

<LIF5000\_STEUERRECHNER\_V0.08.BAS>

```
*****
*
*           Steuerprogramm für LIF5000
*           -----
*
*           Version      0.08
*
*           12.10.2007   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 @@@
```

```
'Include Files für ATMegal28 einbinden -----
$regfile = "m128def.dat"
```

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

```
'Zuweisungen -----
1hz Alias 180143985
Zf Alias 12000           'Zuweisung ZF = 12kHz
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
Reset_dds Alias PORTB.2 'Zuweisung Reset DDS-Baustein = PORTB.2
Strobe_dds Alias PORTB.3 'Zuweisung Strobe DDS-Baustein = PORTB.3
Clk_dds Alias PORTB.0  'Zuweisung Clock DDS-Baustein = PORTB.0
Data_dds Alias PORTB.5 'Zuweisung DATA DDS-Baustein = PORTB.5
T_links Alias PINA.0   'Zuweisung Taste "<"
T_rechts Alias PINA.1  'Zuweisung Taste ">"
T_load Alias PINA.2    'Zuweisung Taste "load"
T_save Alias PINA.3    'Zuweisung Taste "save"
```

```
'Initialisierung Serielle Schnittstelle -----
$crystal = 16000000     'Quarz 16,000MHz
$baud = 9600           '9600Baud
```

```
'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 Compiler-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
```

```
'Initialisierung Port-Konfiguration -----
DDRA.0 = 0           'Config T_links = Input
DDRA.1 = 0           'Config T_rechts = Input
DDRA.2 = 0           'Config T_load = Input
DDRA.3 = 0           'Config T_save = Input
DDRA.4 = 0           'Config Pina.4 = 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 SCLK zum DDS
DDRB.1 = 1           'Config Pinb.1 = Output ISP-Programmierung
DDRB.2 = 1           'Config Pinb.2 = Output Reset_DDS
DDRB.3 = 1           'Config Pinb.3 = Output Strobe zum DDS
DDRB.4 = 0           'Config Pinb.4 = Input READ_DATA_DDS
DDRB.5 = 1           'Config Pinb.5 = Output DATA zum DDS-Baustein
DDRB.6 = 0           'Config Pinb.6 = Input Drehschalter DR0
DDRB.7 = 0           'Config Pinb.7 = Input Drehschalter DR1

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
```

```
'Initialisierung der Ports -----
Dim PORTA.0 As Bit   'Taste 1 (Frequenz links)
Dim PORTA.1 As Bit   'Taste 2 (Frequenz rechts)
Dim PORTA.2 As Bit   'Taste 3 (load Memory)
Dim PORTA.3 As Bit   'Taste 4 (save Memory)
Dim PORTA.4 As Bit   'Reserve
Dim PORTA.5 As Bit   'Reserve
Dim PORTA.6 As Bit   'Reserve
Dim PORTA.7 As Bit   'Reserve

Dim PORTB.0 As Bit   'PB.0 = SCLK zum DDS
Dim PORTB.1 As Bit   'PB.1 = SCK von ISP-Programmierung
Dim PORTB.2 As Bit   'PB.2 = RESET zum DDS-Baustein (low = Reset)
Dim PORTB.3 As Bit   'PB.3 = STROBE zum DDS-Baustein
Dim PORTB.4 As Bit   'PB.4 = READ_DATA_DDS vom DDS-Baustein not used
Dim PORTB.5 As Bit   'PB.5 = DATA zum DDS-Baustein
Dim PORTB.6 As Bit   'PB.6 = Drehschalter DR0
Dim PORTB.7 As Bit   'PB.7 = Drehschalter DR1

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
```

```
'Initialisierung Testpunkte -----
```



```

Declare Function S_value(byval Adc_val As Word) As Byte           'In dieser Function wird der S-Wert aus der AGC-Spannung bestimmt
Declare Sub S_meter(byval S_wert As Byte)                       'In dieser Subroutine wird der S-Wert grafisch dargestellt
Declare Sub Add_analog_digital_s_wert                          'digitalen Vorregler in S-Wert einrechnen
Declare Sub Array_enter(byval S_wert As Byte) As Byte        'In diesem Array wird der aktuelle S-Wert abgelegt
Declare Sub Array_copy                                         'In diesem Array wird eine Array in ein anderes Array kopiert
Declare Sub Array_lcd                                          'Hier werden der Inhalt des aktuellen und des alten Arrays verglichen
Declare Sub Menue                                             'Hier wird das Laden und Speichern der Frequenzen abgearbeitet
Declare Sub Save_eeeprom                                       'Frequenz in EEPROM abspeichern
Declare Sub Load_eeeprom                                       'Frequenz aus EEPROM laden
Declare Sub Reset_preselector                                 'Alle Bandfilter aus im Preselector
Declare Sub Set_preselector                                   'Alle Ausgänge des IC8 setzen nur für Testbetrieb
Declare Sub Preselector                                       'Der Preselector wird in Abhängigkeit der Frequenz geschalten
Declare Sub Set_sensitivity                                   'Empfindlichkeit des LIF5000 selektieren
Declare Sub Tx_sensitivity()                                  'Sendet die Empfindlichkeit an den Preselektor
Declare Sub Calc_ad9951                                       'DDS-Baustein Frequenz berechnen
Declare Sub Set_ad9951                                       'DDS-Baustein AD9951 programmieren
Declare Sub Tst_prt                                          'Testroutine zum Ausdrucken vieler Variablen
Declare Sub Print_sm                                          'Testparameter ausdrucken
Declare Sub Menu_preselector                                 'Utility-Software für Preselektor
Declare Sub Transmit_dds                                    'In dieser Subroutine wird ein Byte an den DDS-Baustein übertragen

```

```

'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

On INTO Tasten                                             'Interrupt-Routine für Tasten
Config INTO = Falling                                       'Interrupt 0 bei H/L-Flanke auslösen
Enable INTO                                                 'Externen Interrupt 0 einschalten

On Timer0 T0_interrupt:                                   'Timer 0 generates a reptition time of 33ms
Config Timer0 = Timer , Prescale = 1024
Enable Timer0

Enable Interrupts                                         'Interrupts global zulassen

