = Zae1 + Zaehler_z                         'letzte Drehbewegung übernehmen
       Zaehler = 0
       Call Dac_data
     Case 6                                            'Mode 6  IC9 Ausgangsteiler
       Call Drehgeber
       Zae_ic9 = Zae_ic9 + Zaehler                     'letzte Drehbewegung übernehmen
       Zaehler = 0
       Call Ic9_data                                   'IC9
     Case 22                                           'Mode 22 not used  IC12
       Call Drehgeber
       Zae_ic12 = Zae_ic12 + Zaehler                   'letzte Drehbewegung übernehmen
       Zaehler = 0
       Call Ic12_data                                  'IC12
     Case 23                                           'Mode 23 not used  S0 ... S1
       Call Drehgeber
       Zae_s = Zae_s + Zaehler                         'letzte Drehbewegung übernehmen
       Zaehler = 0
       Call S_data                                     'S0 + S1
     Case 24                                           'Mode 24 not used  N0 ... N5
       Call Drehgeber
```

```
        Call Drehgeber
        Zae_n = Zae_n + Zaehler                 'letzte Drehbewegung übernehmen
        Zaehler = 0
        Call N_data
      Case 9                                    'Mode 9 not used  (768)
        'not used
      Case 2                                    'Mode 2 HMC698LP5 10,4 MHz
        Call Drehgeber
        Zae_hmc = Zae_hmc + Zaehler             'letzte Drehbewegung übernehmen
        Zaehler = 0
        Call Hmc10_4
      Case 3                                    'Mode 3 HMC698LP5 13 MHz
        Call Drehgeber
        Zae_hmc = Zae_hmc + Zaehler             'letzte Drehbewegung übernehmen
        Zaehler = 0
        Call Hmc13
      Case Else
        Mode_sw = 255                           'ungültiger Wert
    End Select

End Sub


'--------------------------------------------------------------------
Sub Ic9_data
    'Diese Subroutine bereitet die Daten für IC9 (Ausgangsteiler) auf dem Synthesizer-Board
    'PLL vor und zeigt diese am LCD-Display an.
    'Input: Zae_ic9
    'Output:    Bitmuster übertragen an IC9

    Pospar = 14 : Call P_debug                  '#14 Positionsparameter
    If Zae_ic9 > 3 Then Zae_ic9 = 3             'Überlauf verhindern
    If Zae_ic9 < 0 Then Zae_ic9 = 0             'Unterlauf verhindern
    If Zae_ic9 <> Zae_ic9_alt Then              'Zählerstellung IC9 überwachen
      Call Displ_lcd_ic9
    End If
    If Mode_sw <> Mode_sw_alt Then              'Schalterstellung Mode überwachen
      Call Displ_lcd_ic9
    End If
    Call Add_data                               'Daten übertragen
    Zae_ic9_alt = Zae_ic9                       'alte Werte zwischenspeichern
    Mode_sw_alt = Mode_sw

End Sub

Sub Displ_lcd_ic9
  'Print Zeile 3 LCD-Display für IC9
  Locate 3 , 1 : Lcd "                "         'Displayzeile löschen
  Locate 3 , 1 : Lcd "6"                        'Schalterstellung "IC9"
  Locate 3 , 3 : Lcd "IC9 = "
  Select Case Zae_ic9
    Case 0
      Locate 3 , 11 : Lcd "00B"
    Case 1
      Locate 3 , 11 : Lcd "01B"
    Case 2
      Locate 3 , 11 : Lcd "10B"
    Case 3
      Locate 3 , 11 : Lcd "11B"
  End Select

End Sub


'--------------------------------------------------------------------
Sub Ic12_data
    'Diese Subroutine bereitet die Daten für IC12 auf dem Synthesizer-Board PLL vor
    'und zeigt diese am LCD-Display an.
    'Input: Zae_ic12
    'Output:    Bitmuster übertragen an IC12

    Pospar = 15 : Call P_debug                  '#15 Positionsparameter
    If Zae_ic12 > 3 Then Zae_ic12 = 3
    If Zae_ic12 < 0 Then Zae_ic12 = 0
    If Zae_ic12 <> Zae_ic12_alt Then
      Call Displ_lcd_ic12
    End If
    If Mode_sw <> Mode_sw_alt Then              'Schalterstellung Mode überwachen
      Call Displ_lcd_ic12
    End If
    Call Add_data                               'Daten übertragen
    Zae_ic12_alt = Zae_ic12                     'alte Werte zwischenspeichern
    Mode_sw_alt = Mode_sw

End Sub


Sub Displ_lcd_ic12
  'Print Zeile 3 LCD-Display für IC12

    Locate 3 , 1 : Lcd "                "       'Displayzeile löschen
   Locate 3 , 1 : Lcd "6"                       'Schalterstellung "IC12"
    Locate 3 , 3 : Lcd "IC12 = "
    Select Case Zae_ic12
      Case 0
        Locate 3 , 11 : Lcd "00B"
      Case 1
        Locate 3 , 11 : Lcd "01B"
      Case 2
        Locate 3 , 11 : Lcd "10B"
      Case 3
        Locate 3 , 11 : Lcd "11B"
    End Select

End Sub

'--------------------------------------------------------------------
Sub S_data
    'Diese Subroutine bereitet die Daten für S0 und S1 auf dem Synthesizer-Board PLL
    'vor und zeigt diese am LCD-Display an.
    'Input: Zae_s
    'Output:    Bitmuster übertragen an IC16
```

```
    Output:    Bitmuster übertragen an IC16

    Pospar = 16 : Call P_debug                          '#16 Positionsparameter
    If Zae_s > 3 Then Zae_s = 3
    If Zae_s < 0 Then Zae_s = 0
    If Zae_s <> Zae_s_alt Then
      Call Displ_lcd_s_data
    End If
    If Mode_sw <> Mode_sw_alt Then                      'Schalterstellung Mode überwachen
      Call Displ_lcd_s_data
    End If
    Call Add_data                                       'Daten übertragen
    Zae_s_alt = Zae_s                                   'alte Werte zwischenspeichern
    Mode_sw_alt = Mode_sw

End Sub


Sub Displ_lcd_s_data
  'Print Zeile 3 LCD-Display für S0 + S1 Data

    Locate 3 , 1 : Lcd "                "               'Displayzeile löschen
    Locate 3 , 1 : Lcd "7"                              'Schalterstellung "IC12"
    Locate 3 , 3 : Lcd "S0+1 = "
    Locate 3 , 11 : Lcd "       "
    Select Case Zae_s
      Case 0
        Locate 3 , 11 : Lcd "00B"
      Case 1
        Locate 3 , 11 : Lcd "01B"
      Case 2
        Locate 3 , 11 : Lcd "10B"
      Case 3
        Locate 3 , 11 : Lcd "11B"
    End Select

End Sub
```

```
'--------------------------------------------------------------------
Sub N_data
    'Diese Subroutine bereitet die Daten für N0 bis N5 auf dem Synthesizer-Board PLL
    'vor und zeigt diese am LCD-Display an.

