

```

MÄr 05, 06 15:20          mc_v1_12.pic          Seite 1/64

list          p=16f877          ; list directive to define processor
#include      <p16f877.inc>      ; processor specific variable definitions
#include      <makros.inc>

#define       europa            ;Europa = Testfrequenz 6,075 Mhz USA = 10 Mhz
;#define     debug
;#define     test

_CONFIG      _CP_OFF & _LVP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC

LCDStr: MACRO var1
movlw       HIGH var1          ;Highanteil der Adresse laden
movwf      lcdtextadrh        ;und in den Ram fuer alle Baenke laden
movlw      LOW var1           ;die untersten 8 Bit in W laden
call       LCDString          ;und String ausgeben
ENDM

;=====
; Variablen:
;=====
          org      70h          ;Bankunabhaengiger Bereich

w_temp:   res      1           ;70h variable used for context saving
status_temp: res 1           ;71h variable used for context saving
pclath_temp: res 1           ;72h
fsr_temp:  res 1           ;73h
lcdtextadrh: res 1          ;74h Bankunabhaengiger Bereich
option_mirror: res 1         ;75h
LCDByte:   res 1           ;76h Zwischenspeicher fuer LCD-Ausgabe
LCDByte2:  res 1           ;77h Zwischenspeicher fuer Stringausgabe
data_ee_addr: res 1         ;78h Zwischenspeicher interne Eepromadr
data_ee_data: res 1         ;79h Zwischenspeicher interne Eepromdaten
utemp      res 4           ;7ah,7bh,7ch,7dh

          org      020h

bank0:
ramanfng:

;Variable die fast immer gebraucht werden
;die ersten Ramzellen werden vom Eeprom ueberladen
flag1     res      1
ddskonst  res      4
zwischenfrequenz res 4
band1     res      1
band2     res      1
band3     res      1
band4     res      1
band5     res      1
rxdelay   res      1          ;Verzoeigerung TX -> RX in mSek

flag2     res      1
flag3     res      1
flag4     res      1
frequenza res 4           ;Frequenz VFO-A
frequenzb res 4           ;Frequenz VFO-B
band      res      1          ;Merker fuer das Band
pointer1  res      1          ;Zeiger fuer indirekt
pointer2  res      1          ;Zeiger fuer indirekt
delay_counter1: res 1       ;Zellen fuer Delay
delay_counter2: res 1
delay_counter3: res 1
schleife  res      3
tastennummer res 1
ebene     res      1          ;Menuebene
step      res      1          ;Schrittweite
impulse:  res      1          ;Drehimpulse
temp:     res      10         ;temporaere Speicher
tempkonst1: res 8          ;DDS rechenspeicher
tempkonst2: res 8          ;nicht trennen da gemeinsam clr

zs1       res      1          ;Zeit fuer Menuanzeige
zs2       res      1          ;Zeitschleife fuer AGC Haengeanzeige
zs3       res      1          ;Punktlaenge
zs4       res      1          ;Zeitschleife fuer AGC nach Haengezeit
idlecounter res 1          ;Zaehler fuer Leerdurchlaeufer ohne Funktion
uvor      res      1
urueck    res      1

```

```

MÄr 05, 06 15:20          mc_v1_12.pic          Seite 2/64

eeindex     res      1

ramende:    res      1          ;Merker fuer Ramende

          org      0A0h

bank1:
swr         res      3
ukeyer      res      4
keyergeschw res 2
ddsword     res      2
ddsdword    res      4
ddsbintx:   res      4          ;binaerwert fuer DDS laden Empfangsfrequenz
ddsbintx:   res      4          ;binaerwert fuer DDS laden Sendefrequenz
anst        res      1          ;Anstiegstangente AGC beim scannen
agc         res      2          ;AGC Zwischenspeicher beim scannen

wert:
wert1       res      1          ;Messwert AGC (Word)
werth       res      1
s_konst1    res      2          ;AGC-Kurve x-wert (lineare Funktion)
s_konst2    res      2          ;AGC-Kurve y-wert (lineare Funktion)
fscan       res      4
tempbank1   res      12
tonhoehe    res      1
agcmerk     res      3
ubattmerk   res      2

          org      120h

bank2:

          org      1A0h

bank3:

;Flagdefinitionen
#define blightaut    flag1,0    ;Merkerbit Lichtautomatik
#define bkeyer      flag1,1    ;
#define blight      flag1,2    ;Merkerbit fuer Dauerlicht
#define brit        flag1,3    ;1=Rit ein
#define bddsneu     flag1,4    ;DDS neu errechnen und laden
#define blcdneu     flag1,5    ;LCD neu anzeigen
#define bnull       flag1,6    ;Fuehrende Null unterdruecken
#define bmenu       flag1,7    ;Merkerbit fuer Menu
#define bidle       flag2,0    ;Merkerbit fuer keine Funktion
#define bpunkt      flag2,1    ;Merkerbit Punkt
#define bstrich     flag2,2    ;Merkerbit Strich
#define bzslausein  flag2,3    ;Merkerbit Zeitl gestartet
#define bzslausaus  flag2,4    ;Merkerbit Zeitl abgelaufen
#define bsenderein  flag2,5    ;Merkerbit fuer Sender ein
#define bkeyeranz   flag2,6    ;Merkerbit
#define tonenable   flag2,7    ;Mithoerton
#define bzfadd      flag3,0    ;ZF addieren oder subtrahieren
#define LCDr        flag3,1    ;Statusspeicher fuer LCD beim Busylesen
#define bgen        flag3,2    ;Merkerbit radioempfang
#define bscan       flag3,3    ;Merkerbit scannen
#define bddsl0mhz   flag3,4    ;Merkerbit fuer 10Mhz ausgabe
#define bzfdirekt   flag3,5    ;Merkerbit ZF direkt ausgeben
#define bddsohnezf  flag3,6    ;Merkerbit DDS ohne ZF Berechnung
#define bxit        flag3,7    ;Merkerbit XIT
#define bvfo        flag4,0    ;0=VFO_A 1=VFO_B

;=====
;Namen der I/O Leitungen
;=====

;port a
#define a_mton      PORTA,4h    ;

;port b
#define drehtakt    PORTB,0h    ;
#define drehdir     PORTB,1h    ;
#define punktpin    PORTB,2h    ;
#define strichpin   PORTB,3h    ;
#define taste       PORTB,4h    ;
#define tastel     PORTB,5h    ;
#define taste2      PORTB,6h    ;
#define taste3      PORTB,7h    ;

;port c
LCDPort     equ      PORTC          ;Port fuer LCDAusgabe
#define a_LCD_E     PORTC,0h      ;Enable
#define a_LCD_RW    PORTC,1h      ;R/W Pin

```

```

MÄr 05, 06 15:20          mc_v1_12.pic          Seite 3/64

#define a_LCD_RS           PORTC,2h           ;RS Pin an LCD
#define senderein         PORTC,3h           ;Sender tasten
#define ea_LCD_D4         PORTC,4h           ;Daten von LCD
#define ea_LCD_D5         PORTC,5h           ;Daten von LCD
#define ea_LCD_D6         PORTC,6h           ;Daten von LCD
#define ea_LCD_D7         PORTC,7h           ;Daten von LCD

;port d
#define relgem            PORTD,0h           ;
#define rel1              PORTD,1h           ;
#define rel2              PORTD,2h           ;
#define light             PORTD,3h           ;
#define tastungein      PORTD,4h           ;
#define ukap1             PORTD,5h           ;
#define ukap2             PORTD,6h           ;
#define ukap3             PORTD,7h           ;

;port e
#define ddsdaten          PORTE,0h           ;
#define ddsclk            PORTE,1h           ;
#define ddsfsyn           PORTE,2h           ;
;=====
;Konstanten fuer die PIC-Konfiguration
;=====
d_option_reg:            equ      b'01000001'

;
; bit 7: RBPU: PORTB Pull-up Enable bit
;   1 = PORTB pull-ups are disabled
;   0 = PORTB pull-ups are enabled by individual port latch values
;
; bit 6: INTEDG: Interrupt Edge Select bit
;   1 = Interrupt on rising edge of RB0/INT pin
;   0 = Interrupt on falling edge of RB0/INT pin
;
; bit 5: TOCS: TMR0 Clock Source Select bit
;   1 = Transition on RA4/TOCKI pin
;   0 = Internal instruction cycle clock (CLKOUT)
;
; bit 4: TOSE: TMR0 Source Edge Select bit
;   1 = Increment on high-to-low transition on RA4/TOCKI pin
;   0 = Increment on low-to-high transition on RA4/TOCKI pin
;
; bit 3: PSA: Prescaler Assignment bit
;   1 = Prescaler is assigned to the WDT
;   0 = Prescaler is assigned to the Timer0 module
;
; bit 2-0: PS<2:0>: Prescaler Rate Select bits

d_port_a:                equ      00h
d_port_b:                equ      00h
d_port_c:                equ      00h
d_port_d:                equ      00h
d_port_e:                equ      00h
d_tmr0:                  equ      b'00000000'

;0=Ausgang
;1=Eingang
;Funktion siehe Definitionen

d_trisa:                 equ      b'00101111'
d_trisb:                 equ      b'11111111'
d_trisc:                 equ      b'00000000'
d_trisd:                 equ      b'00000000'
d_trise:                 equ      b'00000000'
d_LCD_lesen:            equ      b'10000000' ;zum Lesen aus LCD
d_LCD_schreiben:        equ      b'00000000'

d_intcon:                equ      b'11110000' ;global interrupt enable
;bit 7: GIE: Global Interrupt Enable bit
;   1 = Enables all un-masked interrupts
;   0 = Disables all interrupts
;
;bit 6: PEIE: Peripheral Interrupt Enable bit
;   1 = Enables all un-masked peripheral interrupts
;   0 = Disables all peripheral interrupts
;
;bit 5: TOIE: TMR0 Overflow Interrupt Enable bit
;   1 = Enables the TMR0 interrupt
;   0 = Disables the TMR0 interrupt
;
;bit 4: INTE: RB0/INT External Interrupt Enable bit
;   1 = Enables the RB0/INT external interrupt
;   0 = Disables the RB0/INT external interrupt
;
;bit 3: RBIE: RB Port Change Interrupt Enable bit
;   1 = Enables the RB port change interrupt
;   0 = Disables the RB port change interrupt
;
;bit 2: TOIF: TMR0 Overflow Interrupt Flag bit
;   1 = TMR0 register has overflowed (must be cleared in software)
;   0 = TMR0 register did not overflow
;
;bit 1: INTF: RB0/INT External Interrupt Flag bit