```

**Goto** Hauptprogramm

```

'Interrupts -----
Ondr0:                                                       'Int1-Routine überwacht den Drehschalter "Tune"
  Toggle PORTE.4                                           'Wenn Portb.7 high dann Rechtsdrehung
  If PINB.7 = 0 Then                                       'Links drehung
    Tune = Tune + 1
  Else
    Tune = Tune - 1
  End If
Return

```

```

Tasten:
  If T_links = 0 Then                                       'Wenn Taste < gedrückt increment Dek_pos
    Dek_pos = Dek_pos + 1
    If Dek_pos > 8 Then
      Dek_pos = 8
    End If
  End If
  If T_rechts = 0 Then                                       'Wenn Taste > gedrückt decrement Dek_pos
    If Dek_pos > 0 Then
      Dek_pos = Dek_pos - 1
    End If
  End If
  If T_load = 0 Then                                       'Wenn Taste "load" gedrückt increment load
    Load_m = Load_m + 1
    If Load_m > 8 Then
      Load_m = 8
    End If
  End If
  If T_save = 0 Then                                       'Wenn Taste "save" gedrückt decrement save
    Save_m = Save_m + 1
    If Save_m > 8 Then
      Save_m = 8
    End If
  End If
Return

```

```

T0_interrupt:
  Toggle Tp5                                               'TP5
  If Timeout_01 <> 0 Then
    Timeout_01 = Timeout_01 - 1
  End If
Return

```

'Hauptprogramm AVX-EVAL-Board -----  
**Hauptprogramm:**

```

  Print "Testprogramm fuer Steuerrechner LIF5000 mit ATMegal28 Version 0.x"

```

```

'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

```

```

'DDS-Baustein AD9951 rücksetzen
Reset_dds = 0                                             'DDS-Baustein reset
Waitus 20
Reset_dds = 1

```

```

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

```

```

'AD-Wandler starten
Start ADC

```

```

'Variable setzen
Hz0 = 0
Hz2 = 1
Mw = 50                                                    'Mittelwert ADC über 50 Messwerte
Mittelw = 0                                               'Mittelwert
Nennfrequenz_alt = 0                                       'Nennfrequenz alt rücksetzen
Load_m = 0                                                 'Taste Load gedrückt
Save_m = 0                                                 'Taste Save gedrückt

```

```

Call S_meter(255)                                           'Init S-Meter
Call Reset_preselector                                     'Preselector rücksetzen (alle Bandfilter aus)

```

```

'InitLcd durchführen.
Initlcd                                'LCD-Reset per Instruktion (HD4478-PowerOn-Reset reicht oft nicht)
Cls                                     'Anzeige löschen
Cursor On                               'LCD-Cursor einschalten
Locate 1 , 1 : Lcd " --- LIF5000 ---"  'Begrüßung auf dem LCD-Display

Wait 2                                  '2 Sekunden warten

Do
If Load_m <> 0 Or Save_m <> 0 Then
    Call Menue                            'Taste Load oder Save gedrückt
End If

'AD-Wandler
Channel = 0                             'ADC-Kanal für S-Meter stellen
Adc_val = Getadc(channel)                'ADC starten und Analogwert digitalisieren

If Mw <> 0 Then
    Mittelw = Mittelw + Adc_val           'Mittelwert erfassen, um zitterndes S-Meter zu unterdrücken
    Mw = Mw - 1
Else
    Adc_val = Mittelw / 50                'Mittelwert berechnen
    Mw = 50                               'aus 50 Messungen
    Mittelw = 0                           'und rücssetzen

    Var1 = Adc_val                         '<--- Test
    S_wert = S_value(adc_val)             'ADC-Value in S-Wert umwandeln
    Var2 = S_wert                         '<--- Test

    'S-Meter berechnen
    Call Add_analog_digital_s_wert        'digitalen Vorregler in S-Wert einrechnen

    Var3 = S_wert_1                        '<--- Test
    'S-Meter anzeigen
    Call S_meter(s_wert_1)                'S-Meteranzeige berechnen und anzeigen
    Locate Y , X : Lcd ""                 'Cursor in Frequenzanzeige richtig positionieren

    'Preselektor Verstärkung
    Call Set_sensitivity                  'Vorverstärker und Dämpfung auf Preselektor einstellen

    Call Print_sm                          'Print Testparameter <--- für Testzwecke

End If

Cursor On                                'LCD-Cursor einschalten

'Drehgeber auslesen und Änderung aktualisieren
Call Drehgeber                            'Drehgeber eingelesen
Call Freq_aenderung                       'Frequenzänderung berechnet
Call N_frequenz                           'Nennfrequenz berechnen
Locate Y , X : Lcd ""                     'Cursor setzen
If Nennfrequenz <> Nennfrequenz_alt Then
    Call Preselector                       'hat sich die Frequenz geändert?
    Frequenz = Nennfrequenz                'Bandfilter auf dem Presektor bestimmen
    Call Calc_ad9951                       'Frequenzübergabe
    Call Frequenzanzeige                   'DDS-Baustein programmieren
    Nennfrequenz_alt = Nennfrequenz        'Frequenz am LCD-Display anzeigen
End If

Loop

'-----
Sub Print_sm
'In dieser Subroutine werden Testparameter über die serielle Schnittstelle
'zum PC gesendet.

Hz2 = Hz2 - 1                             'decrement Hilfszelle
If Hz2 = 0 Then                             'nur darstellen, wenn Hz abgelaufen
    Hz2 = 5                                 'Hz neu aufziehen
    Print "ADC-Value " ; ; Print Var1 ; ; Print " " ;
    Print "S-Wert analog " ; ; Print Var2 ; ; Print " " ;
    Print "S-Wert-Anzeige " ; ; Print Var3 ; ; Print " " ;
    Print "Dämpfung " ; ; Print Var4 ; ; Print " " ;
    Print "Gain " ; ; Print Var5
    Print : Print
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

'Berechnen des FTW (Frequenz Tunig Word)
Frequenz = Frequenz + Zf                    'ZF aufaddieren
Frequenz_1 = Frequenz * 4                   'Frequenz_1 = (Empfangsfrequenz + ZF) * 4
Lul = Lookup(0 , Mult)                     'Bits pro Hz aus der Tabelle laden (erster Wert)
F_tw = Lul * Frequenz_1                     'Teilberechnung Frequenz Tunig Word
Call Tst_prt
Print
For I = 1 To 8 Step 1
    Lul = Lookup(i , Mult)                   'Bits pro Hz aus der Tabelle laden
    Y1 = Lul * Frequenz_1                     'Arbeitsfrequenz * Bits pro Hz (Zwischenergebnis)
    Lud = Lookup(i , Divi)                   'Dekade laden
    Z1 = Y1 / Lud
    F_tw = F_tw + Z1                         'Frequenz Tunig Word zusammensetzen
    'Call Tst_prt                             'hier könnten Zwischenergebnisse ausgegeben werden!
Next I

Call Tst_prt : Print                         'Ergebnis mit Zwischenwerten ausgeben

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

Print : Print "Set_ad9951"

