

```

1 '<STEUERRECHNER_V0.05.BAS>
2 '
3 ' *****
4 ' * *
5 ' * Programm für Testaufbau Steuerrechner *
6 ' * ----- *
7 ' * *
8 ' * Version 0.05 *
9 ' * *
10 ' * 25.10.2009 W. Nitsche *
11 ' * *
12 ' * geschrieben in BASCOM-AVR *
13 ' * *
14 ' *****
15 '
16 'Achtung: Für dieses Programm muss der Stack folgendermassen initialisiert sein:
17 ' HW Stack 64
18 ' Soft Stack 64 (für lokale Variable in Subroutinen)
19 ' Frame Size 64 (für lokale Variable in Subroutinen)
20 '
21 '$sim
22 '$sim ist zum Simulieren da und muß normal auskommentiert sein @@@
23 '
24 'Include Files für ATMegal28 einbinden -----
25 $regfile = "m128def.dat"
26
27 'Initialisierung CPU -----
28 $hwstack = 64
29 $framesize = 64
30 $swstack = 64
31
32 'Zuweisungen -----
33 Tp1 Alias Porte.2 'Zuweisung TP1 = PORTE.2
34 Tp2 Alias Porte.3 'Zuweisung TP2 = PORTE.3
35 Tp3 Alias Porte.4 'Zuweisung TP3 = PORTE.4
36 Tp4 Alias Porte.5 'Zuweisung TP4 = PORTE.5
37 Tp5 Alias Porte.6 'Zuweisung TP5 = PORTE.6
38 Tp6 Alias Porte.7 'Zuweisung TP6 = PORTE.7
39
40 'DDS-Baustein
41 Sdio_dds Alias Porta.0 'Zuweisung SDIO DDS-Baustein = PORTA.0
42 Sclk_dds Alias Porta.1 'Zuweisung Clock DDS-Baustein = PORTA.1
43 Iouupdate_dds Alias Porta.2 'Zuweisung IOUPDATE DDS-Baustein = PORTA.2
44 Reset_dds Alias Porta.3 'Zuweisung Reset DDS-Baustein = PORTA.3
45 Sdo_dds Alias Pina.4 'Zuweisung SDO DDS-Baustein = PORTA.4
46
47
48 'Initialisierung Serielle Schnittstelle -----
49 Open "comd.3:9600,8,n,1" For Output As #2 'Channel 2 für Kommunikation mit PC öffnen
50 Print #2 , "init"
51 $crystal = 16000000 'Quarz 16,000MHz
52 $baud = 153600 '9600Baud
53
54 'Initialisierung I2C-Bus -----
55 Config Sda = Portd.5
56 Config Scl = Portd.4
57 I2cinit
58
59 'Initialisierung LC-Display im 4 Bit-Pin-Mode und 16 *2 Character -----
60 Config Lcd = 16 * 2
61 'Die Complier-Einstellungen werde mit "Config Lcdpin" überschrieben
62 Config Lcdpin = Pin , Db4 = Portc.4 , Db5 = Portc.5 , Db6 = Portc.6 , Db7 = Portc.7 , E = Portc.3 , Rs = Portc.1
63
64 'Initialisierung AD-Converter -----
65 'Config Adc = Single , Prescaler = Auto 'Reference = Internal
66 'Reference = Internal darf nicht
67 'eingeschalten werden, wenn eine
68 'externe Spannung angeschlossen ist
69
70 'Initialisiere External RAM
71 Config Xram = Disabled , Waitstatels = 1 , Waitstatehs = 2 'Externen RAM-Zugriff ausschalten
72
73
74 'Initialisierung Port-Konfiguration -----
75 Ddra.0 = 1 'Config SDIO_DDS = Output
76 Ddra.1 = 1 'Config SCLK_DDS = Output
77 Ddra.2 = 1 'Config IOUPDATE_DDS = Output
78 Ddra.3 = 1 'Config RESET = Output
79 Ddra.4 = 0 'Config SDO_DDS = Input
80 Ddra.5 = 0 'Config Pina.5 = Input
81 Ddra.6 = 0 'Config Pina.6 = Input
82 Ddra.7 = 0 'Config Pina.7 = Input

```

```