    'Input: Zae_n
    'Output:    Bitmuster übertragen an IC16

    Pospar = 17 : Call P_debug                          '#17 Positionsparameter
    If Zae_n > 63 Then Zae_n = 63                       'Überlauf verhindern
    If Zae_n < 0 Then Zae_n = 0                         'negativen Überlauf verhindern
    If Zae_n <> Zae_n_alt Then                          'Erkennungstext darstellen
      Call Displ_lcd_n_data
    End If
    If Mode_sw <> Mode_sw_alt Then                      'Schalterstellung Mode überwachen
      Call Displ_lcd_n_data
    End If
    Zae_n_alt = Zae_n                                   'alten Wert zwischenspeichern
    Call Add_data                                       'Daten übertragen
    Mode_sw_alt = Mode_sw                               'Alten Wert zwischenspeichern

End Sub

Sub Displ_lcd_n_data
  'Print Zeile 3 LCD-Display für N0 + N5 Data

  Locate 3 , 1 : Lcd "                "                 'Displayzeile löschen
  Locate 2 , 1 : Lcd "                "                 'Displayzeile löschen
  Locate 3 , 1 : Lcd "8"                                'Schalterstellung "IC12"
  Locate 3 , 3 : Lcd "IC16 = "
  Locate 3 , 11 : Lcd "      "
  For I = 5 To 0 Step -1                                '6 Bit ausgeben, MSB zuerst
    Hz1 = 2 ^ I                                         'richtiges Bit zur Übertragung auswählen
    Tx_bit = Zae_n And Hz1
    If Tx_bit <> 0 Then
      Lcd "1"
    Else
      Lcd "0"
    End If
  Next I
  Print #2 ,

End Sub


'--------------------------------------------------------------------
Sub Hmc10_4
   'In dieser Subroutine werden die Parametrierbits für den Baustein
   'HMC698LP5 für ein Raster mit 10,4 MHz berechnet und die Daten zur
   'Programmierung dieses Bausteins weitergeleitet.
   'Input:     M_value
   'Output:    der Baustein HMC698LP5 wird programmiert

   Pospar = 18 : Call P_debug                           '#18 Positionsparameter
   If Zae_hmc > 65 Then Zae_hmc = 65                    'Überlauf verhindern
   If Zae_hmc < 50 Then Zae_hmc = 50                    'negativen Überlauf verhindern
   If Zae_hmc <> Zae_hmc_alt Then                       'Erkennungstext darstellen
       Call Displ_lcd_hmc10_4_data
       Zae_hmc_alt = Zae_hmc                            'alten Wert merken
   End If
   If Mode_sw <> Mode_sw_alt Then                       'Schalterstellung Mode überwachen
       Call Displ_lcd_hmc10_4_data
       Mode_sw_alt = Mode_sw                            'Alten Wert zwischenspeichern
   End If
   Zae_ic12 = 2                                         'IC12 setzen
   Call Calc_hmc698                                     'Daten für den HMC698 berechnen

End Sub

Sub Displ_lcd_hmc10_4_data
  'Print Zeile 3 LCD-Display für 10,4 MHz
  Locate 3 , 1 : Lcd "                "                 'Displayzeile löschen
  Locate 3 , 1 : Lcd "2"                                'Schalterstellung "10,4 MHz"
```

```
   Locate 3 , 7 : Lcd "N = "
   Locate 3 , 11 : Lcd Zae_hmc
   Locate 2 , 1 : Lcd "                "                    'Displayzeile löschen
   Locate 2 , 1 : Lcd "10,4 MHz Raster"                     'Schalterstellung "13 MHz"

End Sub


'-----------------------------------------------------------------------
Sub Hmc13
   'In dieser Subroutine werden die Parametrierbits für den Baustein
   'HMC698LP5 für ein Raster mit 13 MHz berechnet und die Daten zur
   'Programmierung dieses Bausteins weitergeleitet.
   'Input:     M_value
   'Output:    der Baustein HMC698LP5 wird programmiert

   Pospar = 19 : Call P_debug                               '#19 Positionsparameter
   If Zae_hmc > 53 Then Zae_hmc = 53                        'Überlauf verhindern
   If Zae_hmc < 40 Then Zae_hmc = 40                        'negativen Überlauf verhindern
   If Zae_hmc <> Zae_hmc_alt Then                           'Erkennungstext darstellen
      Call Displ_lcd_hmc13_data
      Zae_hmc_alt = Zae_hmc                                 'alten Wert merken
   End If
   If Mode_sw <> Mode_sw_alt Then                           'Schalterstellung Mode überwachen
      Call Displ_lcd_hmc13_data
      Mode_sw_alt = Mode_sw                                 'Alten Wert zwischenspeichern
   End If
   Zae_ic12 = 1                                             'IC12 setzen
   Call Calc_hmc698                                         'Daten für den HMC698 berechnen

End Sub

Sub Displ_lcd_hmc13_data
  'Print Zeile 3 LCD-Display für 13 MHz
  Locate 3 , 1 : Lcd "                "                    'Displayzeile löschen
  Locate 3 , 1 : Lcd "3"                                    'Schalterstellung "13 MHz"
  Locate 3 , 7 : Lcd "N = "
  Locate 3 , 11 : Lcd Zae_hmc
  Locate 2 , 1 : Lcd "                "                    'Displayzeile löschen
  Locate 2 , 1 : Lcd "13 MHz Raster"                        'Schalterstellung "13 MHz"

End Sub


'-----------------------------------------------------------------------
Sub Calc_hmc698
  'In dieser Subroutine werden die Ansteuerdaten für den Frequenzteiler HMC698LP5
  'berechnet und an den Baustein übertragen.
  'Input:   Zae_hmc
  'Output:  Zae_hmc    =  N0 .... N5
  '         Zae_s     =  S0 + S1

   Pospar = 20 : Call P_debug                               '#20 Positionsparameter
   X1 = Zae_hmc                                             'Frequenzteilerwert übernehmen
   B = 0                                                    'Hilfsvariable
   While X1 > -1                                            'test expression
      X1 = X1 - 4
     Incr B                                                 'increase by one
   Wend                                                     'continue loop
   B = B - 2                                                'Korrektur für Tabelle
   Zae_n = B                                                'B übergeben; Zae_n = N0 .... N5
   B = B + 1                                                'Hardware-Korrektur
   B = B * 4                                                'Berechnung A
   Zae_s = Zae_hmc - B                                      'A übergeben; Zae_s = S0 + S1;  Zae_hmc = N
   Call Add_data                                            'Daten übertragen