```

```

MÄr 05, 06 15:20          mc_v1_12.pic          Seite 4/64

;
;   1 = The RB0/INT external interrupt occurred (must be cleared in software)
;   0 = The RB0/INT external interrupt did not occur
;bit 0: RBIF: RB Port Change Interrupt Flag bit
;   1 = At least one of the RB<7:4> pins changed state (must be cleared in software)
;   0 = None of the RB<7:4> pins have changed state

d_piel:                  equ      b'00000011' ;

;bit 7: Reserved: Always maintain this bit clear
;bit 6: ADIE: A/D Converter Interrupt Enable bit
;   1 = Enables the A/D converter interrupt
;   0 = Disables the A/D converter interrupt
;bit 5-4: Reserved: Always maintain this bit clear
;bit 3: SSPIE: Synchronous Serial Port Interrupt Enable bit
;   1 = Enables the SSP interrupt
;   0 = Disables the SSP interrupt
;bit 2: CCP1IE: CCP1 Interrupt Enable bit
;   1 = Enables the CCP1 interrupt
;   0 = Disables the CCP1 interrupt
;bit 1: TMR2IE: TMR2 to PR2 Match Interrupt Enable bit
;   1 = Enables the TMR2 to PR2 match interrupt
;   0 = Disables the TMR2 to PR2 match interrupt
;bit 0: TMR1IE: TMR1 Overflow Interrupt Enable bit
;   1 = Enables the TMR1 overflow interrupt
;   0 = Disables the TMR1 overflow interrupt

d_tlcon:                 equ      b'00000001' ;

;bit 7-6: Unimplemented: Read as '0'
;bit 5-4: T1CKPS<1:0>: Timer1 Input Clock Prescale Select bits
;   11 = 1:8 Prescale value
;   10 = 1:4 Prescale value
;   01 = 1:2 Prescale value
;   00 = 1:1 Prescale value
;bit 3: T1OSCEN: Timer1 Oscillator Enable Control bit
;   1 = Oscillator is enabled
;   0 = Oscillator is shut off (The oscillator inverter is turned off to eliminate power drain)
;bit 2: T1SYNC: Timer1 External Clock Input Synchronization Control bit
;   TMR1CS = 1
;   1 = Do not synchronize external clock input
;   0 = Synchronize external clock input
;   TMR1CS = 0
;   This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.
;bit 1: TMR1CS: Timer1 Clock Source Select bit
;   1 = External clock from pin RC0/T1OSO/T1CKI (on the rising edge)
;   0 = Internal clock (FOSC/4)
;bit 0: TMR1ON: Timer1 On bit
;   1 = Enables Timer1
;   0 = Stops Timer1

d_t2con:                 equ      b'00000001' ;

;bit 7: Unimplemented: Read as '0'
;bit 6-3: TOUTPS3:TOUTPS0: Timer2 Output Postscale Select bits
;   0000 = 1:1 Postscale
;   0001 = 1:2 Postscale
;   0010 = 1:3 Postscale
;   1111 = 1:16 Postscale
;bit 2: TMR2ON: Timer2 On bit
;   1 = Timer2 is on
;   0 = Timer2 is off
;bit 1-0: T2CKPS1:T2CKPS0: Timer2 Clock Prescale Select bits
;   00 = Prescaler is 1
;   01 = Prescaler is 4
;   1x = Prescaler is 16

tmr1hconst:              equ      0d9H
tmr1lconst:              equ      0h

tmr1word                 equ      .9984

tmr1hconst:              equ      HIGH (0 - tmr1word)
tmr1lconst:              equ      LOW (0 - tmr1word)

tmr2const:               equ      0H

d_pie2:                  equ      b'00000000' ;

;bit 7: Unimplemented: Read as '0'
;bit 6: Reserved: Always maintain this bit clear

```

```

MÄr 05, 06 15:20          mc_v1_12.pic          Seite 5/64

;bit 5: Unimplemented: Read as '0'
;bit 4: EEIE: EEPROM Write Operation Interrupt Enable
;      1 = Enable EE Write Interrupt
;      0 = Disable EE Write Interrupt
;bit 3: BCLIE: Bus Collision Interrupt Enable
;      1 = Enable Bus Collision Interrupt
;      0 = Disable Bus Collision Interrupt
;bit 2-1: Unimplemented: Read as '0'
;bit 0: Reserved: Always maintain this bit clear

d_adcon1:          equ      b'00000010'          ;

;bit 7 ADFM: A/D Result Format Select bit
;      1 = Right justified. Most Significant bits of ADRESH are read as 0
;      0 = Left justified. 6 Least Significant bits of ADRESL are read as 0 i.
;bit 6-4 Unimplemented: Read as '0'
;bit 3-0 PCFG3:PCFG0: A/D Port Configuration Control bits:
;PCFG0 AN7(1)RE2      AN6(1)RE1      AN5(1)RE0      AN4RA5      AN3RA3      AN2RA2      AN1RA1      AN0RA0
;0000 A              A              A              A              A              A              A              A
;0001 A              A              VREF+         A              A              A              A
;0010 D              D              D              A              A              A              A
;0011 D              D              D              A              VREF+         A              A
;0100 D              D              D              D              A              D              A
;0101 D              D              D              D              VREF+         D              A
;011x D              D              D              D              D              D              D
;1000 A              A              A              A              VREF+         VREF-         A
;1001 D              D              D              A              A              A              A
;1010 D              D              D              A              VREF+         A              A
;1011 D              D              D              A              VREF+         VREF-         A
;1100 D              D              D              A              VREF+         VREF-         A
;1101 D              D              D              D              VREF+         VREF-         A
;1110 D              D              D              D              D              D              A
;1111 D              D              D              D              VREF+         VREF-         A

;bit 7 ADFM: A/D Result Format Select bit
;      1 = Right justified. 6 Most Significant bits of ADRESH are read as 0.
;      0 = Left justified. 6 Least Significant bits of ADRESL are read as 0.
;bit 6-4 Unimplemented: Read as '0'
;bit 3-0 PCFG3:PCFG0: A/D Port Configuration Control bits:

err1:              equ      0h
fnull:             equ      1h
tonw:              equ      2h
nodisp:            equ      3h
di:                equ      7          ; input
do:                equ      6          ; output

speicher1:         equ      20h        ;Speicherplatz 1
speicher2:         equ      80h        ;Speicherplatz 2
sp_lang:           equ      5ah        ;maximale Anzahl der Zeichen in den Speichern

x0vvvein           equ      .0
x0vvvaus           equ      .4
;=====
; EEPROM Zellen
;=====

          org 2100h
eflag1             de      b'00000011'      ;flag1 light=off,lightauto=on,keyer=on
edds               de      0b8h,063h,05eh,05h ;DDS konstante
ezf                de      38h,0ffh,4ah,0    ;ZF binär (4,915 MHz)
eband1             de      01h              ;welches Band auf Speicherstelle 1
eband2             de      02h              ;welches Band auf Speicherstelle 2
eband3             de      03h              ;welches Band auf Speicherstelle 3
eband4             de      04h              ;welches Band auf Speicherstelle 4
eband5             de      05h              ;welches Band auf Speicherstelle 4
erxdelay           de      .5              ;default 5 mSek Verzoegerung TX -> RX
epitch            de      3dh              ;Tonhoehe

e80m:              de      0,0,56h,03h
                   de      0,0,56h,03h
e40m:              de      0,0,03h,07h
                   de      0,0,03h,07h
e30m:              de      0,60h,11h,10h
                   de      0,60h,11h,10h
e20m:              de      0,0,06h,14h
                   de      0,0,06h,14h
e17m:              de      0,60h,09h,18h

```

```

MÄr 05, 06 15:20          mc_v1_12.pic          Seite 6/64

egen:              de      0,60h,09h,18h
                   de      0,50h,07h,06h
                   de      0,50h,08h,06h
es_konst1:         de      25h,0           ;S-Meter konstante 1
es_konst2:         de      95h,0           ;S_meter konstante 2

;definitionen fuer Speicherplaetze 1 bis 20
;jedes Byte ausnutzen durch Adressrechnung

elaengesplatz:    equ      .9              ;Laenge eines Speicherplatzes
emem:             equ      21ffh-(.20 * elaengesplatz);Beginn Speicher
;=====
; Hardware initialisieren
;=====

          org 0
PAGE0:
start:
          goto      haupt
;=====
;ISR
;=====

          org 4
isr:
          movwf    w_temp          ;W retten
          swapf   STATUS,W         ;Flags nach W holen
          clrf    STATUS          ;Bank 0 einschalten
          movwf   status_temp      ;Flags retten
          movf    PCLATH,W         ;
          movwf   pclath_temp      ;PCLATH retten
          clrf    PCLATH          ;
          bcf     STATUS, IRP      ;
          movf    FSR,W           ;
          movwf   fsr_temp        ;

          BANKSEL bank0
          btfss   INTCON,INTF      ;RB0 Interrupt durch PICbus
          goto   ir1
          BANKSEL OPTION_REG      ;
          movf    OPTION_REG,W     ;
          BANKSEL PORTA          ;
          movwf   option_mirror    ;
          btfss   option_mirror,6 ;
          goto   ir20
          btfsc   drehdir         ;Drehrichtung
          goto   ir2
          decf    impulse         ;
          goto   ir21
ir2:
          incf    impulse         ;
          goto   ir21
ir20:
          btfss   drehdir         ;Drehrichtung
          goto   ir202
          decf    impulse         ;
          goto   ir21
ir202:
          incf    impulse         ;
ir21:
          btfss   option_mirror,6 ;
          goto   ir22
          BANKSEL OPTION_REG      ;
          bcf     OPTION_REG,6    ;
          BANKSEL PORTA          ;
          goto   ir23
ir22:
          BANKSEL OPTION_REG      ;
          bsf     OPTION_REG,6    ;
          BANKSEL PORTA          ;
ir23:
          bcf     INTCON,INTF      ;Bit wieder bereit
ir1:
          btfss   PIR1,TMR1IF     ;? Timer 1 10mSek vorbei
          goto   ir3
          bcf     PIR1,TMR1IF     ;1 Timer 1 Bit loeschen
          movf    zsl,F           ; Zeitschleife 1 testen
          btfss   STATUS,Z         ; ? Test ob 0
          decf    zsl,F           ; 0 um 1 decrementieren
          movf    zs2,F           ; Zeitschleife 2 testen

```

```

MÄr 05, 06 15:20          mc_v1_12.pic          Seite 7/64
SKPZ                      ; ? Test ob 0
decf    zs2,F              ; 0 1 dec
movf    zs4,F              ; Zeitschleife 4 testen
SKPZ                      ; ? Test ob 0
decf    zs4,F              ; 0 1 dec

movlw   tmrlhconst         ; timer neu laden
movwf   TMR1H              ;
movlw   tmrllconst        ;
movwf   TMR1L              ;
ir3:    btfss  PIR1,TMR2IF  ;Timer 2 ist fuer Punktzeit
        goto   ir4         ;
        bcf    PIR1,TMR2IF  ;Timer 2 Bit loeschen
        movf   zs3,F        ;
        btfss  STATUS,Z     ;
        decf   zs3,F        ;

ir4:    btfss  INTCON,T0IF  ;
        goto   ir6         ;
        bcf    INTCON,T0IF  ;Timer0 Bit loeschen
        BANKSEL tonhoehe   ;
        movf   tonhoehe,W  ;
        andlw  b'01111111' ;
        BANKSEL bank0     ;
        movwf  TMR0        ;
        btfss  tonenable   ;? Ton eingeschaltet
        goto   ir6         ;
        btfsc  a_mton       ;1 Ausgangsbit wechseln
        goto   ir5         ;
        bsf    a_mton       ;
        goto   ir6         ;

ir5:    bcf    a_mton       ;0 Tonbit bei aus auf Low

ir6:    btfsc  PIR2,EEIF    ;
        bcf    PIR2,EEIF    ;

;Restaurieren der Register
iend:   movf   fsr_temp,W   ;FSR restaurieren
        movwf  FSR         ;
        movf   pclath_temp,W ;
        movwf  PCLATH      ;PCLATH restaurieren
        swapf  status_temp,w ;gerettete Flags in W
        movwf  STATUS      ;Flags restaurieren
        swapf  w_temp,f    ;W restaurieren
        swapf  w_temp,w    ;W restaurieren
        retfie

;=====
;Funktion:   Wandlung BCDZahl unteres Nibbel in ASCII-Zeichen
;Eingang:   BCD in W unteres Nibbel
;Ausgang:   entsprechendes Ascii-Zeichen in W
;=====
BCDToASCII:
        andlw  B'00001111' ;sicherheitshalber nur unteres Nibbel
        addwf  PCL         ;zum Programmcounter addieren
        retlw  '0'         ;und mit entsprechenden Ascii zurueck
        retlw  '1'
        retlw  '2'
        retlw  '3'
        retlw  '4'
        retlw  '5'
        retlw  '6'
        retlw  '7'
        retlw  '8'
        retlw  '9'
        retlw  'A'
        retlw  'B'
        retlw  'C'
        retlw  'D'
        retlw  'E'
        retlw  'F'

;=====
u2bpm:   andlw  B'00001111' ;sicherheitshalber nur unteres Nibbel
        addwf  PCL         ;zum Programmcounter addieren
        retlw  .190        ;
        retlw  .180        ;
        retlw  .170        ;