```

```
'Init Datenübertragung zum DDS-Baustein
Reset_dds = 1 : Strobe_dds = 0 : Data_dds = 0 : Clk_dds = 0
Tx_data = 0 : Tx_bit = 0
```

```
'Daten an das Control Function Register 2 (CFR2) übertragen
```

```
Print : Print "CFR2"
Tx_data = &B00000001 'Adresse: Control Function Register 2 (CF2)
Call Transmit_dds 'Daten übertragen
Tx_data = &B00000000
Call Transmit_dds
Tx_data = &B00000000
Call Transmit_dds
Tx_data = &B00101110 'RefMult; VCO; Charge Pump
Call Transmit_dds
```

```
Strobe_dds = 1 'Datenübernahme zum DDS-Baustein = 1
Waitus 20
Strobe_dds = 0 'Datenübernahme zum DDS-Baustein = 0
```

```
'Daten an das Control Function Register 4 (CFR4) übertragen
```

```
Print : Print "CFR4"
Tx_data = &B00000100 'Adresse: Control Function Register 4 (CFR4)
Call Transmit_dds 'Daten übertragen
```

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

```
Tx_data = High(hw) 'MSB High-Byte übertragen
Call Transmit_dds
```

```
Tx_data = Low(hw) 'MSB Low-Byte übertragen
Call Transmit_dds
```

```
Tx_data = High(lw) 'LSB High-Byte übertragen
Call Transmit_dds
```

```
Tx_data = Low(lw) 'LSB Low-Byte übertragen
Call Transmit_dds
```

```
Strobe_dds = 1 'Datenübernahme zum DDS-Baustein = 1
Waitus 20
Strobe_dds = 0 'Datenübernahme zum DDS-Baustein = 0
```

```
End Sub
```

```
'-----
```

```
Sub Transmit_dds()
```

```
'In dieser Subroutine wird ein Byte an den DDS-Baustein übertragen. Zum Test werden die selben
'Daten auch noch über dieserielle Schnittstelle ausgegeben.
```

```
'Input: Tx_data
'Output: Daten werden zu DDS-Baustein geschickt
```

```
Print : Print Tx_data ; : Print " " ;
```

```
For I = 7 To 0 Step -1 '8 Bit ausgeben, MSB zuerst
```

```
Hz1 = 2 ^ I
```

```
Tx_bit = Tx_data And Hz1
```

```
If Tx_bit <> 0 Then 'richtiges Bit zur Übertragung auswählen
```

```
Print "1"; 'Datenbit an der seriellen Schnittstelle ausgeben
```

```
Data_dds = 1 'Datenbit setzen
```

```
Else 'Datenbit an der seriellen Schnittstelle ausgeben
```

```
Print "0";
```

```
Data_dds = 0 'Datenbit rücksetzen
```

```
End If
```

```
Waitus 20 'warten
```

```
Clk_dds = 1 'Clock generieren
```

```
Waitus 20 'warten
```

```
Clk_dds = 0 'Clock zurücksetzen
```

```
Next I
```

```
End Sub
```

```
'-----
```

```
Sub Tst_prt()
```

```
'Diese Subroutine dient nur dem Test im Debug-Mode. Es werden alle Variablen
'der Subroutine zur Berechnung des Frequenz Tuning Words über die serielle
'Schnittstelle ausgegeben.
```

```
Print "Frequenz_1 = " ; : Print Frequenz_1 ; : Print " " ;
```

```
Print "F_tw = " ; : Print F_tw ; : Print " " ;
```

```
Print "I = " ; : Print I ; : Print " " ;
```

```
Print "Lul = " ; : Print Lul ; : Print " " ;
```

```
Print "Lud = " ; : Print Lud ; : Print " " ;
```

```
Print "Y1 = " ; : Print Y1 ; : Print " " ;
```

```
Print "Z1 = " ; : Print Z1 ; : Print " " ;
```

```
Print : Print
```

```
End Sub
```

```
'-----
```

```
Sub Menue()
```

```
'Diese Subroutine verwaltet die Menuesteuerung für die Frequenzspeicherung.
'Drückt man die Taste "load" oder "save", so kommt man vom Hauptprogramm in
'dieses Menue. Mit dem Drehknopf bestimmt man nun den Speicherplatz MEM_01 bis
'MEM_20. Dann betätigt man entweder die Taste "load", um von dem gewählten
'Speicherplatz eine Frequenz zu laden, oder die Taste "save", um die einge-
'stellte Frequenz auf dem angewählten Speicherplatz abzuspeichern.
```

```
'Preselektor Testprogramm:
```

```
'Ist man in diesem Menue, so kommt man mit den Tasten <-- oder --> in das
```

```
'Preselektor-Menue.
```

```
'Output: Subroutine Load_eeeprom oder Save_eeeprom wird aufgerufen
```

```
Print "Menue"
```

```
Load_m = 0 : Save_m = 0 'Subroutine initialisieren
```

```
M_exit = 255 : Hz0 = 0 : Ms = 0 : Mem = 1 : Tune = 0
```

```
Cursor Off 'LCD-Cursor ausschalten
```

```
Tmp0 = Dek_pos 'Dek_pos zwischenspeichern
```

```
While M_exit <> 0 'wiederholen bis Exit
```

```
Select Case Ms
```

```
Case 0
```

```
Locate 2 , 0 : Lcd " " 'Menue-Zeile löschen
```

```
Ms = 1 '2te Zeile löschen
```

```
Case 1
```

```
Locate 2 , 1 : Lcd "load save" 'Menue-Zeile schreiben
```

```
If Mem < 10 Then '2te Zeile beschreiben
```

```
Locate 2 , 8 : Lcd "0" : Lcd Mem 'schreibe Memory Nr.
```

```
Else
```

```
Locate 2 , 8 : Lcd Mem 'schreibe Memory Nr.
```

```

End If
Ms = 2
Case 2
