

```

'-----
' Frequenzzähler, mit externem Interrupt, Atmega64, 4x LS7060
'-----
'letzte Änderung 6.10.17 Dieter's ZF Kalibrierungs Überarbeitung

$regfile = "m64def.DAT"           'Atmega64
$crystal = 10000000                '10MHz externer Oszillator
$hwstack = 100
$swstack = 128
$framesize = 288

Waitms 10
'----- Konfiguration für LCD EA DOGM162 -----
Config Lcdpin = Pin , Db4 = Portb.5 , Db5 = Porta.3 , Db6 = Porte.1 , _
Db7 = Porte.7 , E = Porte.6 , Rs = Porta.7

Waitms 10
Config Lcd = 16 * 2 , Chipset = Dogm162v5 , Contrast = &H78
'lowest Contrast = &H70
'highest Contrast = &H7F

Waitms 10

Cursor Off
Cls
Waitms 10

'----- SPI-Hard Definition -----
Config Spi = Hard , Interrupt = Off , Data Order = Msb , Master = Yes , _
Polarity = Low , Phase = 0 , Clockrate = 4 , Noss = 1
Spiinit

'----- Definition der Variablen -----
Dim E_dummy As Eram Byte           'Dummy zur Belegung des ersten Bytes

Dim E_calwert1 As Eram Long         'Speichert die Calwerte im EEPROM
Dim E_calwert2 As Eram Long
Dim E_calwert3 As Eram Long
Dim E_calwert4 As Eram Long

Dim Calwert As Long                'beinhaltet den Kalibrierwert
Const Null = 0

Dim Zahl As Long                   'ausgelesene Bits werden aufaddiert
Dim Zahl1 As String * 18           'nimmt Zahl als String auf
Dim Zahl2 As Long                  'Endergebnis
Dim Zahl3 As Long                  'zur Berechnung des Calwertes
Dim Osc1 As Long                   'Ergebnis der Oszillatormessung1
Dim Osc2 As Long                   'Ergebnis der Oszillatormessung2
Dim Osc3 As Long                   'Ergebnis der Oszillatormessung3
Dim Osc4 As Long                   'Ergebnis der Oszillatormessung4
Dim Zeile1 As String * 16          'Display Zeile 1
Dim Zeile2 As String * 16          'Display Zeile 2
Dim Count_ended As Byte            'Count beendet = 1
Dim Gatezeit1 As Byte              'Gatezeit1 festlegen
Dim Zf5 As Long                    'feste ZF von 33,3MHz
Dim Gespeichert As Byte
Dim Laenge As Byte
Dim Laenge1 As Byte
Dim Zahl_minus As Byte
Dim Minus1 As String * 1

Dim Test1 As Long
Dim Test2 As Long
Dim Test3 As Long
Dim Test4 As Long
Dim Test5 As String * 7
Dim Temp As Long
Dim Test10 As Byte

Dim 2_56ms As Byte
Dim 25_6ms As Byte
Dim 256ms As Byte
Dim 2_56s As Byte

```

```

2_56ms = 0
25_6ms = 1
256ms = 2
2_56s = 3

Dim Span As Byte           'die Signale SP1-SP5 werden eingelesen
Dim Span_alt As Byte
Dim Span_neu As Byte
Dim Localosc As Byte      'Auswahl der Bereiche durch SP6 + SP7

Dim Ue As Byte
Ue = &B10000001           'Das Zeichen ü in binär

Dim Zahlhigh As String * 5 'Anzeige vor dem Komma
Dim Zahllow As String * 9  'Anzeige nach dem Komma
Dim Unguelstig1 As String * 16 'Anzeige des Wortes "Frequenz"
Dim Unguelstig2 As String * 16 'Anzeige des Wortes "ungültig"
Dim Oven As String * 4
Dim Position As String * 4
Dim Source As String * 4

'----- Variablen für MCP3204 -----
Dim A As Byte
Dim Ch0 As Byte
Dim Ch1 As Byte
Dim C(3) As Byte
Dim Werte_mcp As Word
Dim Adc1 As Word
Dim Adc2 As Word
Dim Schleife As Byte
Ch0 = &B01100001           'Channel 0
Ch1 = &B01100101           'Channel 1

'-----Variablen für Potiberechnungen aus ADC-Werten-----
Dim Osc1_zw As Long
Dim Osc2_zw As Long
Dim Osc3_zw As Long
Dim Osc4_zw As Long
Dim Osc1_alt As Long
Dim Osc3_alt As Long
Dim Osc1_mem As Single
Dim Osc3_mem As Single
Dim Osc1_single As Single
Dim Osc3_single As Single
Dim Osc3_disp1 As Long
Dim Osc3_disp2 As Single
Dim Osc1_disp1 As Long
Dim Osc1_disp2 As Single
Dim Adc1_alt As Word
Dim Adc1_konst As Single
Dim Adc11_konst As Single
Dim Adc1_mem As Word
Dim Adc1_mem2 As Word
Dim Adc2_alt As Word
Dim Adc2_konst As Single
Dim Adc21_konst As Single
Dim Adc2_mem As Word
Dim Adc2_mem2 As Word
Dim Fragezeichen As String * 1
Fragezeichen = "?"
Dim Blank_fz As String * 1
Dim Flag As Byte

'-----zusätzliche Variablen für Kalibrierung-----
Dim Adc1kl As Word
Dim Adc1gr As Word
Dim Adc2kl As Word
Dim Adc2gr As Word
Dim Fgkl As Long
Dim Fggr As Long
Dim Ffkl As Long
Dim Ffgr As Long
Dim Adc1diff As Word
Dim Adc2diff As Word
Dim Fgdiff As Long
Dim Ffdiff As Long
Dim E_calwert5 As Eram Single
Dim E_calwert6 As Eram Single
Dim Pointer_calwert_setzen As Byte