83
84 Ddrb.0 = 1 'Config Pinb.0 = Output
85 Ddrb.1 = 1 'Config Pinb.1 = Output ISP-Programmierung
86 Ddrb.2 = 1 'Config Pinb.2 = Output
87 Ddrb.3 = 1 'Config Pinb.3 = Output
88 Ddrb.4 = 0 'Config Pinb.4 = Input
89 Ddrb.5 = 1 'Config Pinb.5 = Output
90 Ddrb.6 = 0 'Config Pinb.6 = Input
91 Ddrb.7 = 0 'Config Pinb.7 = Input
92
93 Ddrc.0 = 1 'Config Pinc.0 = Output
94 Ddrc.2 = 1 'Config Pinc.2 = Output
95
96 Ddrd.0 = 0 'Config Pind.0 = Input
97 Ddrd.1 = 0 'Config Pinc.1 = Input
98 Ddrd.6 = 0 'Config Pinb.6 = Input Drehschalter DR0
99 Ddrd.7 = 0 'Config Pinb.7 = Input Drehschalter DR1
100
101 'Initialisierung der Ports -----
102 Dim Porta.0 As Bit 'SDIO_DDS = Output
103 Dim Porta.1 As Bit 'SCLK_DDS = Output
104 Dim Porta.2 As Bit 'IOUPDATE_DDS = Output
105 Dim Porta.3 As Bit 'RESET = Output
106 Dim Porta.4 As Bit 'SDO_DDS = Input
107 Dim Porta.5 As Bit 'Reserve
108 Dim Porta.6 As Bit 'Reserve
109 Dim Porta.7 As Bit 'Reserve
110
111 Dim Portb.0 As Bit 'PB.0 = Output
112 Dim Portb.1 As Bit 'PB.1 = Output
113 Dim Portb.2 As Bit 'PB.2 = Output
114 Dim Portb.3 As Bit 'PB.3 = Output
115 Dim Portb.4 As Bit 'PB.4 = Input
116 Dim Portb.5 As Bit 'PB.5 = Output
117 Dim Portb.6 As Bit 'PB.6 = Input
118 Dim Portb.7 As Bit 'PB.7 = Input
119
120 Dim Portc.0 As Bit 'LCD-Display Beleuchtung
121 Dim Portc.2 As Bit 'LCD-Display R/W
122
123 Dim Portd.0 As Bit 'INT0 Tasten einlesen
124 Dim Portd.1 As Bit 'INT1 Drehschalter
125 Dim Portd.6 As Bit 'PD.6 = Drehschalter DR0
126 Dim Portd.7 As Bit 'PD.7 = Drehschalter DR1
127
128 'Initialisierung Testpunkte -----
129 Ddre.2 = 1 'Config Pine.2 = Output TP1
130 Ddre.3 = 1 'Config Pine.3 = Output TP2
131 Ddre.4 = 1 'Config Pine.4 = Output TP3
132 Ddre.5 = 1 'Config Pine.5 = Output TP4
133 Ddre.6 = 1 'Config Pine.6 = Output TP5
134 Ddre.7 = 1 'Config Pine.7 = Output TP6
135 Dim Porte.2 As Bit 'TP1
136 Dim Porte.3 As Bit 'TP2
137 Dim Porte.4 As Bit 'TP3
138 Dim Porte.5 As Bit 'TP4
139 Dim Porte.6 As Bit 'TP5
140 Dim Porte.7 As Bit 'TP6
141
142 'Initialisierung globaler Variabler -----
143 Dim Hz1 As Integer 'Hilfsvariable 1
144 Dim Hz2 As Integer 'Hilfsvariable 2
145 Dim Var1 As Word 'Hilfsvariable für Printroutine
146 Dim Posp As Byte 'Hilfsvariable für Printroutine
147
148 Dim Mode_sw As Byte 'ADC-Value für Mode-Schalter
149 Dim Tasten_1 As Byte 'ADC-Value für Tasten_1
150 Dim Tasten_2 As Byte 'ADC-Value für Tasten_2
151 Dim Tastenfeld As Byte 'ADC-Value für Tastenfeld
152 Dim Tastatur As Byte 'Tastatenfeld aktiv
153 Dim Tast_value_alt1 As Byte 'Tastatur-Eingabe zwischenspeichern
154 Dim Tast_value As Byte 'Tastatur-Eingabe
155 Dim Tast_value_alt As Byte 'alte Tastatur-Eingabe
156
157 Dim Tune As Integer 'hier werden die Drehungen am Tune-Knopf gezählt
158 Dim Zaehler As Integer 'Zähler für Drehgeber
159 Dim Dek_pos As Byte 'Zähler für Dekaden-Position
160 Dim X As Byte 'Horizontal-Position auf dem LCD-Display
161 Dim Y As Byte 'Vertikal-Position auf dem LCD-Display
162 Dim Z As Byte , Slave As Byte 'Speicherzelle für I2C-Bus read
163 Dim Differenz As Long 'gewünschte Frequenzänderung
164 Dim Nennfrequenz As Long 'Nennfrequenz ohne Berücksichtigung der ZF usw.

```

```

165 Dim Nennfrequenz_alt As Long 'Nennfrequenz alt zum Vergleich mit Nennfrequenz
166 Dim Freqstr As String * 12 'Frequenzanzeige als String
167 Dim Adc_val As Word , Channel As Byte 'Variablenübergabe A/D-Wandler
168 Dim Gd_val As Byte 'Gain und Decreasing kombiniert
169 Dim Gd_val_old As Byte 'Zwischenspeicher für Gd_val
170 Dim Tx_data As Byte 'Datenbyte zur Übertragung an den DDS
171 Dim Tx_bit As Byte 'Datenbyte zur Maskierung des zu übertragenden Bits
172
173 'Variable zur Berechnung des 32-Bit Tuning Word für den DDS -----
174 Dim Frequenz As Long 'Frequenzübergabe zum Programmieren des DDS-Bausteins