If T_load = 0 Or T_save = 0 Then           'Wenn "load" oder "save" gedrückt
Timeout_01 = 30                          'Wartezeit beginnen
Ms = 3
M_exit = 255                              'kein Exit
Else
Mem = Mem + Tune                          'Drehgeber einlesen
Tune = 0                                  'Drehbewegung übernehmen
If Mem < 1 Then                            'Drehgeber rücksetzen
Mem = 1                                    'Memory-Variable auf
                                           'Plausibilität prüfen
                                           'und korrigieren
End If
If Mem > 20 Then
Mem = 20
End If
Ms = 1                                     'Drehbewegung löschen
End If
If T_links = 0 Or T_rechts = 0 Then
Ms = 6                                     'Taste 1 (Frequenz links)
                                           'Taste 2 (Frequenz rechts)
Print "T_links OR T_rechts"              'Taste 2 (Frequenz rechts)
Waitms 200
If T_links = 0 And T_rechts = 0 Then
Call Menu_preselector                    'Taste 1 und 2 gedrückt
                                           'Utility Preselector ausfrufen
End If
Print "EXIT"
End If                                     'Exit
Case 3
If T_load = 0 Then                         'Wenn Taste load gedrückt
Ms = 4                                     'Load-Taste
M_exit = 255                              'kein Exit
End If
If T_save = 0 Then                         'Wenn Taste save gedrückt
Ms = 5                                     'Save-Taste
M_exit = 255                              'kein Exit
End If
If T_links = 0 Or T_rechts = 0 Then
Ms = 6                                     'Taste 1 (Frequenz links)
End If                                     'Exit
Case 4                                     'Taste "Load"
If T_load = 0 Then                         'Wenn Taste load gedrückt
Ms = 4
If Timeout_01 = 0 Then
Print "Taste Load"                       'Frequenz laden
Call Load_eeeprom                        'Frequenz laden
Ms = 6                                    'Exit
End If
Else
Ms = 0                                     'Laden abgebrochen
End If
Case 5                                     'Taste Save
If T_save = 0 Then                         'Wenn Taste save gedrückt
Ms = 5
If Timeout_01 = 0 Then
Print "Taste Save"                       'Frequenz speichern
Call Save_eeeprom                        'Exit
Ms = 6
End If
Else
Ms = 0                                     'Speicherung abgebrochen
End If
Case 6                                     'Exit
M_exit = 0
Case Else
Print "Error Menue-Steuerung"            'ungültiger Wert
End Select
Waitms 100
Wend
Load_m = 0 : Save_m = 0
Cursor On
Locate 2 , 1 : Lcd " "
Dek_pos = Tmp0
Call S_meter(255)
'Init S-Meter
End Sub

```

```

-----
Sub Load_eeeprom()
'In dieser Subroutine wird die Empfangsfrequenz aus dem gewünschten Speicherplatz
'des EEPROMs geladen.
'Input: Mem           gewünschter Speicherplatz
'Output: Nennfrequenz aktuelle Empfangsfrequenz

Select Case Mem
Case 1               'Frequenz aus Mem_01 laden
Nennfrequenz = Mem_01
Case 2               'Frequenz aus Mem_02 laden
Nennfrequenz = Mem_02
Case 3               'Frequenz aus Mem_03 laden
Nennfrequenz = Mem_03
Case 4               'Frequenz aus Mem_04 laden
Nennfrequenz = Mem_04
Case 5               'Frequenz aus Mem_05 laden
Nennfrequenz = Mem_05
Case 6               'Frequenz aus Mem_06 laden
Nennfrequenz = Mem_06
Case 7               'Frequenz aus Mem_07 laden
Nennfrequenz = Mem_07
Case 8               'Frequenz aus Mem_08 laden
Nennfrequenz = Mem_08
Case 9               'Frequenz aus Mem_09 laden
Nennfrequenz = Mem_09
Case 10              'Frequenz aus Mem_10 laden
Nennfrequenz = Mem_10
Case 11              'Frequenz aus Mem_11 laden
Nennfrequenz = Mem_11
Case 12              'Frequenz aus Mem_12 laden
Nennfrequenz = Mem_12
Case 13              'Frequenz aus Mem_13 laden
Nennfrequenz = Mem_13
Case 14              'Frequenz aus Mem_14 laden
Nennfrequenz = Mem_14
Case 15              'Frequenz aus Mem_15 laden
Nennfrequenz = Mem_15
Case 16              'Frequenz aus Mem_16 laden
Nennfrequenz = Mem_16
Case 17              'Frequenz aus Mem_17 laden
Nennfrequenz = Mem_17
Case 18              'Frequenz aus Mem_18 laden
Nennfrequenz = Mem_18
Case 19              'Frequenz aus Mem_19 laden