End Sub


'-----------------------------------------------------------------------
Sub Add_data
   'In dieser Subroutine werde die Daten der einzelnen ICs zusammengesetzt
   'und dann an das Synthesizer-Board PLL geschickt.
   'Input:     Zae_ic9  =  A + B
   '           Zae_ic12 =  A + B
   '           Zae_s    =  S0 + S1
   '           Zae_n    =  N0 .... N5
   'Output:    Daten werden an die Hardware gesendet
   '
   'Datenformat der Übertragung zum Synthesizer-Board:
   '+----+----+----+----+----+----+----+----+-------+-------+--------+--------+--------------+----+----+----+
   '| N5 | N4 | N3 | N2 | N1 | N0 | S1 | S0 | IC9-A | IC9-B | IC12-B | IC12-A | DAC-Tristate | no | no | no |
   '+----+----+----+----+----+----+----+----+-------+-------+--------+--------+--------------+----+----+----+
   '  D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |   D8  |  D9   |  D10   |  D11   |     D13      | D14| D15| D16|

   Pospar = 21 : Call P_debug                               '#21 Positionsparameter
   'Speicher initialisieren
   Data_595 = 0
   'Wertigkeiten der einzelnen Bits zusammenzählen
   Data_595 = Zae_ic9 * 64                                  'Wertigkeit IC9 bestimmen

   Add_data_hw = Dac_aktiv * 8                              'Wertigkeit für die Übertragung bestimmen
   Data_595 = Data_595 + Add_data_hw                        'Wertigkeit Tristate-Flag dazuaddieren

   Zae_m = Zae_ic12                                         'Wert zwischenspeichern
   Add_data_hw = Zae_m * 16                                 'Wertigkeit für die Übertragung bestimmen
   Data_595 = Data_595 + Add_data_hw                        'Wertigkeit IC12 dazuaddieren

   Add_data_hw = Zae_s * 256                                'Wertigkeit für die Übertragung bestimmen
   Data_595 = Data_595 + Add_data_hw                        'Wertigkeit Data S0 + S1 dazuaddieren

   Add_data_hw = Zae_n * 1024                               'Wertigkeit für die Übertragung bestimmen
   Data_595 = Data_595 + Add_data_hw                        'Wertigkeit N0 ... N5 dazuaddieren

   Call Transmit_hc595

End Sub


'-----------------------------------------------------------------------
Sub Dac_data
   'Diese Subroutine bereitet die Daten für den DAC AD5611 auf dem
   'PLL-Synthesizer-Board auf und schreibt sie in einen Zwischenspeicher.
```

```
    'Das Datenformat: PD1 PD0 D9 D8 .... D1 D0 x x x x
    'PD1 wird zuerst übertragen, dann folgen die anderen Daten
    'PD1   PD0
    ' 0     0    =   Normal Operation
    ' 0     1    =   1k Ohm to GND
    ' 1     0    =   100k Ohm to GND
    ' 1     1    =   Three-State

    'Input:  Zae1 = ADC-Datenwert
    'Output: DAC_VAL

    Pospar = 22 : Call P_debug                          '#22 Positionsparameter
    If Zae1 > 1023 Then Zae1 = 1023
    If Zae1 < 0 Then Zae1 = 0
    If Zae1 <> Zae1_alt Then                            'Drehbewegung löschen
       Call Displ_lcd_dac_data
    End If
    If Mode_sw <> Mode_sw_alt Then                      'Schalterstellung Mode überwachen
      Call Displ_lcd_dac_data
    End If
    Zae1_alt = Zae1                                     'alten Wert zwischenspeichern
    Zae1_data = Zae1                                    'Normalmode = DAC-Daten + 0
    Dac_val = Zae1_data                                 'AD5611 Data übergeben
    Mode_sw_alt = Mode_sw                               'alten Wert zwischenspeichern

End Sub


Sub Displ_lcd_dac_data
    'Print Zeile 3 LCD-Display für DAC-Data

    Locate 3 , 1 : Lcd "                "               'Displayzeile löschen
    Locate 3 , 3 : Lcd "DAC ="
    Locate 3 , 11 : Lcd Zae1
    Select Case Mode_sw
      Case 2
        Locate 3 , 1 : Lcd "2"
      Case 3
        Locate 3 , 1 : Lcd "3"
      Case 4
        Locate 3 , 1 : Lcd "4"
    End Select

End Sub


'-----------------------------------------------------------------------
Sub Dac_tx()
    'In dieser Subroutine wird die Taste "Load" abgefragt. Ist diese Taste gedrückt,
    'dann wird der Zustand des DAC geändert und die neuen Daten an den DAC auf dem
    'PLL-Synthesizer-Board übertragen. Der DAC kann "aktiv" und "tristate" annehmen.
    'Das Datenformat: PD1 PD0 D9 D8 .... D1 D0 x x x x
    'PD1 wird zuerst übertragen, dann folgen die anderen Daten
    'PD1   PD0
    ' 0     0    =   Normal Operation
    ' 0     1    =   1k Ohm to GND
    ' 1     0    =   100k Ohm to GND
    ' 1     1    =   Three-State

    'PD1 = 32768
    'PD2 = 16384
    'PD1 + PD2 = 49152 = 1100 0000 0000 0000 B  / 16 = 3072

    'Input:     Tasten_1  1 bis 4
    '           Dac_val
    'Outup:     Daten werden an den DAC gesendet

    Pospar = 23 : Call P_debug                          '#23 Positionsparameter
   If Tasten_1 = 3 Then                                 'Taste "Load" gedrückt
     If Dac_aktiv = 0 Then
         Dac_aktiv = 1
         Dac_data = Dac_val * 16                        'Daten übernehmen + Shift 4 x left zur korrekten Positionierung
         Call Transmit_dac                              'Daten an den DAC übertragen
         Call Add_data                                  'Daten zur Steuerung übertragen
         Locate 4 , 1 : Lcd "*"                         'DAC-Status darstellen (DAC = aktiv)
     Else
         Dac_aktiv = 0
         Dac_data = 49152                               'DAC in Tristate schalten
         Call Transmit_dac                              'Daten an den DACübertragen
         Call Add_data                                  'Daten zur Steuerung übertragen
         Locate 4 , 1 : Lcd "&"                         'DAC-Status darstellen (DAC = tristate)
     End If
     Wait 1                                             '1 Sekunde warten
   End If