```

```

Dim Pointer_cal_zf As Byte
Dim Pointer_cal_adc As Byte
Dim Pointer_fg_kleiner As Byte
Dim Pointer_fg_groesser As Byte
Dim Pointer_ff_kleiner As Byte
Dim Pointer_ff_groesser As Byte
Dim Pointer_berechnung As Byte

```

```

Dim Adc101 As Single
Dim Adc102 As Single
Dim Adc103 As Single
Dim Adc104 As Single
Dim Adc201 As Single
Dim Adc202 As Single
Dim Adc203 As Single
Dim Adc204 As Single

```

```

Adc101 = E_calwert5
Adc102 = E_calwert5
Adc103 = E_calwert5
Adc104 = E_calwert5
Adc201 = E_calwert6
Adc202 = E_calwert6
Adc203 = E_calwert6
Adc204 = E_calwert6
Adc101 = Adc101 / 1000
Adc102 = Adc102 / 100
Adc103 = Adc103 / 10
Adc201 = Adc201 / 1000
Adc202 = Adc202 / 100
Adc203 = Adc203 / 10

```

```

'-----

```

```

Const Blank0 = "" 'Keine Leerstelle
Const Blank1 = " " '1 Leerstelle
Const Blank2 = "  " '2 Leerstellen
Const Blank3 = "   " '3 Leerstellen
Const Blank4 = "    " '4 Leerstellen
Const Blank5 = "     " '5 Leerstellen
Const Blank6 = "      " '6 Leerstellen
Const Blank7 = "       " '7 Leerstellen
Const Blank8 = "        " '8 Leerstellen
Const Blank16 = "         " '16 Leerstellen

```

```

Const Komma = ","
Const Punkt = "."
Const Plus = "+"
Const Negativ = "-"

```

```

Minus1 = Plus 'Startwert = positiv
Const Mhz = "MHz"

```

```

Unguelteig1 = Blank4 + "Frequenz" + Blank4
Unguelteig2 = Blank4 + "ung" + Chr(ue) + "ltig" + Blank4

```

```

Config Pina.0 = Input 'D4, Counter Bit 4
Config Pina.1 = Input 'D5, Counter Bit 5
Config Pina.2 = Input 'D3, Counter Bit 3
Config Pina.4 = Input 'Cent
Config Pina.5 = Input 'Ext
Config Pina.6 = Input 'Cold

```

```

Porta.0 = 1
Porta.1 = 1 'Pullup-Widerstände einschalten
Porta.2 = 1
Porta.4 = 1
Porta.5 = 1
Porta.6 = 1

```

```

Config Portb.0 = Output      '/CS  Chip Select ADC15
'PortB.1 = SCK vom ISP und ADC-Clock
'PortB.2 = ADC16
'PortB.3 = ADC18= ADC-Out
Config Pinb.4 = Output      'Abbruch
Config Portb.6 = Input      'D2
Config Portb.7 = Input      'D6
Config Pinc.0 = Input       'SP6
Config Pinc.1 = Input       'SP5
Config Pinc.2 = Input       'SP4
Config Pinc.3 = Input       'SP3
Config Pinc.4 = Input       'SP2
Config Pinc.5 = Input       'SP1
Config Portc.6 = Output     'OSC_OFF
Config Pinc.7 = Input       'SP7

Portc.0 = 1
Portc.1 = 1
Portc.2 = 1
Portc.3 = 1
Portc.4 = 1
Portc.5 = 1
Portc.7 = 1

Config Portd.0 = Input      'Trigger, PD2 SWP
Config Portd.1 = Output     'Counter_reset
Config Portd.2 = Output     'Counter_store
Config Portd.3 = Output     'Scanimpuls, ST0
Config Portd.4 = Output     'Enable1, ST1
Config Portd.5 = Output     'CountStart
Config Portd.6 = Output     'Time_B
Config Portd.7 = Output     'Time_A
Config Pine.0 = Input       'PortE.0 = MOSI vom ISP und PR4
                                'PortE.1 = MISO vom ISP und DB6
Config Pine.2 = Input       'Counter Bit 1
Config Pine.3 = Input       'Counter Bit 7
Config Pine.4 = Input       'Counter Bit 0
Config Pine.5 = Input       'Count_ended, Interrupt5

Config Portf.0 = Output     'PD9, Sweephold
Config Pinf.1 = Input       'PR3, Switch vorwärts
Config Pinf.2 = Input       'PR2, Switch speichern
Config Pinf.3 = Input       'PR1, kalibrieren
'PortF.4 = TCK von JTAG
'PortF.5 = TMS von JTAG
'PortF.6 = TDO von JTAG
'PortF.7 = TDI von JTAG

Portf.1 = 1
Portf.2 = 1
Portf.3 = 1
Porte.0 = 1

Config Debounce = 50        'verlängert die Wartezeit
                                'nach Tastendruck von 25 auf 50ms
                                'um Pellen zu verhindern

'    die 8 Bits der Zähler
Bit1 Alias Pine.4          'Bit0
Bit2 Alias Pine.2          'Bit1
Bit3 Alias Pinb.6          'Bit2
Bit4 Alias Pina.2          'Bit3
Bit5 Alias Pina.0          'Bit4
Bit6 Alias Pina.1          'Bit5
Bit7 Alias Pinb.7          'Bit6
Bit8 Alias Pine.3          'Bit7

Cent Alias Pina.4          'Center of Poti
Ext Alias Pina.5           'External Oscillator
Cold Alias Pina.6          'Oscillator not hot

Cs Alias Portb.0           'ADC_Chip-Select
Cs = 1