```

```

MÄr 05, 06 15:20          mc_v1_12.pic          Seite 8/64
        retlw  .160        ;
        retlw  .150        ;
        retlw  .140        ;
        retlw  .130        ;
        retlw  .120        ;
        retlw  .110        ;
        retlw  .100        ;
        retlw  .90         ;
        retlw  .80         ;
        retlw  .70         ;
        retlw  .60         ;
        retlw  .50         ;
        retlw  .40         ;

;=====
;Funktion   Keyergeschwindigkeit 9-40 BpM in Werte umsetzen die die
;           Zeitschleife entsprechend steuern
;Eingang   wertebereich 0-FFh vom DA wandler
;Ausgang   Zahl der Schleifenwerte multipliziert mit lmsek
;
ukey2msek:
        BANKSEL bank1
        movf   keyergeschw+1,W
        BANKSEL bank0
        movwf  schleife
        rrf    schleife,F
        rrf    schleife,F
        rrf    schleife,W
        andlw  B'00011111'
        addwf  PCL
        retlw  .133        ;9
        retlw  .120        ;10
        retlw  .109        ;11 BpM
        retlw  .100        ;12 BpM
        retlw  .92         ;13 BpM
        retlw  .86         ;14 BpM
        retlw  .80         ;15 BpM
        retlw  .75         ;16 BpM
        retlw  .71         ;17 BpM
        retlw  .67         ;18 BpM
        retlw  .63         ;19 BpM
        retlw  .60         ;20 BpM
        retlw  .57         ;21 BpM
        retlw  .55         ;22 BpM
        retlw  .52         ;23 BpM
        retlw  .50         ;24 BpM
        retlw  .48         ;25 BpM
        retlw  .46         ;26 BpM
        retlw  .44         ;27 BpM
        retlw  .43         ;28 BpM
        retlw  .41         ;29 BpM
        retlw  .40         ;30 BpM
        retlw  .39         ;31 BpM
        retlw  .38         ;32 BpM
        retlw  .36         ;33 BpM
        retlw  .35         ;34 BpM
        retlw  .34         ;35 BpM
        retlw  .33         ;36 BpM
        retlw  .32         ;37 BpM
        retlw  .32         ;38 BpM
        retlw  .31         ;39 BpM
        retlw  .30         ;40 BpM
        retlw  .30         ;40 BpM

;=====
;Funktion   Keyergeschwindigkeit 9-40 BpM in Werte umsetzen die die
;           Zeitschleife entsprechend steuern
;Eingang   wertebereich 0-FFh vom DA wandler
;Ausgang   HEX Ziffer zur Anzeige
;
ukey2anz:
        movwf  schleife
        rrf    schleife,F
        rrf    schleife,F
        rrf    schleife,W
        andlw  B'00011111'
        addwf  PCL
        retlw  9h
        retlw  10h
        retlw  11h

```

MÄr 05, 06 15:20

mc_v1_12.pic

Seite 9/64

```

retlw 12h
retlw 13h
retlw 14h
retlw 15h
retlw 16h
retlw 17h
retlw 18h
retlw 19h
retlw 20h
retlw 21h
retlw 22h
retlw 23h
retlw 24h
retlw 25h
retlw 26h
retlw 27h
retlw 28h
retlw 29h
retlw 30h
retlw 31h
retlw 32h
retlw 33h
retlw 34h
retlw 35h
retlw 36h
retlw 37h
retlw 38h
retlw 39h
retlw 40h
;=====
MenuStart:
  call quittungston
  clrf impulse ;Impulse loeschen
  movlw 1 ;Tastennummer 1 voreinstellen
  movwf tastennummer ;
  movlw .10 ;Zeitschleife aufbauen 500 mSek
  movwf schleife ;
MenuStart1:
  call Tastegedruackt ;SCHLEIFE(1)
  SKPC ;
  goto MenuStart2 ; break --> Taste nicht gedruackt Tastennummer 1
  call t50mSek ; Display unveraendert
  decfsz schleife,F ;ENDE(1) nach 500 msek Taste gedruackt
  goto MenuStart1 ;
  call quittungston ;
  call t50mSek ;
  call quittungston ;
  call LCDDisplayClear ;Display loeschen
  LCDStr text5 ;"Menu" im Display
  movlw 2 ;nach 500 mSek Tastennummer 2
  movwf tastennummer ;
MenuStart3:
  call Tastegedruackt ;SCHLEIFE(2)
  SKPNC ;
  goto MenuStart3 ;ENDE(2) keine Taste gedruackt
MenuStart2:
  return
;=====
;Abfrage ob Taste gedruackt ist
;Ausgang C=1 Taste gedruackt
; C=0 Taste nicht gedruackt
;
Tastegedruackt:
  bsf STATUS,C
  btfsc taste
  bcf STATUS,C
  return
;=====
;Initialisierung des µC
;=====
init:
  clrf STATUS ; Flags loeschen

  BANKSEL TRISA ; Registerbank 80h..0AFh
  movlw d_option_reg
  movwf OPTION_REG
  movlw d_trisa
  movwf TRISA
  movlw d_trisb
  movwf TRISB
  movlw d_trisc

```

MÄr 05, 06 15:20

mc_v1_12.pic

Seite 10/64

```

movwf TRISC
movlw d_trisd
movwf TRISD
movlw d_trise
movwf TRISE
movlw d_adcon1
movwf ADCON1
movlw d_piel
movwf PIE1
movlw d_pie2
movwf PIE2
BANKSEL PORTA ; Registerbank 0..2FH

movlw d_intcon
movwf INTCON
movlw d_port_a
movwf PORTA
movlw d_port_b
movwf PORTB
movlw d_port_c
movwf PORTC
movlw d_port_d
movwf PORTD
movlw d_port_e
movwf PORTE
movlw d_tlcon
movwf T1CON
movlw d_t2con
movwf T2CON

movlw ramanfang ;Ram loeschen
movwf FSR ;fsr fuer indirekte Adressierung laden
ramclr1:
  clrf INDF ;Fileregister indirekt loeschen
  movlw ramende ;Ramende in Akku
  subwf FSR,W ;Test ob Ende erreicht
  btfsc STATUS,Z ;Auswertung des Zeroflags
  goto anfl1 ;Ende erreicht
  incf FSR,F ;Ende nicht erreicht
  goto ramclr1 ;wieder von vorn
anfl1:
  movlw tmr1hconst ;Timer1 Zeit laden
  movwf TMR1H
  movlw tmr1lconst
  movwf TMR1L
  BANKSEL bank1
  movlw 1
  movwf agcmerk+2 ;Anzeigewert auf FFh setzen
  movwf ubattmerk+1 ;
  clrf agcmerk ;
  clrf agcmerk+1 ;
  BANKSEL bank0
  return
;=====
;Zeitschleife
;1 Einheit= 1 mSek
;Einsprung bei DELAY wird mit Wert in W gerechnet
;=====
t1Sek:
  movlw .250
  call DELAY
  movlw .250
  call DELAY
  movlw .250
  call DELAY
  movlw .250
  call DELAY
t250mSek:
  movlw .250
  goto DELAY
t100mSek:
  movlw .100
  goto DELAY
t50mSek:
  movlw .50
  goto DELAY
t10mSek:
  movlw .10
  goto DELAY
t1mSek:
  movlw .1
  goto DELAY
DELAY:
  movlw .1

```

```

MÄr 05, 06 15:20          mc_v1_12.pic          Seite 11/64
;=====  

ifdef  debug              ;Zeitschleife verkuerzen  
return  ;beim Simulieren  
endif  
  
movwf  delay_counter1  
incf  delay_counter1,F    ;min 1 auswerten  
DELAY_1:  
DECFSZ delay_counter1,F  ; dec F + result in F  
GOTO  DELAY_2            ; if F=0: skip next instr.  
return  
  
DELAY_2:  
MOVLW .19  
MOVWF delay_counter2  
  
DELAY_3:  
DECFSZ delay_counter2,F  
GOTO  DELAY_4  
  
DELAY_4:  
GOTO  DELAY_1  
  
DELAY_5:  
MOVLW .16  
MOVWF delay_counter3  
  
DELAY_5:  
DECFSZ delay_counter3,F  
GOTO  DELAY_5  
GOTO  DELAY_3  
  
;=====  
;Initialisierung der LCD Anzeige  
;  
;=====  
;Function Set: 4 bit Datenbreite; 2 Zeilen  
ib1  equ  B'00100000'    ;Function set 1. nibel 4.Bitmodus  
ib2  equ  B'10000000'    ;Function set 2. nibel  
;=====  
;Entry Mode Set: increment, display shift  
; d7 d6 d5 d4 d3 d2 d1 d0  
; 0 0 0 0 0 1 I/D S  
; I/D 1= increment      0= decrement  
; S 1= display shift   0= display freeze  
ib3  equ  B'00000110'    ;Bit1=I/D, Bit0=S  
;=====  
;Display on/off control: display on, cursor off , cursor not blink  
; d7 d6 d5 d4 d3 d2 d1 d0  
; 0 0 0 0 1 D C B  
; D 1= display on      0= display off  
; C 1= cursor on       0= cursor off  
; B 1= cursor blink    0= cursor not blink  
ib4  equ  B'00001100'    ;Bit2=D, Bit1=C, Bit0=B  
;=====  
;Cursor Display shift: display shift, right shift  
; d7 d6 d5 d4 d3 d2 d1 d0  
; 0 0 0 1 S/C F/L * *  
; S/C 1= display shift  0= cursor move  
; F/L 1= right shift    0= left shift  
ib5  equ  B'00010100'    ;Bit3=S/C, Bit2=R/L  
;=====  
LCDInit:  
bcf   a_LCD_E            ;Enable auf LOW setzen  
  
movlw .15                ;15 mSek warten bevor 1. Byte geladen wird  
call  DELAY  
  
movlw ib1                ;Funktion setzen  
call  LCDAusgabe  
movlw ib2  
call  LCDAusgabe  
  
movlw .5                 ;5 mSek warten  
call  DELAY  
  
movlw ib1                ;Funktion setzen  
call  LCDAusgabe  
movlw ib2  
call  LCDAusgabe  
  
movlw .1                 ;1 mSek warten  
call  DELAY  
  
movlw ib3                ;LCD entsprechend einstellen  
call  LCDCom

```

```

MÄr 05, 06 15:20          mc_v1_12.pic          Seite 12/64
movlw  ib4  
call   LCDCom  
movlw  ib5  
call   LCDCom  
  
call   LCDDisplayClear ;1 LCD wieder loeschen  
return  
  
;=====  
;Ausgabe eines Char auf dem Display  
;RS  = 1  
;R/W = 0  
  
LCDChar:  
movwf  LCDByte          ;Byte merken  
call   LCDRdy           ;ob LCD bereit  
movf   LCDByte,w        ;  
andlw  B'11110000'     ;oberes Nibbel verwenden  
iorlw  B'00000100'     ;RS hinzuschalten  
call   LCDAusgabe      ;zur LCD schicken  
swapf  LCDByte,w       ;unteres Nibbel laden vom gemerkten Byte  
andlw  B'11110000'     ;und zur LCD-Ausgabe vorbereiten  
iorlw  B'00000100'     ;RS hinzuschalten  
goto   LCDAusgabe      ;call einsparen und an LCD ausgeben  
  
;-----  
;Ausgabe eines LCD-Commandos zur Steuerung der LCD  
;RS  = 0  
;R/W = 0  
  
LCDCom:  
movwf  LCDByte          ;Byte merken  
call   LCDRdy           ;ist LCD bereit  
movf   LCDByte,w        ;zuerst oberes Nibbel  
andlw  B'11110000'     ;  
call   LCDAusgabe      ;ausgeben  
swapf  LCDByte,w       ;und dann unteres Nibbel  
andlw  B'11110000'     ;ausgeben  
  
;-----  
LCDAusgabe:  
btfsc  LCDPort,3       ;? test ob Bit gesetzt  
iorlw  B'00001000'     ;1 Bit nicht veraendern  
movwf  LCDPort         ;an PORT anlegen  
  
;-----  
Enable:  
bsf    a_LCD_E         ;LH Flanke uebernimmt LCD die 4 Bit  
bcf    a_LCD_E  
return  
  
;-----  
;lesen des Statusbits der LCD-anzeige  
;BS  = 0  
;R/W = 1  
  
LCDRdy:  
ifdef  debug          ;Zeitschleife verkuerzen  
return ;beim Simulieren  
endif  
  
bsf    LCDPort,7  
BANKSEL TRISC  
movlw  d_LCD_lesen    ;Port vorbereiten zum Lesen  
movwf  TRISC  
BANKSEL PORTC  
  
bcf    a_LCD_RS  
bsf    a_LCD_RW  
  
LCDRdy1:  
bcf    LCDr  
bsf    a_LCD_E  
nop  
btfss  LCDPort,7  
bsf    LCDr  
bcf    a_LCD_E  
nop  
bsf    a_LCD_E  
nop  
bcf    a_LCD_E  
btfss  LCDr  
goto   LCDRdy1  
  
BANKSEL TRISC