End Sub


'-----------------------------------------------------------------------
Sub Calc_ad9951()
    'In dieser Subroutine wird das Frequenz Tunig Word für den DDS-Baustein AD9951
    'berechnet und die Daten an die Subroutine "Set_AD9951" übergeben. Die Tabelle
    'zur Berechnung steht am Ende der Software. Das verlangt BASIC so.
    '
    'Input:     Frequenz     Frequenz
    'Output:                 DDS-Baustein programmiert

    Pospar = 24 : Call P_debug                          '#24 Positionsparameter
    'Berechnen des FTW (Frequenz Tunig Word)
    Frequenz_1 = Frequenz                               'Frequenz übernehmen
    Lu1 = Lookup(0 , Mult)                              'Bits pro Hz aus der Tabelle laden (erster Wert)
    F_tw = Lu1 * Frequenz_1                             'Teilberechnung Frequenz Tunig Word
    'Print
    For I = 1 To 8 Step 1
       Lu1 = Lookup(i , Mult)                           'Bits pro Hz aus der Tabelle laden
       Y1 = Lu1 * Frequenz_1                            'Arbeitsfrequenz * Bits pro Hz (Zwischenergebnis)
       Lud = Lookup(i , Divi)                           'Dekade laden
       Z1 = Y1 / Lud
       F_tw = F_tw + Z1                                 'Frequnz Tunig Word zusammensetzen
    Next I
    Call Set_ad9951                                     'Daten an den DDS-Baustein übertragen

End Sub
```

```
'-------------------------------------------------------------------------
Sub Set_ad9951
  'In dieser Subroutine werden dem DDS-Baustein der Inhalt des Control Function Register 2 und
  'das Frequenz Tuning (CFR4) übertragen.
  '
  'Input:     F_tw    Frequenz Tuning Word
  'Output:    Daten im DDS-Baustein

  Pospar = 25 : Call P_debug                              '#25 Positionsparameter
  'Init Datenübertragung zum DDS-Baustein
  Reset_dds = 0 : Ioupdate_dds = 0 : Sdio_dds = 0 : Sclk_dds = 0
  Tx_data = 0 : Tx_bit = 0

  'Daten an  das Control Function Register 1 (CFR1) übertragen
  Tx_data = &B00000000                                    'Adresse: Control Function Register 1 (CFR1) <0x00>
  Call Transmit_dds                                       'Daten übertragen
  'Print #2 , " " ;

  Tx_data = &B00000000                                    '<31:24> MSB High-Byte übertragen           <0x00>
  Call Transmit_dds
  'Print #2 , " " ;

  Tx_data = &B00000000                                    '<23:16> MSB Low-Byte übertragen            <0x00>
  Call Transmit_dds
  'Print #2 , " " ;

  Tx_data = &B01100010                                    '<15:8> LSB High-Byte übertragen            <0x62>
  Call Transmit_dds
  'Print #2 , " " ;

  Tx_data = &B01000000                                    '<7:0> LSB Low-Byte übertragen              <0x40>
  Call Transmit_dds
  'Print #2 , " "

  Ioupdate_dds = 1                                        'Datenübernahme zum DDS-Baustein = 1
  Waitus 20
  Ioupdate_dds = 0                                        'Datenübernahme zum DDS-Baustein = 0

  'Daten an  das Control Function Register 2 (CFR2) übertragen
  'Print #2 , ""
  Tx_data = &B00000001                                    'Adresse: Control Function Register 2 (CF2) <0x01>
  Call Transmit_dds                                       'Daten übertragen
  'Print #2 , " " ;
  Tx_data = &B00000000                                    '<23:16> MSB - Byte                         <0x00>
  Call Transmit_dds
  'Print #2 , " " ;
  Tx_data = &B00000000                                    '<15:8>                                     <0x00>
  Call Transmit_dds
  'Print #2 , " " ;
  Tx_data = &B00000110                                    '<7:0>  LSB                                 <0x06>
  Call Transmit_dds                                       'RefMult; VCO; Charge Pump für BAVARIX
  'Print #2 , " "

  Ioupdate_dds = 1                                        'Datenübernahme zum DDS-Baustein = 1
  Waitus 20
  Ioupdate_dds = 0                                        'Datenübernahme zum DDS-Baustein = 0

  'Daten an  das Control Function Register 4 (CFR4) übertragen
  'Print #2 , ""
  Tx_data = &B00000100                                    'Adresse: Control Function Register 4 (CFR4) <0x04>
  Call Transmit_dds                                       'Daten übertragen
  'Print #2 , " " ;

  Hw = Highw(f_tw)                                        'Frequenz Tuning Word übernehmen und in 2 Words zerlegen
  Lw = F_tw

  'Tx_data = &B00000110
  Tx_data = High(hw)                                      '<31:24> MSB High-Byte übertragen
  Call Transmit_dds
  'Print #2 , " " ;

  'Tx_data = &B10000001
  Tx_data = Low(hw)                                       '<23:16> MSB Low-Byte übertragen
  Call Transmit_dds
  'Print #2 , " " ;
    Tx_data = High(lw)                                    '<15:8> LSB High-Byte übertragen
    'Tx_data = &B00010101
    Call Transmit_dds
    'Print #2 , " " ;

    'Tx_data = &B11000100
    Tx_data = Low(lw)                                     '<7:0> LSB Low-Byte übertragen
    Call Transmit_dds
    'Print #2 , " "

    Ioupdate_dds = 1                                      'Datenübernahme zum DDS-Baustein = 1
    Waitus 20
    Ioupdate_dds = 0                                      'Datenübernahme zum DDS-Baustein = 0

End Sub


'-------------------------------------------------------------------------
Sub Transmit_dds()
  'In dieser Subroutine wird ein Byte an den DDS-Baustein übertragen.
  'Input:     Tx_data
  'Output:    Daten werden zu DDS-Baustein geschickt

  'Pospar = 26 : Call P_debug                               '#26 Positionsparameter <--- nicht anzeigen!
  For I = 7 To 0 Step -1                                  '8 Bit ausgeben, MSB zuerst
     Hz1 = 2 ^ I                                          'richtiges Bit zur Übertragung auswählen
     Tx_bit = Tx_data And Hz1
     If Tx_bit <> 0 Then
        'Print #2 , "1";                                   'Datenbit an der seriellen Schnittstelle ausgeben
       Sdio_dds = 1                                       'Datenbit setzen
     Else
       'Print #2 , "0";                                   'Datenbit an der seriellen Schnittstelle ausgeben
       Sdio_dds = 0                                       'Datenbit rücksetzen
     End If
     Waitus 20                                            'warten
     Sclk_dds = 1                                         'Clock generieren
```

```
      Waitus 20                                    'warten
      Sclk_dds = 0                                 'Clock zurücksetzen
   Next I