175 Dim F_tw As Long 'Frequenz Tunig Word
176 Dim Frequenz_1 As Long 'DDS-Frequenz
177 Dim I As Byte 'Laufvariable
178 Dim Lul As Byte 'Variable aus Lookup-Tabelle Mult:
179 Dim Lud As Long 'Hilfsvariable
180 Dim Y1 As Long 'Hilfsvariable
181 Dim Z1 As Long 'Hilfsvariable
182 Dim Hw As Word 'Hilfsvariable
183 Dim Lw As Word 'Hilfsvariable
184 Dim Ftw(5) As Byte 'Array zur Übertragung der Daten zum DDS-Baustein
185
186 'Initialisierung von I2C-Bus-Adressen -----
187 'Const Pcf8574_exp = &H44 'PCF8574 I2C-Bus Expander <===== Test mit I2C-Bus-Expander
188 Const Pcf8574_ic8 = &H40 'PCF8574 Bandfilter Preselector IC8
189 'LED0 = BF_A = 50kHz bis 500kHz
190 'LED1 = BF_B = 500kHz bis 1,8MHz
191 'LED2 = BF_C = 1,5MHz bis 2,47MHz
192 'LED3 = BF_D = 2,47MHz bis 4,07MHz
193 'LED4 = BF_E = 4,07MHz bis 6,7MHz
194 'LED5 = BF_F = 6,7MHz bis 11,05 MHz
195 'LED6 = BF_G =11,05MHz bis 18,2MHz
196 'LED7 = BF_H = 18,2MHz bis 30MHz
197
198 Const Pcf8574_ic13 = &H42 'PCF8574 Gain Preselector IC13
199 'LED0 = Gain Eingangsdämpfungsglied
200 'LED1 = Gain G1 Verstärker
201 'LED2 = Gain G2 Verstärker
202 'LED3 = Gain G3 Verstärker
203 'LED4 = Gain G4 Verstärker
204 'LED5 = NC_1
205 'LED6 = NC_2
206 'LED7 = NC_3
207
208 'Initialisierung Procedures und Functions -----
209 Declare Sub Drehgeber 'Der Drehgeber wird eingelesen
210 Declare Sub Freq_aenderung 'Die Frequenzänderung wird berechnet
211 Declare Sub N_frequenz 'In dieser Subroutine wird die Nennfrequenz berechnet
212 Declare Sub Frequenzanzeige 'Frequenzanzeige am LCD-Display
213
214 Declare Function M_value(byval Adc_val As Word) As Byte 'In dieser Function wird der Mode-Wert bestimmt
215 Declare Function T1_value(byval Adc_val As Word) As Byte 'In dieser Function wird der Tastencode der T1-Tastengruppe bestimmt
216 Declare Function T2_value(byval Adc_val As Word) As Byte 'In dieser Function wird der Tastencode der T2-Tastengruppe bestimmt
217 Declare Function Tf_value(byval Adc_val As Word) As Byte 'In dieser Function wird der Tastencode der Tastatur bestimmt
218
219 Declare Sub Calc_ad9951 'DDS-Baustein Frequenz berechnen
220 Declare Sub Set_ad9951 'DDS-Baustein AD9951 programmieren
221 Declare Sub Transmit_dds 'In dieser Subroutine wird ein Byte an den DDS-Baustein übertragen
222 Declare Sub Tastenfeldeingabe 'Tastenfeldeingabe
223
224 Declare Sub Print_sm 'Testroutine zum Ausdrucken von Parametern über die serielle
Schnittstelle
225
226
227 'Interrupts initialisieren -----
228 On Int1 Ondr0 'Interrupt-Routine für Drehschalter
229 Config Int1 = Falling 'Interrupt 1 bei H/L-Flanke auslösen
230 Enable Int1 'Externen Interrupt 1 einschalten
231
232 Enable Interrupts 'Interrupts global zulassen
233
234
235 Goto Hauptprogramm
236
237 'Interrupts -----
238 Ondr0: 'Int1-Routine überwacht den Drehschalter "Tune"
239 If Tp2 = 0 Then Tp2 = 1 Else Tp2 = 0 'Toggle TP2 für <=====
Testzwecke
240 If Pind.6 = Pind.7 Then 'Rechtsdrehung
241 Tune = Tune + 1
242 Else 'Linksdrehung
243 Tune = Tune - 1
244 End If

```

```

245 Return
246
247
248 'Hauptprogramm AVX-EVAL-Board ----- @@@
249 Hauptprogramm:
250
251 Print #2 , "Testaufbau Steuerrechner Version 0.05 --- Inbetriebnahme DDS-Baustein"
252 Print #2 ,
253
254 'Init Preselector
255 Gd_val = 3 'Gain = 1 Dämpfung = ein
256 Gd_val = Gd_val Xor 255 'negative Logik
257 I2cstart 'generiere Start
258 I2cwbyte Pcf8574_ic13 'send slave Adresse
259 I2cwbyte Gd_val
260 I2cstop
261
262 'Beleuchtung LCD-Display ein
263 Portc.0 = 1
264 'LCD-Display R/W auf write stellen
265 Portc.2 = 0
266
267 'AD-Wandler starten
268 Start Adc
269
270 'Variable setzen
271 Nennfrequenz = 2500000 'Nennfrequenz nach Start
272 Nennfrequenz_alt = 0 'Nennfrequenz alt rücksetzen
273
274 'Tastatureingabe aus
275 Tastatur = 0
276
277 'InitLcd durchführen.