```

März 05, 06 15:20

mc_v1_12.pic

Seite 13/64

```

movlw d_LCD_schreiben ;Port wieder zurueck
movwf TRISC
BANKSEL PORTC
bcf a_LCD_RW

return
;-----
LCDDisplayClear:
movlw B'00000001'
goto LCDCCom
;-----
LCDCursorHome:
movlw B'00000010'
goto LCDCCom
;-----
LCDCursorZeile1:
movlw B'10000000'
goto LCDCCom
;-----
LCDCursorZeile2:
movlw B'11000000'
goto LCDCCom
;-----
LCDPos:
iorlw B'10000000'
goto LCDCCom
;-----
LCDString:
BANKSEL EEADR
movwf EEADR ;Uebergabe der LOW Adr von W nach EEADR
movf lcdtextadrh,W ;Uebergabe der High Adr an EEADR
movwf EEADRH
BANKSEL EECON1
bsf EECON1, EEPGD ;lesen aus dem Programmspeicher

LCDString3:
bcf INTCON,GIE
btfsc INTCON,GIE
goto LCDString3
bsf EECON1, RD ;lesen
nop ;nach 2 Takten sind Daten verfuegbar
nop
bsf INTCON,GIE
BANKSEL EEADR
movf EEDATA, W ;Byte holen
BANKSEL PORTC
call LCDPos ;1. Byte ist die Position in der LCD
BANKSEL EEADR
incf EEADR,F ;naechstes Byte
SKPNZ
incf EEADRH,F ;Ueberlauf behandeln

LCDString1:
BANKSEL EECON1
bsf EECON1, EEPGD ;und wieder Lesen aus dem Programmspeicher

LCDString2:
bcf INTCON,GIE
btfsc INTCON,GIE
goto LCDString2
bsf EECON1, RD
nop
nop
bsf INTCON,GIE
BANKSEL EEADR
movf EEDATA, W
BANKSEL PORTC
andlw B'11111111' ;letztes Byte ist 0
btfsc STATUS, Z ;? letztes Byte erreicht
return ;! Ende

call LCDChar ;0 Ausgabe auf LCD
BANKSEL EEADR
incf EEADR,F ; naechste Adresse
SKPNZ
incf EEADRH,F ; und Ueberlauf behandeln
goto LCDString1
;-----
LCDSpace:
movlw ' '
goto LCDChar
;-----

```

Sonntag März 05, 2006

mc_v1_12.pic

März 05, 06 15:20

mc_v1_12.pic

Seite 14/64

```

LCDHEX:
movwf LCDByte2
swapf LCDByte2,W
call BCDDtoASCII
call LCDChar
movf LCDByte2,W
call BCDDtoASCII
goto LCDChar
;-----
haupt:
call init ;MC initialisieren
call dds_init ;DDS initialisieren
bsf light
call LCDInit ;LCD initialisieren
LCDStr txtc1
LCDStr txtc2
movlw .7
movwf schleife

haupt0_2:
call t100mSek
movlw '.'
call LCDChar
decfsz schleife,F
goto haupt0_2
call eeladen ;Grunddaten aus Eeprom laden

btfsc taste
goto haupt0_1
call LCDDisplayClear ;LCD loeschen + Cursor auf null
LCDStr txtc3
LCDStr txtc4
call t1Sek
call t1Sek
call t1Sek
call t1Sek
call t1Sek

haupt0_3:
btfss taste
goto haupt0_3
PAGESEL PAGE2
goto msetup
PAGESEL PAGE0

haupt0_1:
call LCDDisplayClear ;LCD loeschen + Cursor auf null
btfss blight ;? Dauerlicht im Eeprom
bcf light ;! Licht nicht wieder ausschalten
call zslsetzen2sek ;Timer 2Sek starten
bsf bzslausein ;Funktion Timer ein->aus aktivieren
clrf ebene ;Menueebene 0 einschalten
LDK1 step,2 ;Abstimmsschritte=2 (50Hz)
LDK1 idlccounter,.16 ;Idleschleife initialisieren (16 Durchlaufefe)
bsf blcdneu ;LCD anzeigen
bsf bddsneu ;Startfrequenz einstellen
clrf band ;startband
PAGESEL PAGE2
call umsch_40m ;auf 40mBand schalten
PAGESEL PAGE0
call quitungston ;

haupt1:
bsf bidle ;Bit fuer keine Funktion setzen
movf impulse,F ;? Drehimpulse angefallen
SKPNZ
goto haupt5
bcf bidle ;! Bit fuer keine Funktion loeschen
call zslsetzen2sek
movf impulse,W ;
btfsc impulse,7 ; ? positive Impulse
goto haupt3
clrf impulse ; 1 wieder Vorbereiten fuer Interrupt
andlw B'01111111' ; < 127
movwf temp+9

haupt2:
; SCHLEIFE
call stepadd ; step addieren
decfsz temp+9,F ; ENDE Impulse=0
goto haupt2
goto haupt6

haupt3:
comf impulse,W ; 0 komplementaer bilden
clrf impulse ; wieder vorbereiten fuer Interrupt
addlw 1 ; komplementaer -> negation

```

7/32

MÄr 05, 06 15:20	mc_v1_12.pic	Seite 15/64
haupt4:	movwf temp+9 call stepsub ; SCHLEIFE decfsz temp+9,F ; ENDE Impulse=0 goto haupt4	
haupt6:	bsf bddsneu ; DDS neu berechnen bsf blcdneu ; LCD neu anzeigen	
haupt5:	btfsz blcdneu ;? LCD neu anzeigen Zeile1 goto haupt7 bcf bidle ;1 Bit fuer keine Funktion loeschen call LCDAnzeigeZ1 ; LCD neu anzeigen Zeile1 call LCDAnzeigeZ2 ; LCD neu anzeigen Zeile2 bcf blcdneu ; BIT loeschen LCD neuanzeige	
haupt7:	btfsz bddsneu ;? DDS neu berechnen goto haupt8 bcf bidle ;1 Bit fuer keine Funktion loeschen call ddsbinausrechnen; DDS neu ausrechnen call rx_to_dds ; Empfangsfrequenz laden bcf bddsneu ; bit loeschen	
haupt8:	btfsz taste ;? Taste gedruickt goto haupt9 bcf bidle ;1 Bit fuer keine Funktion loeschen btfsz blightauto ; ? Lichtautomatik an bsf light ; 1 Licht ein call MenuStart ; Entprellung abfragen movf tastennummer,W ; ? Tastennummer xorlw 1 SKPZ goto haupt8_1 ; 1 Taste kurz gedruickt incf step,F ; Abstimmsschritte aendern movf step,W ; ? grobsten Schritt erreicht btfsz bgen ; ? Radioempfang xorlw .5 ; 1 MHz schritte als max btfsz bgen xorlw .4 ; 1kHz schritte als max SKPZ goto haupt8_2 movlw 1 ; 1 wieder feinsten Schritt movwf step	
haupt8_2:	bsf blcdneu ; neue LCDAnzeige goto haupt8_3	
haupt8_1:	movf tastennummer,W ; xorlw 2 SKPZ goto haupt9 ; PAGESEL PAGE2 call Menuauswertung ; 2 Taste lang gedruickt PAGESEL PAGE0 ; Menueinsprung	
haupt8_3:	btfsz blightauto ; ? Lichtautomatik an call zslsetzen2sek ; 1 Lichtautomatik starten	
haupt9:	btfsz bgen ;? Radioempfang goto haupt94 ; btfsz bkeyer ;0 ? keyer ein goto haupt91 ; btfsz punktpin ; 1 ? Punktpaddle bsf bpunkt ; 1 merken btfsz strichpin ; ? Strichpaddle bsf bstrich ; 1 merken btfsz bpunkt ; ? Punkt gemerkt goto haupt10 bcf bidle ; 1 Bit fuer keine Funktion loeschen call tx_to_dds ; Sendefrequenz einstellen bsf tonenable ; Mithoerton ein bsf senderein ; Sender einschalten call punkt ; Punktdauer warten bcf senderein ; Sender ausschalten bcf tonenable ; Mithoerton aus movf rxdelay,W ; rxdelay call DELAY ; warten call rx_to_dds ; Empfangsfrequenz einstellen call punkt ; Punktdauer warten bcf bpunkt ; gemerkten Punkt loeschen	

MÄr 05, 06 15:20	mc_v1_12.pic	Seite 16/64
haupt10:	btfsz bstrich ; ? Strich gemerkt goto haupt11 bcf bidle ; 1 Bit fuer keine Funktion loeschen call tx_to_dds ; Sender einschalten bsf tonenable ; Mithoerton ein bsf senderein ; Sender einschalten call punkt ; Punktdauer warten call punkt ; Punktdauer warten call punkt ; Punktdauer warten bcf senderein ; Sender ausschalten bcf tonenable ; Mithoerton aus movf rxdelay,W ; rxdelay call DELAY ; warten call rx_to_dds ; Empfangsfrequenz einstellen call punkt ; Punktdauer warten bcf bstrich ; gemerkten Strich loeschen goto haupt93	
haupt91:	btfsz punktpin ; 0 ? Handtaste Taste gedruickt goto haupt92 btfsz senderein ; 1 ? Sender noch aus goto haupt93 bcf bidle ; 1 Bit fuer keine Funktion loeschen call tx_to_dds ; Sender einschalten bsf senderein ; Sender einschalten bsf tonenable ; Mithoerton ein goto haupt93	
haupt92:	btfsz senderein ; 0 ? Sender noch an goto haupt93 bcf senderein ; 1 Sender ausschalten bcf tonenable ; Mithoerton aus movf rxdelay,W ; rxdelay call DELAY ; warten call rx_to_dds ; RX-Frequenz ausgeben goto haupt93	
haupt94:	bcf bpunkt ;1 Radioempfang keyeraktivitaeten loeschen bcf bstrich	
haupt93:		
haupt11:	movf zsl,F ;? Timer 2 Sekunden aktiv SKPNZ goto haupt11_1 btfsz bzslausein ;1 ? Funktion aus -> ein schon ausgefuehrt goto haupt12 btfsz blightauto ; 1 ? Autolicht bsf light ; 1 Licht ein bcf bzslausein ; Funktion aus -> ein deaktivieren bsf bzslausein ; Funktion ein -> aus aktivieren bsf blcdneu ; LCD neu Anzeigen goto haupt12	
haupt11_1:	btfsz bzslausein ;0 ? Funktion ein -> aus schon ausgefuehrt goto haupt12 btfsz blightauto ; 1 ? Autolicht bcf light ; 1 Licht aus bsf bzslausein ; Funktion aus -> ein aktivieren bcf bzslausein ; Funktion ein -> aus deaktivieren bsf blcdneu ; LCD neu Anzeigen	
haupt12:	btfsz bgen ;? Radioempfang goto haupt13 btfsz bkeyeranz ;0 ? Keyergeschw. anzeigen goto haupt13 call zslsetzen2sek ; 1 Timer 2 Sek setzen call LCDAnzeigeZ1 ; LCD neu anzeigen Zeile1 call LCDAnzeigeZ2 ; LCD neu anzeigen Zeile2 bcf bidle ; kein idle mehr (Funktionsbit) bcf bkeyeranz ; bit loeschen	
haupt13:	btfsz bidle ;? keine Funktion ausgefuehrt goto haupt80 ; movf idlecouter,W ;1 alle Nebenfunktionen beginnen hier xorlw .1 ; ? Idlefunktion 1 SKPZ goto haupt801 ; PAGESEL PAGE2 call AnzeigeBatt ; 1 Batteriespannung anzeigen (Zeitspalte 1)	


```

MÄr 05, 06 15:20          mc_v1_12.pic          Seite 17/64
PAGESEL PAGE0
haupt801:
movf   idlecounter,W    ;
xorlw  .2                ; 2 Idlefunktion 2 (Zeitspalte 2)
SKPZ
goto   haupt802         ;
PAGESEL PAGE2
call   AnzeigeAGC      ; S-Meter anzeigen
PAGESEL PAGE0
haupt802:
movf   idlecounter,W    ;
xorlw  .3                ; 3 Idlefunktion 3 (Zeitspalte 3)
SKPZ
goto   haupt803         ;
BANKSEL bank1
RL2    ukeyer+2         ; Summe von UKeyer x 16
RL2    ukeyer+2         ; weil nur 16 Idledurchlaeufer
RL2    ukeyer+2         ; wir brauchen 256 zum auswerten
RL2    ukeyer+2         ; (Durchschnittserrechnung)
LD1    ukeyer, ukeyer+3; Summe von UKeyer holen
movf   keyergeschw,W    ; gemerkte Ukeyer holen
subwf  ukeyer+3         ; und vergleichen
SKPNZ  ;                 ; ? Werte sind ungleich keyergesch
goto   haupt803_1       ;
LD1    ukeyer+3,ukeyer ;
incf   keyergeschw,W    ; oder keyergeschw +1
subwf  ukeyer+3         ;
SKPNZ  ;                 ;
goto   haupt803_1       ;
LD1    ukeyer+3,ukeyer ;
defc   keyergeschw,W    ; oder keyergeschw -1
subwf  ukeyer+3         ;
SKPNZ  ;                 ;
goto   haupt803_1       ;
BANKSEL bank0           ; 1 Keyergeschw neu anzeigen
bsf    bkeyeranz
BANKSEL bank1
LD1    keyergeschw, ukeyer; und neuen Wert merken
haupt803_1:
CLR4   ukeyer           ; Summenwert loeschen
BANKSEL bank0
haupt803:
decfsz idlecounter,F    ; Idlefunktion + 1
goto   haupt899         ; ? 16x Idle
movlw  .16              ; 1 wieder von vorn
movwf  idlecounter
haupt899:
call   UmessKeyerPoti   ; Keyerpoti messen
BANKSEL bank1
movwf  ukeyer           ; als word speichern
clrf   ukeyer+1         ; hoeherwertiges byte loeschen
ADD2   ukeyer+2, ukeyer; und als word addieren
BANKSEL bank0
;
sleep
haupt80:
haupt99:
goto   haupt1           ;und wieder von vorn
;=====
;Funktion   Anzeige der Frequenz auf LCD Zeile
;Eingang   Frequenz im Speicher
;Ausgang   LCD
FunktionPos: equ .11
ubattpos:   equ 4ch
LCDAnzeigeZ2:
btfs   bgen             ;? Radioempfang
goto   LCDAnZ202        ;
btfs   bkeyer           ;0 ? Keyer
goto   LCDAnZ202        ;
btfs   bkeyeranz        ; 1 ? Keyergeschw anzeigen
goto   LCDAnZ202        ;
LCDStr  text4           ; 1 Geschw. Anzeigen
call   UmessKeyerPoti
call   ukey2anz
call   LCDHEX
call   LCDSpace
call   LCDSpace
call   LCDSpace