End Sub


'--------------------------------------------------------------------
Sub Transmit_dac()
  'In dieser Subroutine wird der DAC AD5611 programmiert. Es werden 16 Bits
  'übertragen, so wie sie dieser Subroutine übergeben werden.
  'Das Datenformat: PD1 PD0 D9 D8 .... D1 D0 x x x x
  'PD1 wird zuerst übertragen, dann folgen die anderen Daten
  'PD1   PD0
  ' 0     0    =   Normal Operation
  ' 0     1    =   1k Ohm to GND
  ' 1     0    =   100k Ohm to GND
  ' 1     1    =   Three-State

  'Input:     Dac_data
  'Output:    Daten werden zum DAC gesendet

  Pospar = 27 : Call P_debug                       '#27 Positionsparameter
  Tp4 = 1
  Sync_dac = 1                                     'init Sync
  Sclk_dac = 1                                     'init Clock

  Sync_dac = 0                                     'Start Datenübertragung
  For I = 15 To 0 Step -1                          '16 Bit ausgeben, MSB zuerst
     Hw1 = 2 ^ I                                   'richtiges Bit zur Übertragung auswählen
     Tx_word = Dac_data And Hw1
     If Tx_word <> 0 Then
       Sdin_dac = 1                                'Datenbit setzen
     Else
       Sdin_dac = 0                                'Datenbit rücksetzen
     End If
     Waitus 10                                     'warten
     Sclk_dac = 0                                  'Clock generieren
     Waitus 10                                     'warten
     Sclk_dac = 1                                  'Clock zurücksetzen
  Next I
  Sync_dac = 1                                     'Ende Datenübertragung
  Tp4 = 0
End Sub


'--------------------------------------------------------------------
Sub Transmit_hc595()
   'In dieser Subroutine wird das Schieberegister 74HC595 programmiert.
   'Es werden 16 Bits D0 ... D15 übertragen.
   'Das Datenformat: D15 D14 D13 D12 .... D1 D0.
   'Zuerst wird D0 übertragen.
   '
   'D15  =   N5         D7  =  IC9/9
   'D14  =   N4         D6  =  IC9/10
   'D13  =   N3         D5  =  IC12/9
   'D12  =   N2         D4  =  IC12/10
   'D11  =   N1         D3  =  not used
   'D10  =   N0         D2  =  not used
   'D9   =   S1         D1  =  not used
   'D8   =   S0         D0  =  not used
   'Datenübernahme mit der positiven Flanke Latch

   'Input:     Data_595
   'Output:    Daten werden zum 74HC596 gesendet

   Pospar = 28 : Call P_debug                      '#28 Positionsparameter
   Tp5 = 1
   Reset_src = 1                                   'init Reset
   Latch_clk = 0                                   'init Latch
   Shift_clk = 1                                   'init Clock
   'Latch_clk = 1                                   'Start Datenübertragung
   For I = 0 To 15 Step 1                          '16 Bit ausgeben, MSB zuerst
      Hz5 = I + 1
      Hw1 = 2 ^ I                                  'richtiges Bit zur Übertragung auswählen
      Tx_word = Data_595 And Hw1
      If Tx_word <> 0 Then
        Ser_data = 1                               'Datenbit setzen
        'Print #1 , "1";                           'Datenbit an der seriellen Schnittstelle ausgeben
        Mid(hstring , Hz5 , 1) = "1"
      Else
        Ser_data = 0                               'Datenbit rücksetzen
        'Print #1 , "0";                           'Datenbit an der seriellen Schnittstelle ausgeben
        Mid(hstring , Hz5 , 1) = "0"
      End If
      Waitus 10                                    'warten
      Shift_clk = 0                                'Clock generieren
      Waitus 10                                    'warten
      Shift_clk = 1                                'Clock zurücksetzen
   Next I
   Latch_clk = 1                                   'Ende Datenübertragung / Übernahme mit positiver Flanke
   Waitus 10                                       'warten
   Latch_clk = 0                                   'Latch wieder in Grundstellung
   Tp5 = 0
   Print #1 ,
   Call Prt_sh                                     'Daten an Schieberegister drucken
End Sub


'--------------------------------------------------------------------
Sub Prt_sh()
   'Diese Subroutine printet die Daten, welche an das Schieberegister
   'übergeben wurden und stellt als Kommentar die Funktion der Bits dar.
   'Zuerst wird D0 übertragen.
   '
   'D15  =   N5         D7  =  IC9/9
   'D14  =   N4         D6  =  IC9/10
   'D13  =   N3         D5  =  IC12/9
   'D12  =   N2         D4  =  IC12/10
   'D11  =   N1         D3  =  Trestate DAC
   'D10  =   N0         D2  =  not used
   'D9   =   S1         D1  =  not used
   'D8   =   S0         D0  =  not used
```

```
    'Input:  Hstring     Hilfsstring mit den Übertragungsdaten

    If Buglog <> 0 Then
       Print #1 ,
       Print #1 , "Schieberegister 74HC595"
       For Hz5 = 1 To 16
          Select Case Hz5
             Case 1 To 3
                Print #1 , Mid(hstring , Hz5 , 1) ;
                If Hz5 = 3 Then
                   Print #1 , "    not used"
                End If
             Case 4
                Print #1 , Mid(hstring , Hz5 , 1) ;
                Print #1 , "      DAC aktiv"
             Case 5 To 6
                Print #1 , Mid(hstring , Hz5 , 1) ;
                If Hz5 = 6 Then
                   Print #1 , "     IC12"
                End If
             Case 7 To 8
                Print #1 , Mid(hstring , Hz5 , 1) ;
                If Hz5 = 8 Then
                   Print #1 , "     IC9"
                End If
             Case 9 To 10
                Print #1 , Mid(hstring , Hz5 , 1) ;
                If Hz5 = 10 Then
                   Print #1 , "     S1, S0"
                End If
             Case 11 To 16
                Print #1 , Mid(hstring , Hz5 , 1) ;
                If Hz5 = 16 Then
                   Print #1 , " N5...N0"
                   Print #1 ,
                End If
          End Select
       Next
    End If

End Sub


'---------------------------------------------------------------------
Sub Drehgeber()
   'Diese Subroutine dient zur Übernahme der im Interrupt gemessenen Drehbewegung.
   'Die Änderung des Drehgebers seit dem letzten Durchlauf wird erfasst und
   'vorzeichenrichtig zum Zaehler addiert. Die Änderung des Drehgebers  wird dann
   'gelöscht.
   'Input:  Zaehler     noch nicht bearbeitete Zahlenänderung
   '        Tune        noch nich verrechnete Drehänderungen am Drehgeber

   Pospar = 29 : Call P_debug                              '#29 Positionsparameter
   Zaehler = Zaehler + Tune                                'letzte Drehbewegung übernehmen
   Tune = 0                                                'Drehbewegung löschen
End Sub