278 Initlcd 'LCD-Reset per Instruktion (HD4478-PowerOn-Reset reicht oft nicht)
279 Cls 'Anzeige löschen
280 Cursor On 'LCD-Cursor einschalten
281 Locate 4 , 1 : Lcd " Steuerrechner " 'Begrüßung auf dem LCD-Display
282
283 Wait 2 '2 Sekunden warten
284
285 Do
286 If Tpl = 0 Then Tpl = 1 Else Tpl = 0 'Toggle TP1 für Testzwecke
287
288 'AD-Wandler
289 Channel = 0 'ADC für Mode auslesen
290 Adc_val = Getadc(channel) 'ADC starten und Analogwert digitalisieren
291 Mode_sw = M_value(adc_val) 'ADC-Value in Mode-Wert
292 umwandeln 'Variable rücksetzen
293
294 'AD-Wandler
295 Channel = 1 'ADC für Tasten_1 auslesen
296 Adc_val = Getadc(channel) 'ADC starten und Analogwert digitalisieren
297 Tasten_1 = T1_value(adc_val) 'Tastencode von Tastengruppe 1
298 übernehmen 'Variable rücksetzen
299
300 'AD-Wandler
301 Channel = 2 'ADC für Tasten_2 auslesen
302 Adc_val = Getadc(channel) 'ADC starten und Analogwert digitalisieren
303 Tasten_2 = T2_value(adc_val) 'Tastencode von Tastengruppe 2
304 übernehmen 'Variable rücksetzen
305
306 'AD-Wandler
307 Channel = 3 'ADC für Tastenfeld auslesen
308 Adc_val = Getadc(channel) 'ADC starten und Analogwert digitalisieren
309 Var1 = Adc_val '<--- Test
310 Tastenfeld = Tf_value(adc_val) 'Tastencode übernehmen
311 Call Print_sm '<--- Test
312
313 'Tastefeldeingabe
314 Call Tastefeldeingabe 'Tastefeld einlesen
315
316 Tast_value_alt1 = Tast_value 'letzte Variable zwischenspeichern
317
318 Cursor On 'LCD-Cursor einschalten
319 'Drehgeber auslesen und Änderung aktualisieren 'Frequenzeingabe via Dehgeber
320 Call Drehgeber 'Drehgeber eingelesen
321 Call Freq_aenderung 'Frequenzänderung berechnet
322 Call N_frequenz 'Nennfrequenz berechnen
323
324 Locate Y , X : Lcd ""
325 If Nennfrequenz <> Nennfrequenz_alt Then 'hat sich die Frequenz geändert?
326 Frequenz = Nennfrequenz 'Frequenzübergabe

```

```

324      Call Calc_ad9951                                'DDS-Baustein programmieren
325      Call Frequenzanzeige                            'Frequenz am LCD-Display anzeigen
326      Nennfrequenz_alt = Nennfrequenz                'letzte Frequenz am Display merken
327      End If
328
329  Loop
330
331
332  '-----
333  Sub Tastenfeldeingabe
334
335      If Tast_value_alt <> Tast_value Then            'Test, ob Tastenwert geändert
336          If Tast_value = 11 Then                    'Tastatureingabe gewünscht
337              Tastatur = 0                            'Rücksetzen der alten Variablen
338              Nennfrequenz = 0
339          End If
340          If Tast_value = 12 Then                      'Tastatur-Eingabe beendet
341              Tastatur = 2
342          End If
343      End If
344      Tast_value_alt = Tast_value                      'letzte Variable zwischenspeichern
345
346      If Tast_value_alt1 <> Tast_value Then            'Test, ob Tastenwert geändert
347          Select Case Tastatur
348              Case 0                                  'Tastefeld nicht aktiv
349                  Tastatur = 1
350              Case 1                                  'Tastatur-Eingabe aktiv
351                  If Tast_value < 10 Then
352                      Nennfrequenz = Nennfrequenz * 10
353                      Nennfrequenz = Nennfrequenz + Tast_value
354                  End If
355              Case 2                                  'Tastefeld-Eingabe beenden
356              Case Else                                'Fehler
357          End Select
358      End If
359
360  End Sub
361
362
363  '-----
364  Function M_value(byval Adc_val As Word) As Byte
365  ' Diese Function bekommt den AD-Wandler-Wert vom Mode-Schalter
366  ' und setzt ihn in einen Mode-Wert um.