```

```

MÄr 05, 06 15:20          mc_v1_12.pic          Seite 18/64
return
LCDAnZ202:
movlw  40H              ;
call   LCDPos           ;Cursor bewegen
btfs   bxit             ;? xit ein
goto   LCDAnZ203        ;
movlw  'x'              ;1 Zeile 2 RX Frequenz
goto   LCDAnZ201        ;
LCDAnZ203:
movlw  'A'              ;
btfs   bvfo             ;? VFO B aktiv
movlw  'B'              ;1 B anzeigen
btfs   brit            ;? Rit aktiv
goto   LCDAnZ201        ;
movlw  'a'              ;1 ? VFO B aktiv
btfs   bvfo             ;
movlw  'b'              ; 1 B anzeigen
LCDAnZ201:
call   LCDChar          ;anzeigen
movlw  frequenza+3     ;
btfs   bvfo             ;? Frequenz B
movlw  frequenzb+3     ;1 VFO B in Indexregister laden
goto   LCDAnzeige      ;
LCDAnzeigeZ1:
clr    LCDPos           ;1. Position Zeile 1
call   LCDPos           ;Cursor bewegen
btfs   bxit             ;
goto   LCDAnZ102        ;
movlw  'r'              ;
goto   LCDAnZ101        ;
LCDAnZ102:
movlw  'A'              ;
btfs   bvfo             ;? VFO B aktiv
movlw  'B'              ;1 B anzeigen
btfs   brit            ;
goto   LCDAnZ101        ;
movlw  'a'              ;
btfs   bvfo             ;? VFO B aktiv
movlw  'b'              ;1 B anzeigen
LCDAnZ101:
call   LCDChar          ;anzeigen
movlw  frequenza+3     ;
btfs   bvfo             ;? Frequenz B
movlw  frequenzb+3     ;1 VFO B in Indexregister laden
LCDAnzeige:
movwf  FSR              ;
bsf    bnul             ;BIT fuehrende Null setzen
swaf   INDF,W           ;
call   LCDfrqbyte       ;oberes Nibbel
movf   INDF,W           ;
call   LCDfrqbyte       ;unteres Nibbel
movlw  '.'              ;Punkt zur besseren Lesbarkeit
call   LCDChar
decf   FSR              ;naechstes Byte
swaf   INDF,W           ;oberes Nibbel
call   LCDfrqbyte
movf   step,W          ;
xorlw  .4
SKPNZ
goto   LCDAnz4
movf   INDF,W           ;unteres Nibbel
call   LCDfrqbyte
decf   FSR              ;naechstes Byte
swaf   INDF,W           ;oberes Nibbel
call   LCDfrqbyte
movlw  '.'              ;Punkt zur besseren Lesbarkeit
call   LCDChar
movf   step,W          ;? 1kHz schritte
xorlw  3
btfs   STATUS,Z        ;
goto   LCDAnz1
movf   INDF,W           ;0 weiter mit der Anzeige
call   LCDfrqbyte
movf   step,W          ;? 50Hz Schritte
xorlw  2
btfs   STATUS,Z        ;
goto   LCDAnz2
decf   FSR              ;0 weiter mit der Anzeige

```

```

MÄr 05, 06 15:20                mc_v1_12.pic                Seite 19/64

        swapf   INDF,W
        call    LCDfrqbyte
        movf    step,W
        xorlw   1                ;? 10Hz Schritte
        btfs   STATUS,Z
        goto    LCDAnz3
        movf    INDF,W           ;0 1Hz anzeigen
        call    LCDfrqbyte
        goto    LCDAnz9
LCDAnz4:                ;5 Leerzeichen ausgeben
        call    LCDSpace
        call    LCDSpace
LCDAnz1:                ;3 Leerzeichen ausgeben
        call    LCDSpace
LCDAnz2:                ;3 Leerzeichen ausgeben
        call    LCDSpace
LCDAnz3:                ;3 Leerzeichen ausgeben
        call    LCDSpace
LCDAnz9:                ;kein Leerzeichen ausgeben
        return
;=====
;Funktion   eine Zahl auf LCD ausgeben Fuehrende 0 als Leerzeichen
LCDfrqbyte:
        andlw   B'00001111'     ;unteres Nibbel filetern
        btfs   STATUS,Z         ;? Zeichen = 0
        goto    LCDfrq3
        btfs   bnull           ;1 ? fuehrende Null
        goto    LCDfrq1
        movlw   '              ; 1 Leerzeichen ausgeben
        goto    LCDfrq4
LCDfrq1:
LCDfrq3:
        iorlw   0x30            ;
        bcf    bnull
LCDfrq4
        goto    LCDChar
;=====
;Abfragen Tastenstatus mit Entprellung
;Ausgang   Nummer der Taste in tastennummer
;          1-2 sind gueltige Tasten 0= ungueltig
tastcounter    equ    .10      ;Anzahl der Eingabesequenzen die geprueft werden
                                ;auf Gleichheit

tastaturstatus:
        movlw   tastcounter
        movwf   schleife       ;zwischenspeichern
Tastaturst1:   ;SCHLEIFE(1)
        movf    PORTB,W        ;Tastenzustand einlesen
        andlw   B'00010000'    ;filtern
        movwf   schleife+1     ;merken
        call    tlmSek         ;1 mSek warten
        movf    PORTB,W        ;erneut einlesen
        andlw   B'00010000'
        xorwf   schleife+1,W   ;und vergleichen
        btfs   STATUS,Z         ;? war die Eingabe gleich
        goto    tastaturstatus ;0 nein wieder von ganz vorn
        decfsz schleife        ;1 gueltig Zaehler dec
        goto    Tastaturst1    ;? gueltige Eingabenzahl erreicht
        movf   schleife+1,W    ;1 Eingabe auswerten
        andlw   B'00010000'
        btfs   STATUS,Z
        goto    Tastaturst2
        movlw   1
        movwf   tastennummer   ;1 abspeichern
        bsf    STATUS,C
        goto    Tastaturst99    ;ENDE
Tastaturst2:
        movlw   0               ;keine gueltige Tastennummer
        movwf   tastennummer   ;Tastennummer 0 abspeichern
        bcf    STATUS,C
Tastaturst99:
        return
;=====
;Funktion   loeschen des Tempram alle 10 Byte oder 4 untersten Byte
cleartemp:
        clrf   temp+4

```

```

MÄr 05, 06 15:20                mc_v1_12.pic                Seite 20/64

        clrf   temp+5
        clrf   temp+6
        clrf   temp+7
        clrf   temp+8
        clrf   temp+9
cleartemp4:
        clrf   temp
        clrf   temp+1
        clrf   temp+2
        clrf   temp+3
        return
;-----
;Funktion   add des eingestellten Schritt zur Frequenz
;          1kHz, 100Hz, 10Hz oder 1Hz
stepadd:
        movf   step,W           ;? step
        btfs   STATUS,Z         ;0 Schritt = 1 Hz
        goto    bcdadd1
        xorlw   1
        btfs   STATUS,Z
        goto    bcdadd10       ;1 Schritt = 10 Hz
        movf   step,W
        xorlw   2
        btfs   STATUS,Z
        goto    bcdadd50       ;2 Schritt = 50 Hz
        movf   step,W
        xorlw   3
        SKPNZ
        goto    bcdadd1000     ;3 Schritt = 1000 Hz
        movf   step,W
        xorlw   4
        SKPNZ
        goto    bcdadd100k     ;4 Schritt = 100kHz
        return
;-----
bcdadd100k:
        call    fpointerladen
        call    cleartemp4
        movlw   10h
        movwf   temp+2
        goto    bcdadd4
;-----
bcdadd1000:
        call    fpointerladen
        call    cleartemp4
        movlw   10h
        movwf   temp+1
        goto    bcdadd4
;-----
bcdadd100:
        call    fpointerladen
        call    cleartemp4
        movlw   .1
        movwf   temp+1
        goto    bcdadd4
;-----
bcdadd50:
        call    fpointerladen
        call    cleartemp4
        movlw   50H
        movwf   temp
        goto    bcdadd4
;-----
bcdadd10:
        call    fpointerladen
        call    cleartemp4
        movlw   10H
        movwf   temp
        goto    bcdadd4
;-----
bcdadd1:
        call    fpointerladen
        call    cleartemp4
        incf   temp,F
;-----
bcdadd4:
        movlw   4               ;4 byte add
        movwf   temp+4
bcdadd41:

```

März 05, 06 15:20

mc_v1_12.pic

Seite 21/64

```

movf pointer1,W ;pointer1 ins FSR laden
movwf FSR
movf INDF,W
movwf temp+5 ;in temp+9 zwischenspeichern
movf pointer2,W ;pointer2 ins FSR laden
movwf FSR
movf INDF,W
addwf temp+5,F ;und beide werte addieren
movlw 6
addwf temp+5,W
btfss STATUS,DC ;? BCD korrektur low durchfuehren
goto bcdadd42
movwf temp+5 ;! ja
bcdadd42:
movlw 60H ;? BCD korrektur high durchfueren
addwf temp+5,W
btfss STATUS,C
goto bcdadd43
movwf temp+5 ;! ja
movf pointer1,W
movwf FSR
incf FSR,F ; ueberlauf addieren
incf INDF,F
decf FSR,F
bcdadd43:
movf pointer1,W ;ergebnis in pointer1 laden
movwf FSR
movf temp+5,W
movwf INDF
incf pointer1
incf pointer2
decfsz temp+4
goto bcdadd41
return
;=====
;Funktion sub des eingestellten Schritt zur Frequenz
; 1kHz, 50Hz, 10Hz oder 1Hz
stepsub:
movf step,W
btfsc STATUS,Z
goto bcdsub1
xorlw 1
btfsc STATUS,Z
goto bcdsub10
movf step,W
xorlw 2
btfsc STATUS,Z
goto bcdsub50
movf step,W
xorlw 3
btfsc STATUS,Z
goto bcdsub1000 ;3 Schritt = 1000 Hz
movf step,W
xorlw 4
btfsc STATUS,Z
goto bcdsub100k ;4 Schritt = 100kHz
return
;-----
bcdsub100k:
call fpointerladen
call cleartemp4
movlw 10h
movwf temp+2
goto bcdsub4
;-----
bcdsub1000:
call fpointerladen
call cleartemp4
movlw 10h
movwf temp+1
goto bcdsub4
;-----
bcdsub100:
call fpointerladen
call cleartemp4
movlw .1
movwf temp+1
goto bcdsub4
;-----

```

März 05, 06 15:20

mc_v1_12.pic

Seite 22/64

```

bcdsub50:
call fpointerladen
call cleartemp4
movlw 50H
movwf temp
goto bcdsub4
;-----
bcdsub10:
call fpointerladen
call cleartemp4
movlw 10H
movwf temp
goto bcdsub4
;-----
bcdsub1:
call fpointerladen
call cleartemp4
incf temp,F
;-----
bcdsub4:
movlw 4
movwf temp+4
bcdsub41:
movf pointer1,W
movwf FSR
movf INDF,W
movwf temp+5 ;
movf pointer2,W
movwf FSR
movf INDF,W
subwf temp+5,F
rlf temp+6,F ;Carry merken
btfsc STATUS,DC
goto bcdsub42
movlw 6
subwf temp+5,F
bcdsub42:
btfsc temp+6,C
goto bcdsub43
movlw 60H
subwf temp+5,F
incf FSR,F
incf INDF,F
decf FSR,F
bcdsub43:
movf pointer1,W
movwf FSR
movf temp+5,W
movwf INDF
incf pointer1
incf pointer2
decfsz temp+4
goto bcdsub41
return
bcdtobin:
call cleartemp ;temp loeschen
movf pointer1,W ;4 byte nach temp laden
movwf FSR ;A oder B VFO
movf INDF,W
movwf temp
incf FSR,F
movf INDF,W
movwf temp+1
incf FSR,F
movf INDF,W
movwf temp+2
incf FSR,F
movf INDF,W
movwf temp+3
movlw D'32' ;32 bit bcd in bin umwandeln
movwf temp+8
bcdtobin1:
bcf STATUS,C
rrf temp+3
rrf temp+2
rrf temp+1
rrf temp
rrf temp+7

```

MÄr 05, 06 15:20	mc_v1_12.pic	Seite 23/64
	<pre> rrf temp+6 rrf temp+5 rrf temp+4 movlw temp movwf FSR movlw 4 movwf temp+9 bcdtobin3: btfss INDF,7 goto bcdtobin4 movlw 30H subwf INDF,F bcdtobin4: btfss INDF,3 goto bcdtobin5 movlw 3 subwf INDF,F bcdtobin5: incf FSR,F decfsz temp+9,F goto bcdtobin3 decfsz temp+8,F goto bcdtobin1 return </pre>	

addzf:	<pre> LD4 temp, zwischenfrequenz ADD4 temp+4, temp return </pre>	

subzf:	<pre> LD4 temp, zwischenfrequenz SUB4 temp+4, temp return </pre>	

;Funktion	Multiplikation der Frequenz mit der 1Hz DDS-Konst	
;Ausgang	binwert steht in tempkonst2+4	
muldds:	<pre> movf ddskonst,W ;Konstante 4Byte in temp laden movwf tempkonst1 movf ddskonst+1,W movwf tempkonst1+1 movf ddskonst+2,W movwf tempkonst1+2 movf ddskonst+3,W movwf tempkonst1+3 movlw D'32' movwf temp+8 </pre>	
muldds1:	<pre> movlw tempkonst1 ;schleife (1) movwf pointer1 ; pointer1 laden movlw tempkonst2 movwf pointer2 ; pointer2 laden clrf temp+1 ; Ueberlauf loeschen bcf STATUS,C ; Carry loeschen rrf temp+7 ; Faktor1 nach rechts schieben rrf temp+6 rrf temp+5 rrf temp+4 btfss STATUS,C ; ? Carry = 1 nach RR goto muldds2 movlw 8 ; 1 Schleife laden mit 8 movwf temp+2 </pre>	
muldds5:	<pre> movf pointer1,W ; schleife (2) movwf FSR ; 1 Pointer laden movf INDF,W ; wert1 holen movwf temp ; zwischenspeichern movf pointer2,W ; Pointer2 laden movwf FSR ; ins indirekte Register movf temp+1,W ; gemerketen Ueberlauf holen clrf temp+1 ; und loeschen addwf INDF,F ; zum Ergebnis addieren btfss STATUS,C ; ? Ueberlauf goto muldds4 movlw 1 ; 1 neuen Ueberlauf movwf temp+1 ; merken </pre>	
muldds4:	<pre> movf temp,W ; Wert2 = Wert2 + Wert1 </pre>	

MÄr 05, 06 15:20	mc_v1_12.pic	Seite 24/64
	<pre> addwf INDF,F btfss STATUS,C ; ? Ueberlauf goto muldds3 movlw 1 ; 1 neuen Ueberlauf movwf temp+1 ; merken muldds3: incf pointer1 incf pointer2 decfsz temp+2 ; ende (2) goto muldds5 muldds2: bcf STATUS,C ; Carry loeschen rlf tempkonst1 ; Wert1 nach links schieben rlf tempkonst1+1 ; Wert1 = 2 * Wert1 rlf tempkonst1+2 rlf tempkonst1+3 rlf tempkonst1+4 rlf tempkonst1+5 rlf tempkonst1+6 rlf tempkonst1+7 decfsz temp+8,F ;ende (1) 32 mal durchlaufen goto muldds1 return </pre>	

;Funktion	Ram fuer Multiplikation loeschen	
mulramclr:	<pre> movlw temp ;Ram loeschen movwf FSR ;fsr fuer indirekte Adressierung laden mulramclr1: clrf INDF ;Fileregister indirekt loeschen movlw tempkonst2+7 ;Ramende in Akku subwf FSR,w ;Test ob Ende erreicht btfsc STATUS,Z ;Auswertung des Zerroflags goto mulramclr2 ;Ende erreicht incf FSR,f ;Ende nicht erreicht goto mulramclr1 ;wieder von vorn mulramclr2: return </pre>	

;Funktion	laden der entsprechenden Ramadr in den Pointerram	
fpointerladenddstx:	<pre> btfsc brit ;? Rit aktiv goto fp103 ;1 Zeile 2 = tx -> break btfsc bxit ;0 ? xit aktiv goto fp103 ; 1 Zeile 2 = tx -> break goto fp101 ;Zeile 1 = tx </pre>	
fpointerladen:	<pre> movlw temp ;Laden fuer Abstimmung movwf pointer2 btfsc bxit ;? xit goto fp103 ;1 Zeile 2 abstimmen </pre>	
fpointerladendds:	<pre> fp101: movlw frequenza ;Zeile 1 = rx btfsc bvfo ; movlw frequenzb ; goto fp102 ; fp103: movlw frequenza ;Zeile 2 = tx btfss bvfo ; movlw frequenzb ; fp102: movwf pointer1 return </pre>	

ddsbinarechnen:	<pre> btfss bgen ;? Radioempfang goto dbr01 ; LDK4 temp,14h,0,0,0 ; btfss bvfo ;1 ? VFOb aktiv goto dbr02 ; CMP4 temp,frequenzb ; 1 VFO = VFOb goto dbr03 ; dbr02: CMP4 temp,frequenza ; 0 VFO = VFOa ; ? VFO > 14MHz </pre>	

```

MÄr 05, 06 15:20          mc_v1_12.pic          Seite 25/64
dbr03:
    bsf    bzfadd          ; 0 ZF Addieren
    SKPC
    bcf    bzfadd          ; 1 ZF Subtrahieren
dbr01:
    call   mulramclr       ;Ram loeschen
    call   fpointerladends ;entsprechenden VFO laden
    call   bcdtobin       ;BCD to BIN wandeln
    btfs   bddsohnezf     ;? mit ZF
    goto   dbr04          ;
    btfs   bzfadd         ;1 ? ZF subtrahieren
    call   subzf          ; 1 Zf binaer subtrahieren
    btfs   bzfadd         ; ? ZF addieren
    call   addzf          ; 1 ZF binaer addieren
dbr04:
    call   muldds         ;Multiplikation mit 1Hz Konstante
    movf   tempkonst2+3,W ;Ergebnis in den RXbin_speicher laden
    BANKSEL bank1        ;Speicher in Bank1
    movwf  ddsbinrx
    BANKSEL bank0
    movf   tempkonst2+4,W
    BANKSEL bank1
    movwf  ddsbinrx+1
    BANKSEL bank0
    movf   tempkonst2+5,W
    BANKSEL bank1
    movwf  ddsbinrx+2
    BANKSEL bank0
    movf   tempkonst2+6,W
    BANKSEL bank1
    movwf  ddsbinrx+3
    BANKSEL bank0
    call   mulramclr       ;TX Frequenz ausrechnen
    call   fpointerladendstx ;entsprechenden VFO laden
    call   bcdtobin       ;BCD to BIN wandeln
    call   muldds         ;Multiplikation mit 1Hz Konstante
    movf   tempkonst2+3,W ;Ergebnis in den TXbin_speicher laden
    BANKSEL bank1        ;Speicher in Bank1
    movwf  ddsbintx
    BANKSEL bank0
    movf   tempkonst2+4,W
    BANKSEL bank1
    movwf  ddsbintx+1
    BANKSEL bank0
    movf   tempkonst2+5,W
    BANKSEL bank1
    movwf  ddsbintx+2
    BANKSEL bank0
    movf   tempkonst2+6,W
    BANKSEL bank1
    movwf  ddsbintx+3
    BANKSEL bank0
    return
;-----
tx_to_dds:
    BANKSEL bank1        ;TXbin_speicher in den DDS laden
    LD4    ddsdword,dds bintx
    goto   frq28bit_to_dds
rx_to_dds:
    BANKSEL bank1        ;RXbin_speicher in den DDS laden
    LD4    ddsdword,dds binrx
frq28bit_to_dds:
    BANKSEL bank0
    call   control_to_dds
    BANKSEL bank1
    LD2    ddsword,dds dword
    bcf    ddsword+1,7
    bsf    ddsword+1,6
    BANKSEL bank0
    call   word_to_dds
    BANKSEL bank1
    RL4    ddsdword
    RL4    ddsdword
    LD2    ddsword,dds dword+2
    bcf    ddsword+1,7
    bsf    ddsword+1,6
    BANKSEL bank0
    call   word_to_dds
    return