'---------------------------------------------------------------------
Sub Freq_aenderung()
   'In dieser Subroutine wird bestimmt, welche Stelle in der Frequenzanzeige geändert werden
   'soll. Die Variable "Zähler" wird entsprechend der angewählten Dekade multipliziert und die
   'Cursor-Position am Display wird parametriert.
   'Input:  Dek_Pos   angewählte Dekade
   '        Zaehler   Zahlenänderung
   'Output: Differenz   Frequenzänderung mit Vorzeichen
   '        X           Coursor-Position am LCD-Display
   '        Y           Zeile am Display
   '        Zaehler     wieder zurück gesetzt

   Pospar = 30 : Call P_debug                              '#30 Positionsparameter
   Select Case Dek_pos
      Case 0
         Differenz = Zaehler * 1                           'Änderung 1 Hz
         X = 12                                            'Position 11 am LCD-Display bestimmen
      Case 1
         Differenz = Zaehler * 10                          'Änderung 10 Hz
         X = 11                                            'Position 10 am LCD-Display bestimmen
      Case 2
         Differenz = Zaehler * 100                         'Änderung 100 Hz
         X = 10                                            'Position 9 am LCD-Display bestimmen
      Case 3
         Differenz = Zaehler * 1000                        'Änderung 1 kHz
         X = 8                                             'Position 7 am LCD-Display bestimmen
      Case 4
         Differenz = Zaehler * 10000                       'Änderung 10 kHz
         X = 7                                             'Position 6 am LCD-Display bestimmen
      Case 5
         Differenz = Zaehler * 100000                      'Änderung 100 kHz
         X = 6                                             'Position 5 am LCD-Display bestimmen
      Case 6
         Differenz = Zaehler * 1000000                     'Änderung 1 MHz
         X = 4                                             'Position 3 am LCD-Display bestimmen
      Case 7
         Differenz = Zaehler * 10000000                    'Änderung 10 MHz
         X = 3                                             'Position 2 am LCD-Display bestimmen
      Case Else
         'Error
   End Select
   Y = 1                                                   ' Zeile 1  auf dem LCD-Display bestimmen
   Zaehler = 0                                             ' Zähler bearbeitet und zurückgesetzt
End Sub


'---------------------------------------------------------------------
Sub Para_synthe()
   'In dieser Subroutine werden die frequenzbestimmenden Teile des Synthesizers programmiert. Es wird
   'bestimmt, ob alles übertragen wird, oder ob nur einzelne, neue Parametrierungen ausreichen. Auch
   'die Reihenfolge der Übertragung wird hier festgelegt.
   'Input:  Nennfrequenz
   'Output: Synthesizer wurde neu programmiert
   '@@@@
```

```
    eeee

    Pospar = 31 : Call P_debug                          '#31 Positionsparameter
    Y1 = Nennfrequenz / 1000000                         'Index berechnen
    Frq_bereich = Y1                                    'Index übernehmen
    Call F_bereich                                      'Berechnung Zae_ic9, Zae_hmc, Dac_val
    'DDS-Frequenz berechnen                              'F_DDS = (F_OUT * M - N_PLL2 * 13) / 2
    Y1 = Nennfrequenz * Mm                              'Zwischenergebnis F_OUT * M berechnen
    Z1 = Zae_hmc * 13000000                             'Zwischenergebnis N_PLL2 * 13 MHz berechnen
    Dds_frequenz = Y1 - Z1                              'Doppelte DDS-Frequenz berechnen
    Dds_frequenz = Dds_frequenz / 2                     'DDS-Frequenz berechnen
    Call Frequenz_variable                              'Frequenz-Variable darstellen
    Call Bereichs_variable                              'Bereichs-Variable darstellen
    Hz1 = Zae_hmc_alt1 - Zae_hmc                        'Differenz Zae_hmc alt und neu
    If Hz1 < 0 Then                                     'negative Differenz umdrehen
       Hz1 = Hz1 * -1
    End If
    If Frq_bereich = Frq_bereich_alt Then               'ganz kleine Frequenzänderung, nur DDS aktiv
       Frequenz = Dds_frequenz                          'DDS-Frequenz übergeben
       Call Calc_ad9951                                 'DDS-Frequenz berechnen und an DDS übergeben
       Call Frequenzanzeige                             'Frequenz am Display anzeigen
       Print #1 , "Nur DDS-Frequenz geändert" ; : : Print #1 ,
    Else
        Frq_bereich_alt = Frq_bereich
        If Hz1 < 2 Then                                 'kleine Frequenzänderung, Schalter + DAC nicht aktiv
          'neue Parameter übertragen (#20 + #21 + #28)
          '13 MHz Raster und N_PLL2 Teiler setzen
          Print #1 , "Nur DDS-Frequenz + Teiler geändert" ; : Print #1 ,
          Zae_ic12 = 1                                  'IC12 setzen (13 MHz-Raster)
          Call Calc_hmc698                              'Zae_s berechnet und Parameter übertragen
          'DDS-Frequenz übertragen (#24 + #25 + #38)
          Frequenz = Dds_frequenz                       'DDS-Frequenz übergeben
          Call Calc_ad9951                              'DDS-Frequenz berechnen und an DDS übergeben
          Call Frequenzanzeige                          'Frequenz am Display anzeigen
       Else                                             'große Frequenzänderung, Schalter + DAC aktiv
          Print #1 , "DDS-Frequenz + Teiler geändert + Schalter aktiv" ; : Print #1 ,
          'DAC laden + übertragen (#27)
          Dac_data = Dac_val * 16                       'Daten übernehmen + Shift 4 x left zur korrekten Positionierung
          Call Transmit_dac                             'Daten an den DAC übertragen
          'Schalter ein mit alten Parametern übertragen (#21 + #28)
          Dac_aktiv = 1                                 'Schalter ein
          Call Add_data                                 'Schalter-Bit ins Schieberegister eintragen und senden
          'neue Parameter übertragen (#20 + #21 + #28)
          '13 MHz Raster und N_PLL2 Teiler setzen
          Zae_ic12 = 1                                  'IC12 setzen (13 MHz-Raster)
          Call Calc_hmc698                              'Zae_s berechnet und Parameter übertragen
          'DDS-Frequenz übertragen (#24 + #25 + #38)
          Frequenz = Dds_frequenz                       'DDS-Frequenz übergeben
          Call Calc_ad9951                              'DDS-Frequenz berechnen und an DDS übergeben
          Call Frequenzanzeige                          'Frequenz am Display anzeigen
          'Schalter wieder öffnen (#21 + #28)
          Dac_aktiv = 0                                 'Schalter aus
          Call Add_data                                 'Schalter-Bit ins Schieberegister eintragen und senden
        End If
    End If
    Zae_hmc_alt1 = Zae_hmc                              'letzten Wert von N_PLL2 zwischenspeichern
End Sub