```

```

Sp6 Alias Pinc.0      'SP6
Sp5 Alias Pinc.1      'SP5
Sp4 Alias Pinc.2      'SP4
Sp3 Alias Pinc.3      'SP3
Sp2 Alias Pinc.4      'SP2
Sp1 Alias Pinc.5      'SP1
Osc_off Alias Portc.6 'shutoff Oscillator 1,52GHz if not used
Sp7 Alias Pinc.7      'SP7

Scan1 Alias Portd.3    'Impuls zum Weiterschalten der Auslesung
Enable1 Alias Portd.4  'Enable Signal zum Counter
Countstart Alias Portd.5 'Countstart zur Hardware
Time_a Alias Portd.7   'Time_A
Time_b Alias Portd.6   'Time_B

Pd9 Alias Portf.0      'PD9=Sweepfold
Pr3 Alias Pinf.1       'PR3=Vorwärts
Pr2 Alias Pinf.2       'PR2=Speichern
Pr1 Alias Pinf.3       'PR1=Calibration
Abbruch Alias Portb.4  'Count wird abgebrochen
Pr4 Alias Pine.0       'PR4= rückwärts

Abbruch = 0
Enable1 = 1            'Enable counter
Count_ended = 0       'Messung läuft

'-----
Zahl2 = 990000        'fiktiver Startwert
Scan1 = 1
Waitus 20

'----- Interrupts erlauben -----
Enable Interrupts

On Int0 Introutine0   'Bei Interrupt springe zu Introutine0
Enable Int0           'Enable Interrupt 0
Config Int0 = Rising  'Interrupt bei ansteigender Flanke

On Int5 Introutine5   'Bei Interrupt springe zu Introutine5
Enable Int5           'Enable Interrupt 5
Config Int5 = Rising  'Interrupt bei ansteigender Flanke

'=====

Cs = 0
Waitms 1              'Init MCP3204
Cs = 1
Waitms 1

'----- LOGO anzeigen -----

Locate 1, 1
Lcd " Takeda Riken "
Locate 2, 1
Lcd " TR4111A      "
Wait 1
Cls

'-----
'Spanwerte in Variable einlesen
Span = 0              'löschen
Span.0 = Sp5          'SP1 einlesen
Span.1 = Sp4          'SP2 einlesen
Span.2 = Sp3          'SP3 einlesen
Span.3 = Sp2          'SP4 einlesen
Span.4 = Sp1          'SP5 einlesen

Span_neu = Span
Span_alt = Span

```

```

!*****
!
!                               Hauptprogramm
!                               =====
!*****

```

Do

Anfang:

```

Localosc = 0                                'Abfrage der Frequenzbereiche
Localosc.0 = Sp7                            'Bereich1 = 1,5-3,5GHz
Localosc.1 = Sp6                            'Bereich2 = 2,5-4,5GHz
                                           'Bereich3 = 0-2GHz

If Localosc = 3 Then                      'Nur im Bereich3 ist der Oszillator 1,52GHz ein
    Osc_off = 0                            'einschalten
Else
    Osc_off = 1                            'ausschalten
End If

Span = 0                                    'löschen
Span.0 = Sp5                              'SP1 einlesen
Span.1 = Sp4                              'SP2 einlesen
Span.2 = Sp3                              'SP3 einlesen
Span.3 = Sp2                              'SP4 einlesen
Span.4 = Sp1                              'SP5 einlesen

Span_neu = Span

If Span_neu <> Span_alt Then
    Abbruch = 0
    Pulseout Portb , 1 , 100              'Impuls zum Abbrechen der Messung
    Waitus 40
    Portd.1 = 1
    Pulseout Portd , 1 , 100             'Reset-Impuls zum Zähler 40µs
    End If

Span_alt = Span_neu

Select Case Span                          'Torzeiten einstellen,
                                           'ZF5-Stellen festlegen
                                           'und Calwert aus EEprom holen
    Case 7 To 15 :
        Gatetime1 = 2_56ms
        Zf5 = 33
        Calwert = 0

    Case 29 To 31 :
        Gatetime1 = 2_56ms
        Zf5 = 33
        Calwert = 0

    Case 27 To 28 :
        Gatetime1 = 25_6ms
        Zf5 = 330
        Calwert = E_calwert2

    Case 24 To 26 :
        Gatetime1 = 256ms
        Zf5 = 3300
        Calwert = E_calwert3

    Case 20 To 23 :
        Gatetime1 = 2_56s
        Zf5 = 33000
        Calwert = E_calwert4

    Case Else :                          'Ungültige Span's anzeigen
        Zeile1 = "          Span          "
        Zeile2 = Unguelstig2
        Locate 1 , 1
        Lcd Zeile1
        Locate 2 , 1
        Lcd Zeile2

        Goto Anfang                      'Gehe zu Anfang, solange Span ungültig
End Select