```

```

Nennfrequenz = Mem_19
Case 20 'Frequenz aus Mem_20 laden
Nennfrequenz = Mem_20
Case Else
Print "Error Load_eeeprom" 'ungültiger Wert
End Select
Print Mem ; : Print " " ; : Print Nennfrequenz

```

End Sub

```

Sub Save_eeeprom()

```

```

'In dieser Subroutine wird die Empfangsfrequenz in das EEPROM auf den gewünschten
'Speicherplatz abgespeichert.
'Input: Nennfrequenz aktuelle Empfangsfrequenz
'Mem gewünschter Speicherplatz
'Output: Daten im EEPROM abgespeichert

```

```

Select Case Mem
Case 1 'Frequenz in Mem_01 abspeichern
Mem_01 = Nennfrequenz
Case 2 'Frequenz in Mem_02 abspeichern
Mem_02 = Nennfrequenz
Case 3 'Frequenz in Mem_03 abspeichern
Mem_03 = Nennfrequenz
Case 4 'Frequenz in Mem_04 abspeichern
Mem_04 = Nennfrequenz
Case 5 'Frequenz in Mem_05 abspeichern
Mem_05 = Nennfrequenz
Case 6 'Frequenz in Mem_06 abspeichern
Mem_06 = Nennfrequenz
Case 7 'Frequenz in Mem_07 abspeichern
Mem_07 = Nennfrequenz
Case 8 'Frequenz in Mem_08 abspeichern
Mem_08 = Nennfrequenz
Case 9 'Frequenz in Mem_09 abspeichern
Mem_09 = Nennfrequenz
Case 10 'Frequenz in Mem_10 abspeichern
Mem_10 = Nennfrequenz
Case 11 'Frequenz in Mem_11 abspeichern
Mem_11 = Nennfrequenz
Case 12 'Frequenz in Mem_12 abspeichern
Mem_12 = Nennfrequenz
Case 13 'Frequenz in Mem_13 abspeichern
Mem_13 = Nennfrequenz
Case 14 'Frequenz in Mem_14 abspeichern
Mem_14 = Nennfrequenz
Case 15 'Frequenz in Mem_15 abspeichern
Mem_15 = Nennfrequenz
Case 16 'Frequenz in Mem_16 abspeichern
Mem_16 = Nennfrequenz
Case 17 'Frequenz in Mem_17 abspeichern
Mem_17 = Nennfrequenz
Case 18 'Frequenz in Mem_18 abspeichern
Mem_18 = Nennfrequenz
Case 19 'Frequenz in Mem_19 abspeichern
Mem_19 = Nennfrequenz
Case 20 'Frequenz in Mem_20 abspeichern
Mem_20 = Nennfrequenz
Case Else
Print "Error Save_eeeprom" 'ungültiger Wert
End Select

```

End Sub

```

Sub Menu_preselector

```

```

'Diese Subroutine verwaltet die Menuesteuerung für die Preselektor Utilities.
'Drückt man die Taste <--, dann kann man mit dem Drehknopf die Dämpfung ein oder
'ausschalten. Drückt man die Taste -->, dann kann man die Verstärkung G = 1 bis G = 4
'mit dem Drehknopf verändern. Der gewählte Zustand bleibt betehen, bis man diese
'Menuesteuerung verläßt. Dann wird der Preselektor wieder automatisch gesteuert.
'Output: Preselector wunschgemäß gesetzt
'GD_VAL nach Preselektor senden (decreasing)

```

```

Print "Menue Preselektor"
Load_m = 0 : Save_m = 0 'Subroutine initialisieren
M_exit = 255 : Hz0 = 0 : Ms = 0 : Mem = 0 : Tune = 0

```

```

Cursor Off 'LCD-Cursor ausschalten
Tmp0 = Dek_pos 'Dek_pos zwischenspeichern
While M_exit <> 0 'wiederholen bis Exit
Select Case Ms
Case 0
Locate 1 , 1 : Lcd "Preselector Util" 'Menue-Text
Locate 2 , 1 : Lcd " " '2te Zeile löschen
Locate 2 , 1 : Lcd "D = 0 G = 1" '2te Zeile beschreiben
If T_links = 1 And T_rechts = 1 Then
Ms = 2
Else
Ms = 1
End If
Case 1 'warten bis Tasten nicht mehr gedrückt
While T_links = 0 And T_rechts = 0
Wend
Ms = 2
Case 2 'Menue-Zeile schreiben
If T_links = 0 Then
Ms = 3 'Dämpfung ändern
Mem = 0 : Tune = 0
Print "linke Taste gedrückt"
End If
If T_rechts = 0 Then
Ms = 4 'Gain ändern
If Gain < 2 Then
Gain = 2
Print "Gain = 2"
End If
If Gain > 16 Then
Gain = 16
Print "Gain = 16"
End If
Print "rechte Taste gedrückt"
End If
Waitms 500
If T_links = 0 And T_rechts = 0 Then
Print "beide Tasten gedrückt"
Ms = 6 'Exit
End If
Case 3 'linke Taste gedrückt, Dämpfung ändern