'---------------------------------------------------------------------
Sub N_frequenz()
    'In dieser Subroutine wird die Nennfrequenz berechnet. Die Nennfrequenz ist die Frequenz,
    'welche man einstellt. Also die Empfangsfrequenz ohne Berücksichtigung von Korrekturfaktoren
    'und ZF.
    'Input:  Differenz
    '        Nennfrequenz (alt)
    'Output: Nennfrequenz (neu)

    Pospar = 32 : Call P_debug                          '#32 Positionsparameter
    Nennfrequenz = Nennfrequenz + Differenz             'Nennfrequenz berechnen
    If Nennfrequenz > 74999999 Then
       Nennfrequenz = 74999999                          'maximale Nennfrequenz
    End If

    If Nennfrequenz < 45030000 Then
       Nennfrequenz = 45030000                          'minimale Nennfrequenz
    End If
    Differenz = 0                                       'Differenz auf 0 zurücksetzen
End Sub


'---------------------------------------------------------------------
Sub Bereichs_variable()
    'Bereichs-Variable darstellen, wenn das Log-Flag 1 ist.

    Pospar = 34 : Call P_debug                          '#34 Positionsparameter
    If Buglog = 1 Then
       Print #2 ,
       Print #2 , "Zae_ic9         = " ; : Print #2 , "Zae_ic9      = " ; : Print #2 , Zae_ic9
       Print #2 , "M               = " ; : Print #2 , "Mm           = " ; : Print #2 , Mm
       Print #2 , "N_PLL2          = " ; : Print #2 , "Zae_hmc      = " ; : Print #2 , Zae_hmc
       Print #2 , "DA-Wert         = " ; : Print #2 , "Dac_val      = " ; : Print #2 , Dac_val
       Print #2 , "Frequenzbereich = " ; : Print #2 , "Frq_bereich  = " ; : Print #2 , Frq_bereich
    End If

End Sub


'---------------------------------------------------------------------
Sub Frequenz_variable()
    'Frequenz-Variable darstellen, wenn das Log-Flag 1 ist.

    Pospar = 35 : Call P_debug                          '#35 Positionsparameter
    If Buglog = 1 Then
       Print #2 ,
       Print #2 , "Local Osc.      = " ; : Print #2 , "Nennfrequenz         = " ; : Print #2 , Nennfrequenz
       Print #2 , "F_OUT * M       = " ; : Print #2 , "Y1                   = " ; : Print #2 , Y1
       Print #2 , "N_PLL2 * 13MHz  = " ; : Print #2 , "Z1                   = " ; : Print #2 , Z1
       Print #2 , "DDS-Frequenz    = " ; : Print #2 , "Dds-frequenz         = " ; : Print #2 , Dds_frequenz
    End If

End Sub
```

```
'---------------------------------------------------------------------
Sub Single_step()
   'Diese Subroutine wartet, bis irgend eine Taste gedrückt wird. Bis dahin wird diese
   'Subroutine nicht verlassen.

   If Ss = 1 Then                                          'testen, ob Single Step gewünscht
      Print #2 , "warten auf Taste"                        'auf Tastendruck warten
      Locate Y , X : Lcd ""                                'Cursor wieder rücksetzen
      Remote = 0                                           'Remote initialisieren
      While Remote = 0                                     'Solange, bis eine Taste gedrückt
           Remote = Inkey(#2)                              'Serielle Schnittstelle vom PC einlesen
      Wend
   End If

End Sub


'---------------------------------------------------------------------
Sub Rd_pll()
   'Diese Subroutine liest den Analogwert der PLL-Regelschleife aus und zeigt ihn an.

   If Ss = 1 Then                                          'testen, ob Single Step gewünscht
      Print #2 ,
      Print #2 , "--- PLL-Regelschleife ---"
      Print #2 , " Analogwert:           DAC:   "
      Adc_val_alt = Adc_val - 1                            'Routine initialisieren
      For I = 3 To 0 Step -1
         Channel = 7                                       'ADC 7 Analogspannung PLL-Regelschleife einlesen
         Adc_val = Getadc(channel)                         'ADC initialisiseren
         Adc_val = Getadc(channel)                         'ADC starten und Analogwert digitalisieren
         Adc_mvolt = 48 * Adc_val
         Adc_mvolt = Adc_mvolt \ 10
         Print #2 , Adc_mvolt ; : Print #2 , " mV = " ;    'ADC-Value in mV
         Print #2 , Adc_val ; : Print #2 , "    ---    " ; 'ADC-Value als 12 Bit-Wert ausgeben
         Print #2 , " " ; : Print #2 , Dac_val             'DAC-Value ausgeben
         Adc_val_alt = Adc_val                             'alten Wert übergeben
         Waitms 500                                        '0,5 Sek warten
      Next I
   End If

End Sub
'@@@@


'---------------------------------------------------------------------
Sub F_bereich()
   'In dieser Subroutine wird der 1 MHz-Frequenz_Bereich berechnet. In Abhängigkeit
   'dieses Bereiches werden alle benötigten Variablen für die Einstellung des
   'Synthesizers gesetzt.
   'Input:  Frq_bereich
   'Output: Zae_ic9
   '        Zae_hmc
   '        Dac_val
   '        Mm

   Pospar = 37 : Call P_debug                              '#37 Positionsparameter
   Select Case Frq_bereich
      Case 45                                              '45 Mhz-Bereich
         Zae_ic9 = 3
         Zae_hmc = 47
         Dac_val = 550
      Case 46                                              '46 MHz-Bereich
         Zae_ic9 = 3
         Zae_hmc = 48
         Dac_val = 600
      Case 47                                              '47 MHz-Bereich
         Zae_ic9 = 3
         Zae_hmc = 49
         Dac_val = 640
      Case 48                                              '48 MHz-Bereich
         Zae_ic9 = 3
         Zae_hmc = 50
         Dac_val = 700
      Case 49                                              '49 MHz-Bereich
         Zae_ic9 = 2
         Zae_hmc = 44
         Dac_val = 410
      Case 50                                              '50 MHz-Bereich
         Zae_ic9 = 2
         Zae_hmc = 44
         Dac_val = 450
      Case 51                                              '51 MHz-Bereich
         Zae_ic9 = 2
         Zae_hmc = 45
         Dac_val = 490
      Case 52                                              '52 MHz-Bereich
         Zae_ic9 = 2
         Zae_hmc = 46
         Dac_val = 520
      Case 53                                              '53 Mhz-Bereich
         Zae_ic9 = 2
         Zae_hmc = 47
         Dac_val = 570
      Case 54                                              '54 MHz-Bereich
         Zae_ic9 = 2
         Zae_hmc = 48
         Dac_val = 610
      Case 55                                              '55 MHz-Bereich
         Zae_ic9 = 2
         Zae_hmc = 49
         Dac_val = 660
      Case 56                                              '56 MHz-Bereich
         Zae_ic9 = 2
         Zae_hmc = 50
         Dac_val = 700
      Case 57                                              '57 MHz-Bereich
         Zae_ic9 = 2
         Zae_hmc = 51
         Dac_val = 740
      Case 58                                              '58 MHz-Bereich
         Zae_ic9 = 1
         Zae_hmc = 43
```