```

```

MÄr 05, 06 15:20          mc_v1_12.pic          Seite 26/64
;-----
dds_init:
    BANKSEL bank0
    call   control_to_dds
    BANKSEL bank1
    movlw  B'01000100'
    movwf  ddsword+1
    movlw  B'10111011'
    movwf  ddsword
    BANKSEL bank0
    call   word_to_dds
    BANKSEL bank1
    movlw  B'01000111'
    movwf  ddsword+1
    movlw  B'10101110'
    movwf  ddsword
    BANKSEL bank0
    call   word_to_dds
    BANKSEL bank1
    LDK4   ddsdword,2,3dh,70h,0a3h ;7MHz ausgeben als 1. ausgeben
    BANKSEL bank0
    goto   frq28bit_to_dds
;-----
control_to_dds:
    BANKSEL bank1
    movlw  B'00100000'
    movwf  ddsword+1
    movlw  B'00000000'
    movwf  ddsword
;-----
word_to_dds:
    BANKSEL bank0        ;WORD in DDS laden
    bcf    ddsfsyn        ;DDS_aufnahme aktivieren
    BANKSEL bank1
    movf   ddsword+1,W
    BANKSEL bank0
    call   byte_to_dds    ;Byte laden
    BANKSEL bank1
    movf   ddsword,W
    BANKSEL bank0
    call   byte_to_dds    ;Byte laden
    bsf    ddsfsyn        ;Ende des Wortes
    return
;-----
byte_to_dds:
    movwf  temp           ;ein Byte in DDS laden
    movlw .8              ;Byte zwischenspeichern
    movwf  schleife      ;8 Bit
bytetodds01:
    rlf   temp,F          ;SCHLEIFE(1)
    SKPC
    bcf   ddsdaten        ; ddsdaten 0
    SKPNC
    bsf   ddsdaten        ; ddsdaten 1
    bcf   ddsclk          ; fallende clk_flanke
    bsf   ddsclk
    decfsz schleife,F     ;ENDE(1) nach 8 bit
    goto  bytetodds01
    return
;=====
;Definition der ADC-Messkanäle
kanalRA0:
    equ   b'00000000'
kanalRA1:
    equ   b'00001000'
kanalRA2:
    equ   b'00010000'
kanalRA3:
    equ   b'00011000'
kanalRA5:
    equ   b'00100000'
Uagc:
    BANKSEL bank1
    bsf   ADCON1,7        ;auf 10 bit schalten
    BANKSEL bank0
    movlw 0x41 | kanalRA1 ;anal. kanal RA1
    call   Umess
    BANKSEL bank1
    movwf werth
    movf  ADRESL,W
    movwf wert1
    bcf   ADCON1,7        ;auf 8 bit schalten
    BANKSEL bank0

```

März 05, 06 15:20	mc_v1_12.pic	Seite 27/64
	return	
UBatt:	movlw 0x41 kanalRA0 ;anal. kanal RA0 goto Umess	
UPowerRueck:	movlw 0x41 kanalRA3 ;anal. kanal RA3 goto Umess	
UPowerVor:	movlw 0x41 kanalRA2 ;anal. kanal RA2 goto Umess ;Keyerpoti	
UmessKeyerPoti:	movlw 0x41 kanalRA5 ; anal. kanal RA5	
Umess:	movwf ADCON0 ; ... als Messeingang setzen movlw 1 call DELAY bsf ADCON0,2 ;Start Messung	
ubmel:	btfsz ADCON0,2 ;warten bis Messung beendet goto ubmel ;Ergebnis in adresh und adresl movf ADRESH,W ;Ergebnis in W return	
;=====		
	;Funktion Anpassen der Potikurve durch Multiplikation	
	;Eingang Register W	
	;Ausgang Register W	
	;Register temp+9, temp+8, temp+6, temp+7	
ubatt_mul:	movwf temp+9 movlw .154 ;Multiplizieren mit 154 movwf temp+8 ; movf temp+9,W ; clrf temp+9 ; bsf temp+9,3 ;Schleifenzaehler = 8 CLR2 temp+6 ;	
uml:	bcf STATUS,C RL2 temp+6 RLF temp+8 SKPC goto um2 addwf temp+6 SKPNC incf temp+7	
um2:	decfsz temp+9 goto uml movlw .0 ;Addieren mit 0 bcf STATUS,C addwf temp+7,W	
	return	
;=====		
	;Funktion HEX zu BCD wandeln	
	;Eingang HEX in W	
	;Ausgang BCD in temp+6 2 Byte lang	
	;Register temp+5, temp+6, temp+7, temp+8, temp+9	
HEX2BCD:	movwf temp+8 movlw .8 movwf temp+9 clrf temp+6 clrf temp+7	
htb1:	movf temp+6,W andlw B'00001111' movwf temp+5 movlw .5 subwf temp+5,W btfsz STATUS,C goto htb2 movlw .3 addwf temp+6,F	
htb2:	movf temp+6,W andlw B'11110000'	

März 05, 06 15:20	mc_v1_12.pic	Seite 28/64
	movwf temp+5 movlw 50h subwf temp+5,W btfsz STATUS,C goto htb3 movlw 30h addwf temp+6,F	
htb3:	bcf STATUS,C rlf temp+8,F rlf temp+6,F rlf temp+7,F decfsz temp+9,F goto htb1 movf temp+6,W return	
;=====		
	;Funktion Dauer eines Punktes warten. Wartezeit wird lmSek-Einheiten und dem Timer2 gebildet. Waehrend der Wartezeit wird beim Senden die Ausgangsleistung geregelt. Weiter Funktion ist das abfragen der Paddel fuer Punkt oder Strichspeicher	
	;=====	
d_pr2h	equ .242 ;Timer2 Period Register	
d_pr2l	equ .242 ;Timer2 Period Register	
punkt:	ifdef debug return endif call UmessKeyerPoti ;Keyergeschw vom Poti lesen BANKSEL bank1 movwf keyergeschw+1 ;Keyergeschwindigkeit mit Potiwert laden BANKSEL bank0 call ukey2msek ;aus Potiwert umrechnen in mSek movwf zs3 ;Timer setzen movf rxdelay,W ; btfsz senderein ;? Sender ausgeschaltet subwf zs3,F ;1 TX->RX Delay subtrahieren von zs3 movlw d_pr2h ; btfsz senderein ;? Sender eingeschalten movlw d_pr2l ;0 kuerzere Pause BANKSEL PR2 ; movwf PR2 ; BANKSEL PORTA ; bsf T2CON,2 ;Timer starten ueber Interrupt ;SCHLEIFE(1)	
punkt3:	btfsz bkeyer ; goto punkt3_l ; ? keyer ein btfsz punktpin ; 1 eventuell Punkt bsf bpunkt ; zwischenspeichern btfsz strichpin ; eventuell Strich bsf bstrich ; zwischenspeichern	
punkt3_l	movf zs3,F ; SKPZ ;ENDE(1) zs3 = 0 goto punkt3	
punkt5:	bcf T2CON,2 ;Timer stop return	
;=====		
	;Funktion Anpassen der Potikurve durch Multiplikation	
	;Eingang Register W	
	;Ausgang Register W	
	;Register temp+9, temp+8, temp+6, temp+7	
keyermul:	movwf temp+9 movlw .42 movwf temp+8 movf temp+9,W clrf temp+9 bsf temp+9,3 CLR2 temp+6	
kum1:	bcf STATUS,C RL2 temp+6 RLF temp+8 SKPC goto kum2 addwf temp+6	

MÄr 05, 06 15:20

mc_v1_12.pic

Seite 29/64

```

kum2:    SKPNC
        incf    temp+7

        decfsz temp+9
        goto   kum1

        movlw  .9
        bcf   STATUS,C
        addwf temp+7,W

        return
;=====
zslsetzen2sek:
        movlw  .200
        movwf zsl
        return
;=====
;Schreiben in Eeprom
;Eingang:    Addr in data_ee_addr
;            Daten in data_ee_data
;Ausgang:    Daten im Eeprom intern
;=====
ewrite:
ew2:    BANKSEL bank3            ;
        ;SCHLEIFE (1)
        btfsc  EECON1,WR        ; schreibbit abfragen
        goto  ew2              ; ENDE(1) vorheriger Schreibvorgang beendet
        BANKSEL bank0          ;
        movf   data_ee_addr,w    ; Adresse zum EE-Register
        BANKSEL bank2          ;
        movwf  EEADR            ; data memory address to write
        BANKSEL bank0          ; Daten zum EEregister
        movf   data_ee_data,w    ;
        BANKSEL bank2          ;
        movwf  EEDATA          ; data memory value to write
        BANKSEL bank3          ;
        bcf   EECON1, EEPGD     ; point to data memory
        bsf   EECON1, WREN      ; enable writes
        ;SCHLEIFE (2)
        ; disable interrupts
        bcf   INTCON,GIE        ;
        btfsc INTCON,GIE        ;
        goto  ew1              ; ENDE (2) Interrupt wirklich gesperrt
        movlw 55h              ;
        movwf  EECON2          ; write 55h
        movlw 0aah              ;
        movwf  EECON2          ; write aah
        bsf   EECON1,WR        ; set wr bit to begin write
        ;SCHLEIFE (2)
        btfsc  EECON1,WR        ; schreibbit abfragen
        goto  ew3              ; ENDE(2) Schreibvorgang beendet
        bsf   INTCON, GIE      ; enable interrupts
        bcf   EECON1, WREN      ; disable writes
        BANKSEL bank0          ;
        return
;=====
;Lesen von Eeprom
;Eingang:    Addr in data_ee_addr
;Ausgang:    Daten in w und in data_ee_data
;=====
eread:
er2:    BANKSEL bank3            ;
        ;SCHLEIFE (1)
        btfsc  EECON1,WR        ; schreibbit abfragen
        goto  er2              ; ENDE(1) vorheriger Schreibvorgang beendet
        BANKSEL bank0          ;
        movf   data_ee_addr,w    ; Adresse zum
        BANKSEL bank2          ;
        movwf  EEADR            ; EE-Register
        BANKSEL bank3          ;
        bcf   EECON1, EEPGD     ; Datamemory
        bsf   EECON1, RD        ; Lesen aktivieren
        BANKSEL bank2          ;
        movf  EEDATA,W          ; Ergebnis nach W
        BANKSEL bank0          ;
        return
;=====
eeladen:
        movlw  .14
        movwf  schleife
        movlw  flag1

```

Sonntag Mär 05, 2006

mc_v1_12.pic

MÄr 05, 06 15:20

mc_v1_12.pic

Seite 30/64

```

        movwf  FSR
        clrf  data_ee_addr
        call  leread
        movlw .4
        movwf schleife
        movlw utemp
        movwf FSR
        movlw low(es_konst1)
        movwf data_ee_addr
        call  leread
        BANKSEL bank1
        LD4   s_konst1,utemp
        BANKSEL bank0
        movlw low(epitch)
        movwf data_ee_addr
        call  eread
        BANKSEL bank1
        andlw b'01111111'
        movwf tonhoehe
        BANKSEL bank0

        return

leread:
eel01:  call   eread
        movwf INDF
        incf  FSR,F
        incf  data_ee_addr,F
        decfsz schleife,F
        goto  eel01
        return
;=====
lewrite:
ewl01:  movf   INDF,W
        movwf data_ee_data
        call  ewrite
        incf  FSR,F
        incf  data_ee_addr,F
        decfsz schleife,F
        goto  ewl01
        return
;=====
quittungston:
        bsf   tonenable
        call  t50mSek
        bcf   tonenable
        return
;=====
endepage0:
        if endepage0 > 800h
        error "PAGE0 ueberschritten"
        endif

        org   800h
PAGE1:
;=====
u2pwr:  andlw  B'00111111'
        addwf PCL
        retlw 0 ;0
        retlw 0 ;1
        retlw 0 ;2
        retlw 01h ;3
        retlw 01h ;4
        retlw 01h ;5
        retlw 02h ;6
        retlw 03h ;7
        retlw 03h ;8
        retlw 04h ;9
        retlw 05h ;a
        retlw 06h ;b
        retlw 06h ;c
        retlw 07h ;d
        retlw 08h ;e
        retlw 09h ;f
        retlw 10h ;10
        retlw 11h ;11