```

```

If Mem <> Tune Then 'Drehbewegung übernehmen
Waitms 100
If Daempfung < 1 Then 'Dämpfung ändern
Daempfung = 1
Print "Daempfung = 1"
Else
Daempfung = 0
Print "Daempfung = 0"
End If
'Daten an Preselector übertragen
Gd_val = Daempfung + Gain 'Dämpfung und Gain kombiniert
Gd_val = Gd_val Xor 255 'negative Logik
I2cstart 'generiere Start
I2cwrite Pcf8574_ic13 'send slave Adresse
I2cwrite Gd_val 'send Value zm Setzen von Gain und Dämpfung
I2cstop 'generiere Stopp
End If
Mem = Tune
If T_rechts = 0 Then
Ms = 4 'Gain ändern
If Gain < 2 Then
Gain = 2
Print "Gain = 2"
End If
End If
If T_links = 0 And T_rechts = 0 Then
Print "beide Tasten gedrückt"
Ms = 6 'Exit
End If
Case 4
Waitms 100
If Mem > Tune Then 'Drehbewegung übernehmen
If Gain > 2 Then
Gain = Gain / 2
End If
End If 'Drehbewegung übernehmen
If Mem < Tune Then
If Gain < 16 Then
Gain = Gain * 2
End If
End If
Print "Gain = " ; : Print Gain
Mem = Tune
'Daten an Preselector übertragen
Gd_val = Daempfung + Gain 'Dämpfung und Gain kombiniert
Gd_val = Gd_val Xor 255 'negative Logik
I2cstart 'generiere Start
I2cwrite Pcf8574_ic13 'send slave Adresse
I2cwrite Gd_val 'send Value zm Setzen von Gain und Dämpfung
I2cstop 'generiere Stopp
If T_links = 0 Then
Ms = 3 'Dämpfung ändern
Mem = 0 : Tune = 0
Print "linke Taste gedrückt"
End If
If T_links = 0 And T_rechts = 0 Then
Print "beide Tasten gedrückt"
Ms = 6 'Exit
End If
Case 6 'Exit
M_exit = 0
Case Else
Print "Error Menue-Steuerung" 'ungültiger Wert
End Select
Waitms 100
Wend
Load_m = 0 : Save_m = 0 'Tasten rücksetzen
Cursor On 'LCD-Cursor einschalten
Locate 2 , 1 : Lcd " " '2te Zeile löschen
Dek_pos = Tmp0 'Dek_pos zurück laden
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 nicht verrechnete Drehänderungen am Drehgeber

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 parametrisiert.
'Input: Dek_Pos angewählte Dekade
' Zaehler Zahlenänderung
'Output: Differenz Frequenzänderung mit Vorzeichen
' X Cursor-Position am LCD-Display
' Y Zeile am Display
' Zaehler wieder zurück gesetzt
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
X = 3
Case Else
'Error
End Select
Y = 1
Zaehler = 0
End Sub

```

'Änderung 10 MHz  
'Position 2 am LCD-Display bestimmen

' Zeile 1 auf dem LCD-Display bestimmen  
' Zähler bearbeitet und zurückgesetzt

```

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)

Nennfrequenz = Nennfrequenz + Differenz
If Nennfrequenz > 30000000 Then
    Nennfrequenz = 30000000
End If

If Nennfrequenz < 50000 Then
    Nennfrequenz = 50000
End If
Differenz = 0
End Sub

```

'Nennfrequenz berechnen  
'maximale Nennfrequenz  
'minimale Nennfrequenz  
'Differenz auf 0 zurücksetzen

```

Sub Frequenzanzeige()
'Diese Subroutine "Positioniere Frequenz Anzeige" wandelt die Variable
'"Frequenz" und setzt sie mit Abstand zwischen Hz, kHz und MHz wieder zusammen.
Local A As Byte
Local Laenge As Byte
Local Posi As Byte
Local Ziffer As String * 1

Fregstr = Str(nennfrequenz)
Laenge = Len(fregstr)

Locate 1, 1 : Lcd " "
Posi = Laenge - 0
Ziffer = Mid(fregstr, Posi, 1)
Locate 1, 14 : Lcd "Hz"
Locate 1, 12 : Lcd Ziffer
Locate 1, 9 : Lcd "."
If Nennfrequenz < 1000000 Then
    'nichts tun
Else
    Locate 1, 5 : Lcd "."
End If

Posi = Laenge - 1
Ziffer = Mid(fregstr, Posi, 1)
Locate 1, 11 : Lcd Ziffer

Posi = Laenge - 2
Ziffer = Mid(fregstr, Posi, 1)
Locate 1, 10 : Lcd Ziffer

Posi = Laenge - 3
Ziffer = Mid(fregstr, Posi, 1)
Locate 1, 8 : Lcd Ziffer

Posi = Laenge - 4
Ziffer = Mid(fregstr, Posi, 1)
Locate 1, 7 : Lcd Ziffer

Posi = Laenge - 5
Ziffer = Mid(fregstr, Posi, 1)
Locate 1, 6 : Lcd Ziffer

Posi = Laenge - 6
Ziffer = Mid(fregstr, Posi, 1)
Locate 1, 4 : Lcd Ziffer

Posi = Laenge - 7
Ziffer = Mid(fregstr, Posi, 1)
Locate 1, 3 : Lcd Ziffer

Posi = Laenge - 8
Ziffer = Mid(fregstr, Posi, 1)
Locate 1, 2 : Lcd Ziffer

Posi = Laenge - 9
Ziffer = Mid(fregstr, Posi, 1)
Locate 1, 1 : Lcd Ziffer
End Sub

```

'Zeile 1 löschen  
'Komma-Stelle -- --- ---.x  
'Linken Punkt unterdrücken, wenn Anzeige < 1.000.000  
'Hz-Anzeige -- --- --x.x  
'Hz \* 10-Stelle -- --- -xx.x  
'Hz \* 100-Anzeige -- --- xxx.x  
'kHz-Stelle -- --x xxx.x  
'kHz \* 10-Stelle -- -xx xxx.x  
'kHz \* 100-Stelle -- xxx xxx.x  
'MHz -Stelle -x xxx xxx.x  
'MHz \* 10-Stelle -x xxx xxx.x  
'MHz \* 100-Stelle xx xxx xxx.x

```

Function S_value(byval Adc_val As Word) As Byte
'Diese Function bekommt den AD-Wandler-Wert und setzt ihn in einen S-Wert um.
' Input: Adc_val 0 bis 1023
' Output: S_value 0 bis 12
'         0 bis 9 = S0 bis S9
'         10 = S9 +20dB
'         11 = S9 +40dB
'         12 = S9 +60dB

Select Case Adc_val
Case 0 To 36
    S_value = 12
Case 37 To 253
    S_value = 11
Case 254 To 381
    S_value = 10
Case 382 To 469
    S_value = 9
Case 470 To 534
    S_value = 8
Case 535 To 619
    S_value = 7
Case 620 To 729
    S_value = 6
Case 730 To 809
    S_value = 5