367  ' Input:  Adc_val  0 bis 1023
368  ' Output: M_value  Mode 0 bis Mode 11
369
370
371      Select Case Adc_val
372          Case 77 To 97
373              M_value = 1                                'Mode 1
374          Case 160 To 180
375              M_value = 2                                'Mode 2
376          Case 246 To 266
377              M_value = 3                                'Mode 3
378          Case 331 To 351
379              M_value = 4                                'Mode 4
380          Case 416 To 436
381              M_value = 5                                'Mode 5
382          Case 502 To 522
383              M_value = 6                                'Mode 6
384          Case 587 To 607
385              M_value = 7                                'Mode 7
386          Case 672 To 692
387              M_value = 8                                'Mode 8
388          Case 758 To 778
389              M_value = 9                                'Mode 9
390          Case 843 To 863
391              M_value = 10                               'Mode 10
392          Case 928 To 948
393              M_value = 11                               'Mode 11
394          Case Else
395              M_value = 255                               'ungültiger Wert
396      End Select
397
398  End Function
399
400
401  '-----
402  Function T1_value(byval Adc_val As Word) As Byte
403  ' Diese Function bekommt den AD-Wandler-Wert von der Tastengruppe
404  ' Frequency und Memory und setzt ihn in T1_value um.
405  ' Input:  Adc_val  0 bis 1023

```

```

406 ' Output: T1_value 1 bis 4
407
408 Select Case Adc_val
409 Case 180 To 213 'Tastengruppe T1 Save (199)
410 T1_value = 4
411 Case 380 To 410 'Tastengruppe T1 Load (399)
412 T1_value = 3
413 Case 580 To 623 'Tastengruppe T1 > (591)
414 If T1_value = 255 Then
415 If Dek_pos > 0 Then
416 Dek_pos = Dek_pos - 1 'Tastenposition dekrementieren
417 End If
418 End If
419 T1_value = 2
420 Case 740 To 827 'Tastengruppe T1 < (768)
421 If T1_value = 255 Then
422 Dek_pos = Dek_pos + 1 'Tastenposition inkrementieren
423 If Dek_pos > 8 Then
424 Dek_pos = 8
425 End If
426 End If
427 T1_value = 1
428 Case Else
429 T1_value = 255 'ungültiger Wert
430 End Select
431
432 If T1_value <> 255 Then
433 Posp = 1
434 Var1 = Adc_val
435 Call Print_sm
436 End If
437
438 Adc_val = 0 'ADC_VAL rücksetzen
439
440 End Function
441
442
443 '-----
444 Function T2_value(byval Adc_val As Word) As Byte
445 ' Diese Function bekommt den AD-Wandler-Wert von der Tastengruppe
446 ' F1 bis F4 und setzt ihn in einen T2_value um.
447 ' Input: Adc_val 0 bis 1023
448 ' Output: T2_value 1 bis 4
449 '
450 Select Case Adc_val
451 Case 193 To 213
452 T2_value = 4 'Tastengruppe T2 F4
453 Nennfrequenz = 2000000 'Nennfrequenz in Hz, wenn F4 gedrückt
454 Case 377 To 407
455 T2_value = 3 'Tastengruppe T2 F3
456 Nennfrequenz = 3000000 'Nennfrequenz in Hz, wenn F3 gedrückt
457 Case 550 To 600
458 T2_value = 2 'Tastengruppe T2 F2
459 Nennfrequenz = 4000000 'Nennfrequenz in Hz, wenn F2 gedrückt
460 Case 750 To 800
461 T2_value = 1 'Tastengruppe T2 F1
462 Nennfrequenz = 5000000 'Nennfrequenz in Hz, wenn F1 gedrückt
463 Case Else
464 T2_value = 255 'ungültiger Wert
465 End Select
466
467 End Function
468
469
470 '-----
471 Function Tf_value(byval Adc_val As Word) As Byte
472 ' Diese Function bekommt den AD-Wandler-Wert vom Tastenfeld und setzt
473 ' ihn in einen Tastenwert Tast_value um.