```

```

'----- Auslesen des MCP3204 Ch0-----
Schleife = Schleife + 19          'Start der Schleife
If Schleife < 1 Then              'Auslesen 10x überspringen
Goto Ende
Else

A = Ch0

Gosub Mcp3204
Adc1 = Werte_mcp

'----- Auslesen des MCP3204 Ch1-----

A = Ch1

Gosub Mcp3204
Adc2 = Werte_mcp

Schleife = 1                      'Schleifenzahl zurücksetzen
Ende:                               'Sprung zu Ende
End If

'-----Abfrage ob mehr als 2 digit Änderung-----

Adc1_mem = Adc1_alt - 2
If Adc1 < Adc1_mem Then

Gosub Adc1_kleiner
End If

Adc1_mem = Adc1_alt + 2
If Adc1 > Adc1_mem Then

Gosub Adc1_groesser
End If

Adc2_mem = Adc2_alt - 2
If Adc2 < Adc2_mem Then

Gosub Adc2_kleiner
End If

Adc2_mem = Adc2_alt + 2
If Adc2 > Adc2_mem Then

Gosub Adc2_groesser
End If

'-----

If Count_ended = 1 Then           'Durchlauf nur 1x pro Triggerung
Count_ended = 0

Select Case Localosc              'Bereiche zur Berechnung auswählen
Case 1 :                          'abhängig von Span 6+7
Gosub Addition1
Case 2 :
Gosub Addition2
Case 3 :
Gosub Addition3
End Select
End If

'----- Angeigenformatierung -----

Gosub Mhz1

'----- zur Anzeige -----

Gosub Anzeigen

Loop

!*****
!                                     ENDE Hauptprogramm
!                                     =====
!*****

```

```

'-----'Externe Interruptroutine für INT0
'-----'Kommt vom Analyser
Introutine0:

    Pd9 = 1                                'Sweep anhalten
    Time_a = Gatetime1.0                    'Torzeit setzen
    Time_b = Gatetime1.1

    Portd.5 = 0
    Pulseout Portd , 5 , 100                'Triggerimpuls 40µs zum Starten der HW
Return

'-----'Externe Interruptroutine für INT5
'-----'Kommt wenn Messung beendet

Introutine5:

    Portd.2 = 1
    Pulseout Portd , 2 , 100                'Store-impuls 40µs wird gesendet

    Waitus 40
    Portd.1 = 1
    Pulseout Portd , 1 , 100                'Reset-impuls 40µs wird gesendet

    Waitus 40
    Gosub Ausleseroutine                    'Zähler auslesen

    Count_ended = 1

    Pd9 = 0                                'Sweep wieder starten
Return

'----- Anzeigeformat je nach Stellenanzahl hinter dem Komma festlegen -----

Mhz1:

    Zahl1 = Str(zahl2)                      'in String umwandeln
    Laenge = Len(zahl1)                     'Länge der Zahl ermitteln

    Select Case Gatetime1
    Case Is = 0 :                            'Gatezeit = 2,56ms
        If Laenge > 1 Then
            Zahl1low = Right(zahl1 , 1)     '1 Stelle nach dem Komma auswählen
            Laenge1 = Laenge - 1
            Zahl1high = Left(zahl1 , Laenge1) 'Stellen vor dem Komma auswählen
            End If

            Select Case Laenge                'Anzeige je nach Stellenzahl vor Komma
            Case 1
                Zahl1low = Zahl1
                Zahl1high = "0"
                'Stellenzahl 1  0,x
            Case 2
                Zeile1 = Blank5 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
                'Stellenzahl 2  x,x
            Case 3
                Zeile1 = Blank5 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
                'Stellenzahl 3  xx,x
            Case 4
                Zeile1 = Blank5 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
                'Stellenzahl 4  xxx,x
            Case 5
                Zeile1 = Blank4 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
                'Stellenzahl 5  xxxx,x
            End Select
        End Select

    Case Is = 1 :                            'Gatezeit = 25,6ms
        If Laenge > 1 Then
            Zahl1low = Right(zahl1 , 2)     '2 Stellen nach dem Komma auswählen
            Laenge1 = Laenge - 2
            Zahl1high = Left(zahl1 , Laenge1) 'Stellen vor dem Komma auswählen
            End If

        Select Case Laenge                    'Anzeige je nach Stellenzahl vor Komma