```

'S12 = S9 + S3  
'S11 = S9 + S2  
'S10 = S9 + S1  
'S9  
'S8  
'S7  
'S6  
'S5

```

Case 810 To 844
  S_value = 4           'S4
Case 845 To 860
  S_value = 3           'S3
Case 861 To 862
  S_value = 2           'S2
Case 863 To 865
  S_value = 1           'S1
Case 866 To 1023
  S_value = 0           'S0
Case Else
  S_value = 255         'ungültiger Wert
End Select

```

End Function

-----

```

Sub Add_analog_digital_s_wert()
  'In dieser Subroutine wird der S-Wert in Abhängigkeit der digitale Vor-
  'regler bestimmt. Das Ergebnis wird dann zur Anzeige weitergeleitet. @@@
  'Input:  S_wert      S-Meter-Wert vom Linearregler
  '        Daempfung  Vorteiler im Preselektor
  '        Gain       Vorverstärker im Preselektor
  'Output: S_wert_1    S-Meter-Wert zur Anzeige

```

```

Var4 = Daempfung           '<--- Test
Var5 = Gain                '<--- Test

```

```

If Daempfung = 0 Then
  Select Case Gain
    Case 2
      S_wert_1 = S_wert
    Case 4
      S_wert_1 = S_wert
    Case 8
      S_wert_1 = 10
    Case 16
      S_wert_1 = 10
    Case Else
      Print "Error Add_analog_digital_s_wert__Dämpfung = 0"
  End Select

```

```

Else
  Select Case Gain
    Case 2
      S_wert_1 = 10
    Case 4
      S_wert_1 = 10
    Case 8
      S_wert_1 = 11
    Case 16
      S_wert_1 = 12
    Case Else
      Print "Error Add_analog_digital_s_wert__Dämpfung = 1"
  End Select
End If

```

End Sub

-----

```

Sub S_meter(byval S_wert_1 As Byte) As Byte
  'In dieser Subroutine wird der S-Wert sowohl grafisch wie auch numerisch
  'auf dem LCD-Display dargestellt.
  'Input:  S-wert      0 bis 12 ist ein gültiger S-Wert 255 ist Initialisierung
  'Output: LCD-Display
  '        S-Wert_1    Wert zum Schalten der Vorverstärker

```

```

If S_wert_1 <> S_wert_alt Then
  S_meter = 255           '255 = S-Wert hat sich geändert

```

```

  Select Case S_wert_1
    Case 0 To 9
      'S-Wert digital darstellen
      Locate 2 , 11 : Lcd "S   " ; : Locate 2 , 12 : Lcd S_wert
    Case 10
      'S-Wert S9 +20dB darstellen
      Locate 2 , 11 : Lcd "S9 +20dB"
    Case 11
      'S-Wert S9 +40dB darstellen
      Locate 2 , 11 : Lcd "S9 +40dB"
    Case 12
      'S-Wert S9 +60dB darstellen
      Locate 2 , 11 : Lcd "S9 +60dB"
    Case 255
      'Initialisiere S-Wert
      S_wert_1 = 0
      'S-Wert digital darstellen
      Locate 2 , 11 : Lcd "S   " ; : Locate 2 , 12 : Lcd S_wert
  End Select

```

```

  Print "Error S-meter"
End Select
Locate Y , X : Lcd ""           'Cursor in Frequenzanzeige richtig positionieren
'S-Meter Wert in ein Array eintragen
Call Array_enter(s_wert_1)
'Differenz zwischen altem und neuen S-Wert ans LCD-Display übertragen
Call Array_lcd
Locate Y , X : Lcd ""           'Cursor in Frequenzanzeige richtig positionieren
'S-Meter Wert alt zwischenspeichern
Call Array_copy

```

```

End If
S_wert_alt = S_wert_1         'S-Wert zwischenspeicher

```

End Sub

-----

```

Sub Array_enter(byval S_wert As Byte) As Byte
  'In dieser Subroutine wird ein Array mit dem aktuellen S-Wert gefüllt. Jede
  'S-Stufe bekommt ein Zeichen (255) zugewiesen. Die höheren Werte bekommen den
  'Wert " ".
  'Input:  S_wert
  'Output: Array_swert

```

```

Dim Ca As Byte

```

```

Dim Zeichen As Byte

```

```

'now fill the array

```

```

For Ca = 1 To 9

```

```

  If S_wert >= Ca Then
    Zeichen = 255
  Else
    Zeichen = " "
  End If

```

```

  Array_swert(ca) = Zeichen           'Zeichen ins Array eintragen

```

Next

End Sub

```
'-----  
Sub Array_copy()  
'In dieser Subroutine wird ein Array in ein anderes Array gleicher Grösse kopiert.  
'Input: Array_swert  
'Output: Array_swert_alt
```

```
Dim Cb As Byte  
'now fill the array  
For Cb = 1 To 9  
'assign array element  
Array_swert_alt(Cb) = Array_swert(Cb)  
Next
```

End Sub

```
'-----  
Sub Array_lcd()  
'In dieser Subroutine wird ein aktuelles Array mit einem alten Array verglichen.  
'Werden Differenzen erkannt, so werden dies an das LCD-Display gesendet.  
'Input: Array_swert  
'Array_swert_alt  
'Output: LCD-Display
```

```
Dim Cd As Byte  
For Cd = 1 To 9  
If Array_swert_alt(Cd) <> Array_swert(Cd) Then  
Zeichen = Array_swert(Cd)  
Locate 2 , Cd : Lcd Chr(zeichen)  
Locate Y , X : Lcd "" 'Cursor in Frequenzanzeige richtig positionieren  
End If  
Next
```