474 ' Input: Adc_val 0 bis 1023
475 ' Output: Tast_value 0 bis 9 + * + #
476
477
478 Select Case Adc_val
479 Case 350 To 422
480 Tast_value = 1 '1
481 Case 630 To 710
482 Tast_value = 2 '2
483 Case 928 To 1000
484 Tast_value = 3 '3
485 Case 278 To 349
486 Tast_value = 4 '4
487 Case 566 To 629

```

```

488     Tast_value = 5                                '5
489     Case 835 To 927
490     Tast_value = 6                                '6
491     Case 199 To 277
492     Tast_value = 7                                '7
493     Case 495 To 565
494     Tast_value = 8                                '8
495     Case 771 To 834
496     Tast_value = 9                                '9
497     Case 423 To 494
498     Tast_value = 0                                '0
499     Case 80 To 198
500     Tast_value = 11                               '*'
501     Case 711 To 770
502     Tast_value = 12                               '#'
503     Case Else
504     Tast_value = 255                              'ungültiger Wert
505 End Select
506
507 End Function
508
509
510 '-----
511 Sub Print_sm
512 'In dieser Subroutine werden Testparameter für DEBUG-Zwecke über die
513 'serielle Schnittstelle zum PC gesendet @@@.
514 'Input:  Var1 = Wordvariable zum Printen
515 '        Posp = Position der Variablen im Programm
516
517     Print #2 , "* ";
518     Print #2 , Posp ; : Print #2 , "- " ; : Print #2 , Var1 ; : Print #2 , " ";
519     Print #2 , : Print #2,
520
521 End Sub
522
523
524 '-----
525 Sub Calc_ad9951()
526 'In dieser Subroutine wird das Frequenz Tunig Word für den DDS-Baustein AD9951
527 'berechnet und die Daten an die Subroutine "Set_AD9951" übergeben. Die Tabelle
528 'zur Berechnung steht am Ende der Software. Das verlangt BASIC so.
529 '
530 'Input:   Frequenz      Frequenz
531 'Output:  DDS-Baustein programmiert
532
533 'Berechnen des FTW (Frequenz Tunig Word)
534 Frequenz_1 = Frequenz                                'Frequenz übernehmen
535 Lul = Lookup(0 , Mult)                              'Bits pro Hz aus der Tabelle laden (erster Wert)
536 F_tw = Lul * Frequenz_1                             'Teilberechnung Frequenz Tunig Word
537 Print
538 For I = 1 To 8 Step 1
539     Lul = Lookup(i , Mult)                            'Bits pro Hz aus der Tabelle laden
540     Y1 = Lul * Frequenz_1                             'Arbeitsfrequenz * Bits pro Hz (Zwischenergebnis)
541     Lud = Lookup(i , Divi)                            'Dekade laden
542     Z1 = Y1 / Lud
543     F_tw = F_tw + Z1                                 'Frequenz Tunig Word zusammensetzen
544 Next I
545 Call Set_ad9951                                     'Daten an den DDS-Baustein übertragen
546
547 End Sub
548
549
550 '-----
551 Sub Set_ad9951
552 'In dieser Subroutine werden dem DDS-Baustein der Inhalt des Control Function Register 2 und
553 'das Frequenz Tuning (CFR4) übertragen.
554 '
555 'Input:   F_tw      Frequenz Tuning Word
556 'Output:  Daten im DDS-Baustein
557
558 'Init Datenübertragung zum DDS-Baustein
559 Reset_dds = 0 : Iouupdate_dds = 0 : Sdio_dds = 0 : Sclk_dds = 0
560 Tx_data = 0 : Tx_bit = 0
561
562 'Daten an das Control Function Register 1 (CFR1) übertragen
563 Tx_data = &B00000000                                'Adresse: Control Function Register 1 (CFR1) <0x00>
564 Call Transmit_dds                                    'Daten übertragen
565
566 Tx_data = &B00000000                                '<31:24> MSB High-Byte übertragen          <0x00>
567 Call Transmit_dds
568
569 Tx_data = &B00000000                                '<23:16> MSB Low-Byte übertragen          <0x00>

```

```

570 Call Transmit_dds
571
572 Tx_data = &B01100010 ' <15:8> LSB High-Byte übertragen <0x62>
573 Call Transmit_dds
574
575 Tx_data = &B01000000 ' <7:0> LSB Low-Byte übertragen <0x40>
576 Call Transmit_dds
577
578 Iouupdate_dds = 1 'Datenübernahme zum DDS-Baustein = 1
579 Waitus 20
580 Iouupdate_dds = 0 'Datenübernahme zum DDS-Baustein = 0
581
582 'Daten an das Control Function Register 2 (CFR2) übertragen
583
584 Tx_data = &B00000001 'Adresse: Control Function Register 2 (CF2) <0x01>
585 Call Transmit_dds 'Daten übertragen
586
587 Tx_data = &B00000000 ' <23:16> MSB - Byte <0x00>
588 Call Transmit_dds
589
590 Tx_data = &B00000000 ' <15:8> <0x00>
591 Call Transmit_dds
592
593 Tx_data = &B00000110 ' <7:0> LSB <0x06>
594 Call Transmit_dds 'RefMult; VCO; Charge Pump für BAVARIX
595 Print #2 , " "
596
597 Iouupdate_dds = 1 'Datenübernahme zum DDS-Baustein = 1
598 Waitus 20
599 Iouupdate_dds = 0 'Datenübernahme zum DDS-Baustein = 0
600
601 'Daten an das Control Function Register 4 (CFR4) übertragen
602 Tx_data = &B00000100 'Adresse: Control Function Register 4 (CFR4) <0x04>
603 Call Transmit_dds 'Daten übertragen
604
605 Hw = Highw(f_tw) 'Frequenz Tuning Word übernehmen und in 2 Words zerlegen
606 Lw = F_tw
607
608 Tx_data = High(hw) ' <31:24> MSB High-Byte übertragen
609 Call Transmit_dds
610
611 Tx_data = Low(hw) ' <23:16> MSB Low-Byte übertragen
612 Call Transmit_dds
613
614 Tx_data = High(lw) ' <15:8> LSB High-Byte übertragen
615 Call Transmit_dds
616
617 Tx_data = Low(lw) ' <7:0> LSB Low-Byte übertragen
618 Call Transmit_dds
619
620 Iouupdate_dds = 1 'Datenübernahme zum DDS-Baustein = 1
621 Waitus 20
622 Iouupdate_dds = 0 'Datenübernahme zum DDS-Baustein = 0
623
624 End Sub
625
626
627 '-----
628 Sub Transmit_dds()
629 'In dieser Subroutine wird ein Byte an den DDS-Baustein übertragen. Zum Test werden die selben
630 'Daten auch noch über die serielle Schnittstelle ausgegeben.