```



```

Case 1
    Zahl1low = "0" + Zahl1      'Stellenzahl 1   0,0x
    Zahl1high = "0"
Zeile1 = Blank5 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
Case 2
    Zahl1low = Zahl1            'Stellenzahl 2   0,xx
    Zahl1high = "0"

Zeile1 = Blank5 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
Case 3 :
    Zahl1low = "0" + Zahl1      'Stellenzahl 3   x,xx
    Zahl1high = "0"
Zeile1 = Blank5 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
Case 4 :
    Zahl1low = Zahl1            'Stellenzahl 4   xx,xx
    Zahl1high = "0"
Zeile1 = Blank4 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
Case 5 :
    Zahl1low = "0" + Zahl1      'Stellenzahl 5   xxx,xx
    Zahl1high = "0"
Zeile1 = Blank3 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
Case 6 :
    Zahl1low = Zahl1            'Stellenzahl 6   xxxx,xx
    Zahl1high = "0"
Zeile1 = Blank2 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
End Select

Case Is = 2 :
    'Gatezeit = 256ms
    If Laenge > 2 Then
        Zahl1low = Right(zahl1 , 3)      '3 Stellen nach dem Komma auswählen
        Laenge1 = Laenge - 3
        Zahl1high = Left(zahl1 , Laenge1) 'Stellen vor dem Komma auswählen
    End If

    Select Case Laenge
        'Anzeige je nach Stellenzahl vor Komma
        Case 1
            Zahl1low = "00" + Zahl1      'Stellenzahl 1   0,00x
            Zahl1high = "0"
Zeile1 = Blank4 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
        Case 2
            Zahl1low = "0" + Zahl1      'Stellenzahl 2   0,0xx
            Zahl1high = "0"
Zeile1 = Blank4 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
        Case 3 :
            Zahl1low = Zahl1            'Stellenzahl 3   0,xxx
            Zahl1high = "0"
Zeile1 = Blank4 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
        Case 4 :
            Zahl1low = "0" + Zahl1      'Stellenzahl 4   x,xxx
            Zahl1high = "0"
Zeile1 = Blank4 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
        Case 5 :
            Zahl1low = "0" + Zahl1      'Stellenzahl 5   xx,xxx
            Zahl1high = "0"
Zeile1 = Blank3 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
        Case 6 :
            Zahl1low = "0" + Zahl1      'Stellenzahl 6   xxx,xxx
            Zahl1high = "0"
Zeile1 = Blank2 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
        Case 7 :
            Zahl1low = "0" + Zahl1      'Stellenzahl 7   xxxx,xxx
            Zahl1high = "0"
Zeile1 = Blank1 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
    End Select

Case Is = 3 :
    'Gatezeit = 2,56s
    If Laenge > 3 Then
        Zahl1low = Right(zahl1 , 4)      '4 Stellen nach dem Komma auswählen
        Laenge1 = Laenge - 4
        Zahl1high = Left(zahl1 , Laenge1) 'Stellen vor dem Komma auswählen
    End If

    Select Case Laenge
        'Anzeige je nach Stellenzahl vor Komma
        Case 1
            Zahl1low = "000" + Zahl1      'Stellenzahl 1   0,000x
            Zahl1high = "0"
Zeile1 = Blank3 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
        Case 2
            Zahl1low = "00" + Zahl1      'Stellenzahl 2   0,00xx
            Zahl1high = "0"
Zeile1 = Blank3 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
        Case 3 :
            Zahl1low = "0" + Zahl1      'Stellenzahl 3   0,0xxx
            Zahl1high = "0"
Zeile1 = Blank3 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
        Case 4 :
            Zahl1low = Zahl1            'Stellenzahl 4   0,xxxx
            Zahl1high = "0"
Zeile1 = Blank3 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
        Case 5 :
            Zahl1low = "0" + Zahl1      'Stellenzahl 5   x,xxxx
            Zahl1high = "0"
Zeile1 = Blank3 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
        Case 6 :
            Zahl1low = "0" + Zahl1      'Stellenzahl 6   xx,xxxx
            Zahl1high = "0"
Zeile1 = Blank2 + Minus1 + Zahl1high + Komma + Zahl1low + Blank1 + Mhz + Blank1
    End Select

```

```

Case 7 : 'Stellenzahl 7 xxx,xxx
Zeile1 = Blank1 + Minus1 + Zahlhigh + Komma + Zahllo + Blank1 + Mhz + Blank1
Case 8 : 'Stellenzahl 8 xxxx,xxx
Zeile1 = Blank0 + Minus1 + Zahlhigh + Komma + Zahllo + Blank1 + Mhz + Blank1
End Select

```

End Select

Return

***** Routine zur Erfassung der Zählung *****

Ausleseroutine:

```

Enable1 = 0
Gosub Auslesen 'Auslesen Zähler1
Osc1 = Zahl

Gosub Auslesen 'Auslesen Zähler2
Osc2 = Zahl

Gosub Auslesen 'Auslesen Zähler3
Osc3 = Zahl

Gosub Auslesen 'Auslesen Zähler4
Osc4 = Zahl

Waitms 1

Enable1 = 1

Adc1_alt = Adc1
Adc2_alt = Adc2

Blank_fz = Blank1

```

Return

'----- Subroutine zur Anzeige -----
Anzeigen:

```

If Pr1 = 0 Then 'Wenn Schalter Calwert = ein
Gosub Calwert_setzen
Else

If Cold = 1 Then
Oven = "cold"
Else
Oven = "warm"
End If

If Cent = 0 Then
Position = "cent"
Else
Position = "left"
End If

If Ext = 1 Then
Source = "ext."
Else
Source = "int."
End If

Locate 1 , 1
Lcd Zeile1
Locate 1 , 16
Lcd Blank_fz
Locate 2 , 1
Lcd Blank1 ; Oven ; Blank1 ; Position ; Blank1 ; Source ; Blank1
End If
Return

```

```

'*****
'----- Subroutine zur Addition in den drei Bereichen -----
'Bereich1      1500-3500MHz
Addition1:

  Shift Osc2 , Right , 1
  Shift Osc3 , Right , 2
  Shift Osc4 , Right , 7

  Zahl2 = Osc1
  Zahl2 = Zahl2 - Osc3
  Zahl2 = Zahl2 - Osc4
  Zahl2 = Zahl2 + Zf5
  Zahl2 = Zahl2 - Calwert
  Minus1 = Plus

  Osc1_alt = Zahl2
  Osc3_alt = Zahl2
Return

'Bereich2      2500-4500MHz
Addition2:

  Shift Osc2 , Right , 1
  Shift Osc3 , Right , 2
  Shift Osc4 , Right , 7

  Zahl2 = Osc1
  Zahl2 = Zahl2 + Osc3
  Zahl2 = Zahl2 + Osc4
  Zahl2 = Zahl2 - Zf5
  Zahl2 = Zahl2 + Calwert
  Minus1 = Plus

  Osc1_alt = Zahl2
  Osc3_alt = Zahl2
Return

'Bereich3      0-2000MHz
Addition3:

  Shift Osc2 , Right , 1
  Shift Osc3 , Right , 2
  Shift Osc4 , Right , 7

  Zahl2 = Osc1
  Zahl2 = Zahl2 - Osc2
  Zahl2 = Zahl2 - Osc3
  Zahl2 = Zahl2 - Osc4
  Zahl2 = Zahl2 + Zf5

  If Pr1 = 0 Then
    Zahl3 = Zahl2
  Else
    Zahl2 = Zahl2 - Calwert
  End If

  If Zahl2 < 0 Then
    Zahl2 = Abs(zahl2)
    Minus1 = Negativ
  Else
    Minus1 = Plus
  End If

  Osc1_alt = Zahl2
  Osc3_alt = Zahl2
Return