```
            Zae_hmc = 45
            Dac_val = 380
        Case 59                                 '59 MHz-Bereich
            Zae_ic9 = 1
            Zae_hmc = 44
            Dac_val = 410
        Case 60                                 '60 MHz-Bereich
            Zae_ic9 = 1
            Zae_hmc = 45
            Dac_val = 440
        Case 61                                 '61 MHz-Bereich
            Zae_ic9 = 1
            Zae_hmc = 45
            Dac_val = 480
        Case 62                                 '62 MHz-Bereich
            Zae_ic9 = 1
            Zae_hmc = 46
            Dac_val = 510
        Case 63                                 '63 MHz-Bereich
            Zae_ic9 = 1
            Zae_hmc = 47
            Dac_val = 550
        Case 64                                 '64 MHz-Bereich
            Zae_ic9 = 1
            Zae_hmc = 48
            Dac_val = 580
        Case 65                                 '65 MHz-Bereich
            Zae_ic9 = 1
            Zae_hmc = 48
            Dac_val = 620
        Case 66                                 '66 MHz-Bereich
            Zae_ic9 = 1
            Zae_hmc = 49
            Dac_val = 660
        Case 67                                 '67 MHz-Bereich
            Zae_ic9 = 1
            Zae_hmc = 50
            Dac_val = 690
        Case 68                                 '68 Mhz-Bereich
            Zae_ic9 = 0
            Zae_hmc = 40
            Dac_val = 240
        Case 69                                 '69 MHz-Bereich
            Zae_ic9 = 0
            Zae_hmc = 41
            Dac_val = 280
        Case 70                                 '70 MHz-Bereich
            Zae_ic9 = 0
            Zae_hmc = 41
            Dac_val = 310
        Case 71                                 '71 MHz-Bereich
            Zae_ic9 = 0
            Zae_hmc = 42
            Dac_val = 340
        Case 72                                 '72 MHz-Bereich
            Zae_ic9 = 0
            Zae_hmc = 43
            Dac_val = 360
        Case 73                                 '73 MHz-Bereich
            Zae_ic9 = 0
            Zae_hmc = 43
            Dac_val = 390
        Case 74                                 '74 MHz-Bereich
            Zae_ic9 = 0
            Zae_hmc = 44
            Dac_val = 420
        Case Else                               'Error
            'Error
    End Select

    Select Case Zae_ic9                         'Berechnung Variable M
        Case 0
            Mm = 8
        Case 1
            Mm = 10
        Case 2
            Mm = 12
        Case 3
            Mm = 14
    End Select

End Sub


'--------------------------------------------------------------------
Sub Frequenzanzeige()
'Diese Subroutine "Positioniere Frequenz Anzeige" wandelt die Variable
'"Frequenz" und setzt sie mit Abstand zwischen Hz, kHz und MHz wieder zusammen.
'Input:  Nennfrequenz
'Output: Anzeige am LCD-Display

Local A As Byte
Local Laenge As Byte
Local Posi As Byte
Local Ziffer As String * 1

    Pospar = 38 : Call P_debug                  '#38 Positionsparameter
    Freqstr = Str(nennfrequenz)
    Laenge = Len(freqstr)

    Locate 1 , 1 : Lcd "                "       'Zeile 1 löschen
    Posi = Laenge - 0                           'Komma-Stelle     -- --- ---.x
    Ziffer = Mid(freqstr , Posi , 1)
    Locate 1 , 14 : Lcd "Hz"
    Locate 1 , 12 : Lcd Ziffer
    Locate 1 , 9 : Lcd "."                      'Linken Punkt unterdrücken, wenn Anzeige < 1.000.000
    If Nennfrequenz < 1000000 Then
        'nichts tun
    Else
        Locate 1 , 5 : Lcd "."
    End If

    Posi = Laenge - 1                           'Hz-Anzeige       -- --- --x.x
    Ziffer = Mid(freqstr , Posi , 1)
```

```
    Locate 1 , 11 : Lcd Ziffer

    Posi = Laenge - 2                       'Hz * 10-Stelle   -- --- -xx.x
    Ziffer = Mid(freqstr , Posi , 1)
    Locate 1 , 10 : Lcd Ziffer

    Posi = Laenge - 3                       'Hz * 100-Anzeige -- --- xxx.x
    Ziffer = Mid(freqstr , Posi , 1)
    Locate 1 , 8 : Lcd Ziffer

    Posi = Laenge - 4                       'kHz-Stelle       -- --x xxx.x
    Ziffer = Mid(freqstr , Posi , 1)
    Locate 1 , 7 : Lcd Ziffer

    Posi = Laenge - 5                       'kHz * 10-Stelle  -- -xx xxx.x
    Ziffer = Mid(freqstr , Posi , 1)
    Locate 1 , 6 : Lcd Ziffer

    Posi = Laenge - 6                       'kHz * 100-Stelle -- xxx xxx.x
    Ziffer = Mid(freqstr , Posi , 1)
    Locate 1 , 4 : Lcd Ziffer

    Posi = Laenge - 7                       'MHz -Stelle      -x xxx xxx.x
    Ziffer = Mid(freqstr , Posi , 1)
    Locate 1 , 3 : Lcd Ziffer

    Posi = Laenge - 8                       'MHz * 10-Stelle  -x xxx xxx.x
    Ziffer = Mid(freqstr , Posi , 1)
    Locate 1 , 2 : Lcd Ziffer

    Posi = Laenge - 9                       'MHz * 100-Stelle xx xxx xxx.x
    Ziffer = Mid(freqstr , Posi , 1)
    Locate 1 , 1 : Lcd Ziffer

End Sub

'-----------------------------------------------------------------------
'Daten zur Berechnung des Frequenz Tuning Words. Diese werden in der Subroutine "Calc_ad9951"
'benötigt

Mult:
      Data 10 , 3 , 2 , 4 , 4 , 4 , 0 , 6 , 2
Divi:
      Data 1& , 10& , 100& , 1000& , 10000& , 100000 , 1000000 , 10000000 , 100000000
```