631 'Input: Tx_data
632 'Output: Daten werden zu DDS-Baustein geschickt
633
634 Tp4 = 1
635 For I = 7 To 0 Step -1 '8 Bit ausgeben, MSB zuerst
636 Hz1 = 2 ^ I 'richtiges Bit zur Übertragung auswählen
637 Tx_bit = Tx_data And Hz1
638 If Tx_bit <> 0 Then
639 Print #2 , "1"; 'Datenbit an der seriellen Schnittstelle ausgeben
640 Sdio_dds = 1 'Datenbit setzen
641 Tp5 = 1 'Test
642 Else
643 Print #2 , "0"; 'Datenbit an der seriellen Schnittstelle ausgeben
644 Sdio_dds = 0 'Datenbit rücksetzen
645 Tp5 = 0 'Test
646 End If
647 Waitus 20 'warten
648 Tp6 = 1 'Test
649 Sclk_dds = 1 'Clock generieren
650 Waitus 20 'warten
651 Sclk_dds = 0 'Clock zurücksetzen

```

```

652         Tp6 = 0                                'Test
653         Next I
654         Tp4 = 0
655
656     End Sub
657
658
659     '-----
660     Sub Drehgeber()
661         'Diese Subroutine dient zur Übernahme der im Interrupt gemessenen Drehbewegung.
662         'Die Änderung des Drehgebers seit dem letzten Durchlauf wird erfasst und
663         'vorzeichenrichtig zum Zaehler addiert. Die Änderung des Drehgebers wird dann
664         'gelöscht.
665         'Input:  Zaehler      noch nicht bearbeitete Zahlenänderung
666         '         Tune        noch nicht verrechnete Drehänderungen am Drehgeber
667
668         Zaehler = Zaehler + Tune                  'letzte Drehbewegung übernehmen
669         Tune = 0                                  'Drehbewegung löschen
670     End Sub
671
672
673     '-----
674     Sub Freq_aenderung()
675         'In dieser Subroutine wird bestimmt, welche Stelle in der Frequenzanzeige geändert werden
676         'soll. Die Variable "Zähler" wird entsprechend der angewählten Dekade multipliziert und die
677         'Cursor-Position am Display wird parametrisiert.
678         'Input:  Dek_Pos     angewählte Dekade
679         '         Zaehler     Zahlenänderung
680         'Output: Differenz   Frequenzänderung mit Vorzeichen
681         '         X           Cursor-Position am LCD-Display
682         '         Y           Zeile am Display
683         '         Zaehler     wieder zurück gesetzt
684         Select Case Dek_pos
685             Case 0
686                 Differenz = Zaehler * 1          'Änderung 1 Hz
687                 X = 12                          'Position 11 am LCD-Display bestimmen
688             Case 1
689                 Differenz = Zaehler * 10        'Änderung 10 Hz
690                 X = 11                          'Position 10 am LCD-Display bestimmen
691             Case 2
692                 Differenz = Zaehler * 100       'Änderung 100 Hz
693                 X = 10                          'Position 9 am LCD-Display bestimmen
694             Case 3
695                 Differenz = Zaehler * 1000      'Änderung 1 kHz
696                 X = 8                           'Position 7 am LCD-Display bestimmen
697             Case 4
698                 Differenz = Zaehler * 10000     'Änderung 10 kHz
699                 X = 7                           'Position 6 am LCD-Display bestimmen
700             Case 5
701                 Differenz = Zaehler * 100000    'Änderung 100 kHz
702                 X = 6                           'Position 5 am LCD-Display bestimmen
703             Case 6
704                 Differenz = Zaehler * 1000000   'Änderung 1 MHz
705                 X = 4                           'Position 3 am LCD-Display bestimmen
706             Case 7
707                 Differenz = Zaehler * 10000000  'Änderung 10 MHz
708                 X = 3                           'Position 2 am LCD-Display bestimmen
709             Case Else
710                 'Error
711         End Select
712         Y = 1                                    ' Zeile 1 auf dem LCD-Display bestimmen
713         Zaehler = 0                              ' Zähler bearbeitet und zurückgesetzt
714     End Sub
715
716
717     '-----
718     Sub N_frequenz()
719         'In dieser Subroutine wird die Nennfrequenz berechnet. Die Nennfrequenz ist die Frequenz,
720         'welche man einstellt. Also die Empfangsfrequenz ohne Berücksichtigung von Korrekturfaktoren
721         'und ZF.