```

```

'----- Subroutine zum Auslesen der Zähler LS7060 in die Variable Zahl -----
Auslesen:
    Zahl = 0
    'Stage1
Scan1 = 1
Waitus 5
    Zahl.0 = Bit1
    Zahl.1 = Bit2
    Zahl.2 = Bit3
    Zahl.3 = Bit4
    Zahl.4 = Bit5
    Zahl.5 = Bit6
    Zahl.6 = Bit7
    Zahl.7 = Bit8

Scan1 = 0
Waitus 5

'Ende Stage1
'-----
'Stage2
Scan1 = 1
Waitus 5
    Zahl.8 = Bit1
    Zahl.9 = Bit2
    Zahl.10 = Bit3
    Zahl.11 = Bit4
    Zahl.12 = Bit5
    Zahl.13 = Bit6
    Zahl.14 = Bit7
    Zahl.15 = Bit8

Scan1 = 0
Waitus 5

'Ende Stage2
'-----
'Stage3
Scan1 = 1
Waitus 5
    Zahl.16 = Bit1
    Zahl.17 = Bit2
    Zahl.18 = Bit3
    Zahl.19 = Bit4
    Zahl.20 = Bit5
    Zahl.21 = Bit6
    Zahl.22 = Bit7
    Zahl.23 = Bit8

Scan1 = 0
Waitus 5

'Ende Stage3
'-----
'Stage4
Scan1 = 1
Waitus 5
    Zahl.24 = Bit1
    Zahl.25 = Bit2
    Zahl.26 = Bit3
    Zahl.27 = Bit4
    Zahl.28 = Bit5
    Zahl.29 = Bit6
    Zahl.30 = Bit7
    Zahl.31 = Bit8

'Ende Stage4    Anfang Stage5
Scan1 = 0
Waitus 5
Return

```

```

'*****
'                               Erfassen des Korrekturwertes
'                               Immer im Bereich 0-2000MHz und Torzeit 2,56s ermitteln
'*****

```

```
Calwert_setzen:
```

```

If Pointer_cal_adc = 0 And Pointer_cal_zf = 0 Then
  Locate 1, 1
  Lcd Blank2 ; "Kalibrieren" ; Blank3
  Locate 2, 1
  Lcd "<ZF      Poti>"
  Debounce Pine.0 , 0 , Cal_zf , Sub  'Taste rückwärts
  Debounce Pinf.1 , 0 , Cal_adc , Sub  'Taste vorwärts

```

```
End If
```

```

If Pointer_cal_adc = 1 Then
  Gosub Cal_adc
End If
If Pointer_cal_zf = 1 Then
  Gosub Cal_zf
End If

```

```
Return
```

```
'----- Menue ZF-Korrektur -----
```

```
Cal_zf:
```

```
Pointer_cal_zf = 1
```

```
Debounce Pine.0 , 0 , Zurueck , Sub  'Zurück
```

```
If Gespeichert = 0 Then
```

```
  'Pointer_cal_zf = 1
```

```
  Locate 1, 1
```

```
  Lcd "Calwert= " ; Zahl3 ; Blank4
```

```
  Locate 2, 1
```

```
  Lcd "Taste= speichern"
```

```
  Debounce Pinf.2 , 0 , Speichern , Sub  'Taste speichern
```

```
End If
```

```

If Gespeichert = 1 Then
  Gosub Gespeichert_anzeigen
End If

```

```
Return
```

```
'----- ZF-Korrektur gespeichert -----
```

```
Gespeichert_anzeigen:
```

```
  Locate 1, 1
```

```
  Lcd "      Calwert      "
```

```
  Locate 2, 1
```

```
  Lcd " gespeichert      "
```

```
Debounce Pine.0 , 0 , Zurueck , Sub  'Zurück
```

```
Return
```

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

```
Zurueck:
```

```
Pointer_cal_zf = 0
```

```
Gespeichert = 0
```

```
Gosub Anfang  'zum Anfang des Hauptprogrammes
```

```
Return
```