End Sub

```
'-----  
Sub Reset_preselector()  
'In dieser Subroutine wird das IC8 auf dem Preselector-Board rückgesetzt.  
'Es werden alle Bandfilter abgeschaltet.  
'Input: kein Input  
'Output: IC8 auf dem Preselector ist rückgesetzt (kein Bandfilter selctiert)
```

```
I2cstart 'generiere Start  
I2cwrite Pcf8574_ic8 'send slave Adresse  
I2cwrite 255 'send 255 zum Rücksetzen  
I2cstop 'generiere Stopp  
End Sub
```

```
'-----  
Sub Set_preselector()  
'In dieser Subroutine werden alle Ausgänge des IC8 auf dem Preselector-Board gesetzt.  
'Dies ist eine Testroutine, welche im normalen Betrieb des LIF5000 nicht vorkommt.  
'Input: kein Input  
'Output: Alle Ausgänge des IC8 auf dem Preselector sind gesetzt
```

```
I2cstart 'generiere Start  
I2cwrite Pcf8574_ic8 'send slave Adresse  
I2cwrite 128 'send 128 zum Setzen  
I2cstop 'generiere Stopp  
End Sub
```

```
'-----  
Sub Preselector()  
'In dieser Subroutine werden die richtigen Bandfilter in Abhängigkeit der Empfangsfrequenz  
'ausgewählt und eingeschaltet.  
'Input: Nennfrequenz  
'Output: IC8 im Preselector richtig geschaltet
```

```
Select Case Nennfrequenz  
Case 50000 To 499999  
'Bandfilter A 50kHz bis 500kHz ein  
Tmp0 = 1  
Case 500000 To 1599999  
'Bandfilter B 500kHz bis 1,6MHz ein  
Tmp0 = 2  
Case 1600000 To 2469999  
'Bandfilter C 1,6MHz bis 2,47MHz ein  
Tmp0 = 4  
Case 2470000 To 4069999  
'Bandfilter D 2,47MHz bis 4,07MHz ein  
Tmp0 = 8  
Case 4070000 To 6699999  
'Bandfilter E 4,07MHz bis 6,7MHz ein  
Tmp0 = 16  
Case 6700000 To 11049999  
'Bandfilter F 6,7MHz bis 11,05 MHz ein  
Tmp0 = 32  
Case 11050000 To 18199999  
'Bandfilter G 11,05MHz bis 18,2MHz ein  
Tmp0 = 64  
Case 18200000 To 30000000  
'Bandfilter H 18,2MHz bis 30MHz ein  
Tmp0 = 128  
Case Else  
Print "Error Preselector"  
End Select  
Tmp0 = Tmp0 Xor 255 'negative Logik  
I2cstart 'generiere Start  
I2cwrite Pcf8574_ic8 'send slave Adresse  
I2cwrite Tmp0 'send Value zum Setzen des richtigen Bandfilters  
I2cstop 'generiere Stopp
```

End Sub

```
'-----  
Sub Set_sensitivity()  
'In dieser Subroutine wird die Empfindlichkeit des LIF5000 auf dem Preselector  
'bestimmt. Dazu wird der Vorteiler und der Verstärker je nach Bedarf ge-  
'schaltet.  
'Input: S-meter Interner S-Meter-Wert  
'Output: Rx_sens Empfindlichkeit RX
```

```

If S_meter = 255 Then
  If S_wert < 7 Then
    'Vorverstärker empfindlicher
    If Rx_sens > 0 Then
      Rx_sens = Rx_sens - 1
    End If
  End If
  If S_wert > 9 Then
    'Vorverstärker unempfindlicher
    If Rx_sens < 7 Then
      Rx_sens = Rx_sens + 1
    End If
  End If
End If

Call Tx_sensitivity          'berechnete Empfindlichkeit übertragen

```

```
End Sub
```

```
Sub Tx_sensitivity()
```

```

'In dieser Subroutine wird die Empfindlichkeit an den Preselector
'übertragen.
'Input: Rx_sens      Empfindlichkeit RX
'Output: Vorteiler und Verstärker im Preselector richtig geschaltet

```

```
Select Case Rx_sens
```

```

Case 0
  'höchste Empfindlichkeit einstellen
  Daempfung = 0 : Gain = 2

```

```

Case 1
  'nächst kleinere Empfindlichkeit
  Daempfung = 0 : Gain = 4

```

```

Case 2
  'nächst kleinere Empfindlichkeit
  Daempfung = 0 : Gain = 8

```

```

Case 3
  'nächst kleinere Empfindlichkeit
  Daempfung = 0 : Gain = 16

```

```

Case 4
  'nächst kleinere Empfindlichkeit
  Daempfung = 1 : Gain = 2

```

```

Case 5
  'nächst kleinere Empfindlichkeit
  Daempfung = 1 : Gain = 4

```

```

Case 6
  'nächst kleinere Empfindlichkeit
  Daempfung = 1 : Gain = 8

```

```

Case 7
  'nächst kleinere Empfindlichkeit
  Daempfung = 1 : Gain = 16

```

```

Case Else
  Print "Error Tx_sensitivity"
End Select

```

```

Gd_val = Daempfung + Gain          'Dämpfung und Gain kombiniert
Gd_val = Gd_val Xor 255            'negative Logik

```

```

If Gd_val <> Gd_val_old Then       'nur übertragen wenn geändert
  'Daten an Preselector übertragen
  I2cstart                          'generiere Start
  I2cwbyte Pcf8574_ic13             'send slave Adresse
  I2cwbyte Gd_val                   'send Value zum Setzen von Gain und Dämpfung
  I2cstop                           'generiere Stopp
  Gd_val_old = Gd_val               'alten Wert speichern
End If

```

```
End Sub
```

```
End          'end program
```

```

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

```

```

Mult:
  Data 10 , 7 , 3 , 7 , 4 , 1 , 8 , 2 , 4

```

```

Divi:
  Data 1& , 10& , 100& , 1000& , 10000& , 100000 , 1000000 , 10000000 , 100000000

```