722         'Input:  Differenz
723         '         Nennfrequenz (alt)
724         'Output: Nennfrequenz (neu)
725
726         Nennfrequenz = Nennfrequenz + Differenz  'Nennfrequenz berechnen
727         If Nennfrequenz > 30000000 Then
728             Nennfrequenz = 30000000             'maximale Nennfrequenz
729         End If
730
731         If Nennfrequenz < 0 Then
732             Nennfrequenz = 0                    'minimale Nennfrequenz
733         End If

```

```

734      Differenz = 0                                'Differenz auf 0 zurücksetzen
735
736      End Sub
737
738
739      '-----
740      Sub Frequenzanzeige()
741      'Diese Subroutine "Positioniere Frequenz Anzeige" wandelt die Variable
742      '"Frequenz" und setzt sie mit Abstand zwischen Hz, kHz und MHz wieder zusammen.
743      Local A As Byte
744      Local Laenge As Byte
745      Local Posi As Byte
746      Local Ziffer As String * 1
747
748      Freqstr = Str(nennfrequenz)
749      Laenge = Len(freqstr)
750
751      Locate 1 , 1 : Lcd "                "          'Zeile 1 löschen
752      Posi = Laenge - 0                               'Komma-Stelle      -- --- ---.x
753      Ziffer = Mid(freqstr , Posi , 1)
754      Locate 1 , 14 : Lcd "Hz"
755      Locate 1 , 12 : Lcd Ziffer
756      Locate 1 , 9 : Lcd "."                          'Linken Punkt unterdrücken, wenn Anzeige < 1.000.000
757      If Nennfrequenz < 1000000 Then
758          'nichts tun
759      Else
760          Locate 1 , 5 : Lcd "."
761      End If
762
763      Posi = Laenge - 1                               'Hz-Anzeige      -- --- --x.x
764      Ziffer = Mid(freqstr , Posi , 1)
765      Locate 1 , 11 : Lcd Ziffer
766
767      Posi = Laenge - 2                               'Hz * 10-Stelle  -- --- -xx.x
768      Ziffer = Mid(freqstr , Posi , 1)
769      Locate 1 , 10 : Lcd Ziffer
770
771      Posi = Laenge - 3                               'Hz * 100-Anzeige -- --- xxx.x
772      Ziffer = Mid(freqstr , Posi , 1)
773      Locate 1 , 8 : Lcd Ziffer
774
775      Posi = Laenge - 4                               'kHz-Stelle     -- --x xxx.x
776      Ziffer = Mid(freqstr , Posi , 1)
777      Locate 1 , 7 : Lcd Ziffer
778
779      Posi = Laenge - 5                               'kHz * 10-Stelle -- -xx xxx.x
780      Ziffer = Mid(freqstr , Posi , 1)
781      Locate 1 , 6 : Lcd Ziffer
782
783      Posi = Laenge - 6                               'kHz * 100-Stelle -- xxx xxx.x
784      Ziffer = Mid(freqstr , Posi , 1)
785      Locate 1 , 4 : Lcd Ziffer
786
787      Posi = Laenge - 7                               'MHz -Stelle    -x xxx xxx.x
788      Ziffer = Mid(freqstr , Posi , 1)
789      Locate 1 , 3 : Lcd Ziffer
790
791      Posi = Laenge - 8                               'MHz * 10-Stelle -x xxx xxx.x
792      Ziffer = Mid(freqstr , Posi , 1)
793      Locate 1 , 2 : Lcd Ziffer
794
795      Posi = Laenge - 9                               'MHz * 100-Stelle xx xxx xxx.x
796      Ziffer = Mid(freqstr , Posi , 1)
797      Locate 1 , 1 : Lcd Ziffer
798
799      End Sub
800
801
802      '-----
803      'Daten zur Berechnung des Frequenz Tuning Words. Diese werden in der Subroutine "Calc_ad9951"
804      'benötigt
805
806      Mult:
807          Data 10 , 3 , 2 , 4 , 4 , 4 , 0 , 6 , 2
808      Divi:
809          Data 1& , 10& , 100& , 1000& , 10000& , 100000 , 1000000 , 10000000 , 100000000
810

```