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

```

Speichern:
    Gespeichert = 1

    'Unterroutine speichert den Calwert
    'Wenn calwert = 0 dann  EEPROM auf 0
    If Zahl3 = 0 Then
        E_calwert4 = Null
        E_calwert3 = Null
        E_calwert2 = Null
    Else
        E_calwert4 = Zahl3          '3 Stellen in EEPROM

        Temp = Zahl3 / 10
        E_calwert3 = Temp          '2 Stellen in EEPROM

        Temp = Temp / 10
        E_calwert2 = Temp          '1 Stelle in EEPROM

    End If

'----- Menue ADC-Kalibrieren -----

Cal_adc:
    Pointer_cal_adc = 1

    If Pointer_fg_kleiner = 1 Then
        Goto Fg_kleiner
    End If

    Locate 1, 1
    Lcd "ADC1  ADC2  "
    Locate 2, 1
    Lcd Adc1 ; Blank4 ; Adc2 ; Blank5
    If Pine.0 = 0 Then
        Waitms 100
        Pointer_cal_adc = 0
        Debounce Pine.0, 0, Calwert_setzen, Sub
    End If
    If Pinf.1 = 0 Then
        Waitms 100
        Pointer_fg_kleiner = 1
    End If

    Goto Fg_kleiner
End If

Return

'----- Menue untere Frequenz grob -----

Fg_kleiner:
    Pointer_fg_kleiner = 1

    If Pointer_fg_groesser = 1 Then
        Goto Fg_groesser
    End If

    Locate 1, 1
    Lcd "FG " ; Zahl2 ; Blank5

    If Adc1 > 400 Then
        Locate 2, 1
        Lcd "<400 ? " ; Adc1 ; Blank6
    Else
        Locate 2, 1
        Lcd "<400 " ; Adc1 ; Blank6
    End If

    If Adc1 < 400 And Pr2 = 0 Then
        Adc1kl = Adc1
        Fgkl = Zahl2
        Locate 2, 1
        Lcd "gespeichert" ; Blank4
        Waitms 1000
    End If

```

```

If Pine.0 = 0 Then
Waitms 100
Pointer_fg_kleiner = 0
Debounce Pine.0 , 0 , Cal_adc , Sub
End If

If Pinf.1 = 0 Then
Waitms 100
Pointer_fg_groesser = 1

Goto Fg_groesser
End If

Return

/-----Menue obere Frequenz grob -----

Fg_groesser:

Pointer_fg_groesser = 1

If Pointer_ff_kleiner = 1 Then
Goto Ff_kleiner
End If

Locate 1 , 1
Lcd "FG " ; Zahl2 ; Blank5

If Adc1 < 3800 Then
Locate 2 , 1
Lcd ">3800 ? " ; Adc1 ; Blank6
Else
Locate 2 , 1
Lcd ">3800 " ; Adc1 ; Blank6
End If

If Adc1 > 3800 And Pr2 = 0 Then

Adc1gr = Adc1
Fggr = Zahl2
Locate 2 , 1
Lcd "gespeichert" ; Blank4
Waitms 1000
End If

If Pine.0 = 0 Then
Waitms 100
Pointer_fg_groesser = 0
Debounce Pine.0 , 0 , Fg_kleiner , Sub
End If
If Pinf.1 = 0 Then
Waitms 100
Pointer_ff_kleiner = 1
'Debounce Pinf.1 , 0 , ff_kleiner , Sub
Goto Ff_kleiner
End If

Return

/----- Menue untere Frequenz fein -----

Ff_kleiner:

Pointer_ff_kleiner = 1

If Pointer_ff_groesser = 1 Then
Goto Ff_groesser
End If

Locate 1 , 1
Lcd "FF " ; Zahl2 ; Blank5

If Adc2 > 400 Then
Locate 2 , 1
Lcd "<400 ? " ; Adc2 ; Blank6
Else
Locate 2 , 1
Lcd "<400 " ; Adc2 ; Blank6
End If

```

```

If Adc2 < 400 And Pr2 = 0 Then

    Adc2kl = Adc2
    Ffkl = Zahl2
    Locate 2, 1
    Lcd "gespeichert" ; Blank4
    Waitms 1000
End If

If Pine.0 = 0 Then
Waitms 100
Pointer_ff_kleiner = 0
Debounce Pine.0, 0, Fg_groesser, Sub
End If
If Pinf.1 = 0 Then
Waitms 100
Pointer_ff_groesser = 1
'Debounce Pinf.1, 0, Ff_groesser, Sub
Goto Ff_groesser
End If

Return

/----- Menue obere Frequenz fein -----
    Ff_groesser:

    Pointer_ff_groesser = 1

    If Pointer_berechnung = 1 Then
Goto Berechnung
End If

    Locate 1, 1
    Lcd "FF " ; Zahl2 ; Blank5

    If Adc2 < 3800 Then
Locate 2, 1
    Lcd ">3800 ? " ; Adc2 ; Blank6
    Else
        Locate 2, 1
        Lcd ">3800 " ; Adc2 ; Blank6
    End If

    If Adc2 > 3800 And Pr2 = 0 Then

        Adc2gr = Adc2
        Ffgr = Zahl2
        Locate 2, 1
        Lcd "gespeichert" ; Blank4

        Waitms 1000
    End If

    If Pine.0 = 0 Then
Waitms 100
Pointer_ff_groesser = 0
Debounce Pine.0, 0, Ff_kleiner, Sub
End If
If Pinf.1 = 0 Then
Waitms 100
Pointer_berechnung = 1
'Debounce Pinf.1, 0, ff_kleiner, Sub
Goto Berechnung
End If

Return

```


----- Menue Berechnung ADC Konstanten -----

Berechnung:

Pointer_berechnung = 1

Fgdiff = Fggr - Fgkl
 Adc1diff = Adc1gr - Adc1kl
 Adc11_konst = Fgdiff / Adc1diff

Ffdiff = Ffgr - Ffkl
 Adc2diff = Adc2gr - Adc2kl
 Adc21_konst = Ffdiff / Adc2diff

If Pr2 = 0 Then
 E_calwert5 = Adc11_konst
 E_calwert6 = Adc21_konst

Locate 2 , 1
 Lcd "gespeichert" ; Blank4
 Waitms 1000
 Else

Locate 1 , 1
 Lcd "FG" ; Adc11_konst
 Locate 2 , 1
 Lcd "FF" ; Adc21_konst

End If

If Pine.0 = 0 Then
 Waitms 100
 Pointer_berechnung = 0

Debounce Pine.0 , 0 , Ff_groesser , Sub
 'Debounce Pinf.1 , 0 , Calwert_setzen , Sub
 End If

If Pinf.1 = 0 Then
 Waitms 100
 Pointer_calwert_setzen = 1

Goto Calwert_setzen

End If

Return

----- Subroutine zum Auslesen des MCP3204 -----

Mcp3204:

Spiinit	'SPI-Hard Ports festlegen
Waitms 10	
C(1) = 0	'Variable C löschen
C(2) = 0	
C(3) = 0	
Cs = 0	'chipselect active
Waitms 1	
Spiout A , 1	'sende Controlreg, 1 byte, Adresse
Spiin C(1) , 3	'lese 12 Bits in 2 byte
Cs = 1	'chipselect off

Werte_mcp = C(1)	'eingelesene Bits aufbereiten
Shift Werte_mcp , Left , 4	
Shift C(2) , Right , 4	
Werte_mcp = Werte_mcp + C(2)	

Return

```
'----- Potiabfrage kleiner Frequenz grob -----
Adc1_kleiner:
```

```
    If Gatetime1 = 0 Then
        Adc1_konst = Adc101
        ' Adc1_konst = 5.352025390625

    End If

    If Gatetime1 = 1 Then
        Adc1_konst = Adc102
        ' Adc1_konst = 53.52025390625

    End If

    If Gatetime1 = 2 Then
        Adc1_konst = Adc103
        ' Adc1_konst = 535.2025390625

    End If

    If Gatetime1 = 3 Then
        Adc1_konst = Adc104
        ' Adc1_konst = 5352.025390625

    End If
```

```
Adc1_mem2 = Adc1_alt - Adc1      'Frequenzänderung in ADC-Digits
Osc1_mem  = Adc1_mem2 * Adc1_konst 'Frequenzänderung in MHz
Osc1_single = Osc1_alt - Osc1_mem 'geschätzte Frequenz
Zahl2 = Osc1_single
```

```
Return
```

```
'----- Potiabfrage größer Frequenz grob -----
```

```
Adc1_groesser:
```

```
    If Gatetime1 = 0 Then
        Adc1_konst = Adc101
        ' Adc1_konst = 5.352025390625

    End If

    If Gatetime1 = 1 Then
        Adc1_konst = Adc102
        ' Adc1_konst = 53.52025390625

    End If

    If Gatetime1 = 2 Then
        Adc1_konst = Adc103
        ' Adc1_konst = 535.2025390625

    End If

    If Gatetime1 = 3 Then
        Adc1_konst = Adc104
        ' Adc1_konst = 5352.025390625

    End If
```

```

    Adc1_mem2 = Adc1 - Adc1_alt      'Frequenzänderung in ADC-Digits
    Osc1_mem  = Adc1_mem2 * Adc1_konst 'Frequenzänderung in MHz
    Osc1_single = Osc1_alt + Osc1_mem 'geschätzte Frequenz
    Zahl2 = Osc1_single

    Blank_fz = Fragezeichen
    Flag = 1
    Gosub Mhz1
    Gosub Anzeigen
    Waitms 50

'----- Potiabfrage kleiner Frequenz fein -----
Return
Adc2_kleiner:
    If Gatetime1 = 0 Then
        ' Adc2_konst = .003148760316
        Adc2_konst = Adc201

        End If

        If Gatetime1 = 1 Then
            Adc2_konst = Adc202
            ' Adc2_konst = 0.03148760316

            End If

        If Gatetime1 = 2 Then
            Adc2_konst = Adc203
            ' Adc2_konst = 0.3148760316

            End If

            If Gatetime1 = 3 Then
                Adc2_konst = Adc204
                ' Adc2_konst = 3.148760316

                End If

            Adc2_mem2 = Adc2_alt - Adc2      'Frequenzänderung in ADC-Digits
            Osc3_mem  = Adc2_mem2 * Adc2_konst 'Frequenzänderung in KHz
            Osc3_single = Osc3_alt - Osc3_mem 'geschätzte Frequenz
            Zahl2 = Osc3_single              '+ Osc1_single

            Blank_fz = Fragezeichen
            Flag = 1
            Gosub Mhz1
            Gosub Anzeigen
            Waitms 50

Return

```

'-----Potiabfrage größer Frequenz fein -----'

Adc2_groesser:

```

    If Gatetime1 = 0 Then
        ' Adc2_konst = 0.003148760316
        Adc2_konst = Adc201
    End If

    If Gatetime1 = 1 Then
        Adc2_konst = Adc202
        ' Adc2_konst = 0.03148760316
    End If

    If Gatetime1 = 2 Then
        Adc2_konst = Adc203
        ' Adc2_konst = 0.3148760316
    End If

    If Gatetime1 = 3 Then
        Adc2_konst = Adc204
        ' Adc2_konst = 3.148760316
    End If

    Adc2_mem2 = Adc2 - Adc2_alt      'Frequenzänderung in ADC-Digits
    Osc3_mem = Adc2_mem2 * Adc2_konst 'Frequenzänderung in KHz
    Osc3_single = Osc3_alt + Osc3_mem 'geschätzte Frequenz
    Zahl2 = Osc3_single

    Blank_fz = Fragezeichen
    Flag = 1
    Gosub Mhz1
    Gosub Anzeigen
    Waitms 50
Return
End

```