```

15/32

MÄr 05, 06 15:20

mc_v1_12.pic

Seite 31/64

```

retlw 12h ;12
retlw 14h ;13
retlw 15h ;14
retlw 17h ;15
retlw 18h ;16
retlw 19h ;17
retlw 20h ;18
retlw 22h ;19
retlw 24h ;1a
retlw 25h ;1b
retlw 27h ;1c
retlw 30h ;1d
retlw 32h ;1e
retlw 34h ;1f
retlw 35h ;20
retlw 37h ;21
retlw 39h ;22
retlw 40h ;23
retlw 43h ;24
retlw 45h ;25
retlw 47h ;26
retlw 50h ;27
retlw 52h ;28
retlw 54h ;29
retlw 56h ;2a
retlw 58h ;2b
retlw 60h ;2c
retlw 63h ;2d
retlw 65h ;2e
retlw 67h ;2f
retlw 69h ;30
retlw 72h ;31
retlw 74h ;32
retlw 76h ;33
retlw 78h ;34
retlw 81h ;35
retlw 83h ;36
retlw 85h ;37
retlw 87h ;38
retlw 90h ;39
retlw 92h ;3a
retlw 94h ;3b
retlw 96h ;3c
retlw 99h ;3d
retlw 99h ;3e
retlw 99h ;3f

textc2: DT 40h,"V1.12 init",0
textc4: DT 40h,"V1.12 01.03.06",0

textc1: DT 0, "Blue Cool Radio ",0
textc3: DT 0, "DDS-VFO(c)DL4JAL",0
text3: DT 0, "SWR:" ,0
text4: DT 40H,"WpM:" ,0
text5: DT 0, "Menu",0
text6: DT 40H,"scanning",0
text7: DT 40H,"break " ,0
text8: DT .9, "save " ,0
text9: DT .9, "cancel " ,0
text10: DT 0, "S=0 0 uV (ok)",0
text11: DT 0, "S=9 50 uV (ok)",0
text12: DT -.9, "left " ,0
text13: DT 0, "cancel " ,0
text14: DT 0, "store " ,0
text15: DT 0, "read + scanning " ,0
text16: DT 0, "break " ,0
text17: DT 0, "spot " ,0
text18: DT 0, "tuning " ,0

texton: DT 40H," on " ,0
textoff: DT 40H," off " ,0

text80m: DT 40H," 80m " ,0
text40m: DT 40H," 40m " ,0
text30m: DT 40H," 30m " ,0
text20m: DT 40H," 20m " ,0
text17m: DT 40H," 17m " ,0
textgen: DT 40H," RADIO " ,0

```

MÄr 05, 06 15:20

mc_v1_12.pic

Seite 32/64

```

tbreak: DT 0, " 0 break " ,0
tvfoab: DT 0, " 1 VFO A/B " ,0
tband1: DT 0, " 2 Band 1 " ,0
tband2: DT 0, " 3 Band 2 " ,0
tband3: DT 0, " 4 Band 3 " ,0
tband4: DT 0, " 5 Band 4 " ,0
tband5: DT 0, " 6 Band 5 " ,0
tlight: DT 0, " 7 light on/off " ,0
tlightauto: DT 0, " 8 light auto " ,0
tsetup: DT 0, " 9 SETUP " ,0
tkeyer: DT 0, "10 keyer " ,0
tmemread: DT 0, "11 memory read " ,0
tmemwrite: DT 0, "12 memory store " ,0
ttune: DT 0, "13 tune " ,0
txitlk: DT 0, "14 xit lk " ,0
txit2k: DT 0, "15 xit 2k " ,0
trit: DT 0, "16 rit " ,0

stbreak: DT 0, " 0 Setup break " ,0
stddskonst: DT 0, " 1 DDS-Takt " ,0
stzf: DT 0, " 2 ZF (Hz) " ,0
stsmeter: DT 0, " 3 S-Meter eich." ,0
strxdelay: DT 0, " 4 TX->RX delay " ,0
stnton: DT 0, " 5 pitch " ,0
stdefault: DT 0, " 6 set default " ,0
;=====
endepagel:
    if endepagel > 1000h
        error "PAGE1 ueberschritten"
    endif

    org 1000h

PAGE2:
;-----
smenuanzeige:
    andlw B'00011111'
    addwf PCL
    goto smtbreak
    goto smtddskonst
    goto smtstof
    goto smtstmeter
    goto smstrxdelay
    goto smstnton
    goto smstdefault
    goto mtbreak
    goto mtbreak
    goto mtbreak
    goto mtbreak
    goto mtbreak
    goto mtbreak
    goto mtbreak
;-----
menuanzeige:
    andlw B'00011111'
    addwf PCL
    goto mtbreak
    goto mtvfoab
    goto mtband1
    goto mtband2
    goto mtband3
    goto mtband4
    goto mtband5
    goto mtlight
    goto mtlightauto
    goto mtsetup
    goto mtkeyer
    goto mtmemread
    goto mtmemwrite
    goto mttune
    goto mtxitlk
    goto mtxit2k
    goto mtrit
    goto mtbreak
    goto mtbreak
    goto mtbreak
    goto mtbreak
    goto mtbreak

```


März 05, 06 15:20

mc_v1_12.pic

Seite 33/64

```

goto mtbreak
goto mtbreak
goto mtbreak
goto mtbreak
goto mtbreak
goto mtbreak
goto mtbreak
goto mtbreak
goto mtbreak
goto mtbreak
;-----
smenuausfuehren:
andlw B'00001111'
addwf PCL
goto smbbreak
goto smddsconst
goto smzf
goto smsmeter
goto smrxdelay
goto smmton
goto smdefault
goto mbreak
goto mbreak
goto mbreak
goto mbreak
goto mbreak
goto mbreak
goto mbreak
;-----
menuausfuehren:
andlw B'00011111'
addwf PCL
goto mbreak
goto mvfoab
goto mband1
goto mband2
goto mband3
goto mband4
goto mband5
goto mlight
goto mlightauto
goto msetup
goto mkeyer
goto mmemread
goto mmemwrite
goto mtune
goto mxit1k
goto mxit2k
goto mrit
goto mbreak
goto mbreak
goto mbreak
goto mbreak
goto mbreak
goto mbreak
goto mbreak
goto mbreak
goto mbreak
goto mbreak
goto mbreak
goto mbreak
goto mbreak
;-----
uagc2s:
andlw B'00001111'
addwf PCL
retlw '0' ;0
retlw '1' ;1
retlw '2' ;2
retlw '3' ;3
retlw '4' ;4
retlw '5' ;5
retlw '6' ;6
retlw '7' ;7
retlw '8' ;8

```

März 05, 06 15:20

mc_v1_12.pic

Seite 34/64

```

retlw '+' ;9
retlw '+' ;10
retlw '+' ;11
retlw '+' ;12
retlw '+' ;13
retlw '+' ;14
retlw '+' ;15
;-----
bandspeichern:
movf band,W
andlw B'00001111'
addwf PCL
goto sp_ee_40m
goto sp_ee_30m
goto sp_ee_20m
goto sp_ee_17m
goto sp_ee_gen
goto sp_ee_gen
goto sp_ee_gen
;-----
eebandumsch:
andlw B'00001111'
addwf PCL
goto mband1
goto mband2
goto mband3
goto mband4
goto mband5
goto mband5
goto mband5
goto mband5
;-----
sp_ee_80m:
bcf INTCON,INTE ;
movlw .8
movwf schleife
movlw frequenza
movwf FSR
movlw LOW(e80m)
movwf data_ee_addr
sp_ee_80m_01:
movf INDF,W
movwf data_ee_data
PAGESEL PAGE0
call ewrite
PAGESEL PAGE2
incf FSR,F
incf data_ee_addr,F
decfsz schleife,F
goto sp_ee_80m_01
bsf INTCON,INTE ;
return
;-----
sp_ee_40m:
bcf INTCON,INTE ;
movlw .8
movwf schleife
movlw frequenza
movwf FSR
movlw LOW(e40m)
movwf data_ee_addr
sp_ee_40m_01:
movf INDF,W
movwf data_ee_data
PAGESEL PAGE0
call ewrite
PAGESEL PAGE2
incf FSR,F
incf data_ee_addr,F
decfsz schleife,F
goto sp_ee_40m_01
bsf INTCON,INTE ;
return
;-----
sp_ee_30m:
bcf INTCON,INTE ;
movlw .8
movwf schleife
movlw frequenza

```

```

MÄ€r 05, 06 15:20          mc_v1_12.pic          Seite 35/64

movwf FSR
movlw LOW(e30m)
movwf data_ee_addr
sp_ee_30m_01:
movf INDF,W
movwf data_ee_data
PAGESEL PAGE0
call ewrite
PAGESEL PAGE2
incf FSR,F
incf data_ee_addr,F
decfsz schleife,F
goto sp_ee_30m_01
bsf INTCON,INTE ;
return
;-----
sp_ee_20m:
bcf INTCON,INTE ;
movlw .8
movwf schleife
movlw frequenza
movwf FSR
movlw LOW(e20m)
movwf data_ee_addr
sp_ee_20m_01:
movf INDF,W
movwf data_ee_data
PAGESEL PAGE0
call ewrite
PAGESEL PAGE2
incf FSR,F
incf data_ee_addr,F
decfsz schleife,F
goto sp_ee_20m_01
bsf INTCON,INTE ;
return
;-----
sp_ee_17m:
bcf INTCON,INTE ;
movlw .8
movwf schleife
movlw frequenza
movwf FSR
movlw LOW(e17m)
movwf data_ee_addr
sp_ee_17m_01:
movf INDF,W
movwf data_ee_data
PAGESEL PAGE0
call ewrite
PAGESEL PAGE2
incf FSR,F
incf data_ee_addr,F
decfsz schleife,F
goto sp_ee_17m_01
bsf INTCON,INTE ;
return
;-----
sp_ee_gen:
bcf INTCON,INTE ;
movlw .8
movwf schleife
movlw frequenza
movwf FSR
movlw LOW(egen)
movwf data_ee_addr
sp_ee_gen_01:
movf INDF,W
movwf data_ee_data
PAGESEL PAGE0
call ewrite
PAGESEL PAGE2
incf FSR,F
incf data_ee_addr,F
decfsz schleife,F
goto sp_ee_gen_01
bsf INTCON,INTE ;
return
;-----
umsch_80m:

```

```

MÄ€r 05, 06 15:20          mc_v1_12.pic          Seite 36/64

bcf INTCON,GIE ;
movlw .8
movwf schleife
movlw frequenza
movwf FSR
movlw LOW(e80m)
movwf data_ee_addr
umsch_80m_01:
PAGESEL PAGE0
call eread
PAGESEL PAGE2
movwf INDF
incf FSR,F
incf data_ee_addr,F
decfsz schleife,F
goto umsch_80m_01
bsf INTCON,GIE ;
bsf bddsneu
bsf bzfadd ;ZF addieren
call TP40_30m
movlw .2
movwf step
return
;-----
umsch_40m:
bcf INTCON,GIE ;
movlw .8
movwf schleife
movlw frequenza
movwf FSR
movlw LOW(e40m)
movwf data_ee_addr
umsch_40m_01:
PAGESEL PAGE0
call eread
PAGESEL PAGE2
movwf INDF
incf FSR,F
incf data_ee_addr,F
decfsz schleife,F
goto umsch_40m_01
bsf INTCON,GIE ;
bsf bddsneu
bsf bzfadd ;ZF addieren
call TP40_30m
bcf bgen ;kein Radioempfang
movlw .2
movwf step
return
;-----
umsch_30m:
bcf INTCON,GIE ;
movlw .8
movwf schleife
movlw frequenza
movwf FSR
movlw LOW(e30m)
movwf data_ee_addr
umsch_30m_01:
PAGESEL PAGE0
call eread
PAGESEL PAGE2
movwf INDF
incf FSR,F
incf data_ee_addr,F
decfsz schleife,F
goto umsch_30m_01
bsf INTCON,GIE ;
bsf bddsneu
bcf bzfadd ;ZF subdrahieren
call TP40_30m
bcf bgen ;kein Radioempfang
movlw .2
movwf step
return
;-----
umsch_20m:
bcf INTCON,GIE ;
movlw .8
movwf schleife ;8 Bytes laden

```

MÄr 05, 06 15:20

mc_v1_12.pic

Seite 37/64

```

movlw   frequenza
movwf   FSR
movlw   LOW(e20m)      ;aus Eeprom
movwf   data_ee_addr  ;in VFOa und VFOb
umsch_20m_01:
PAGESEL PAGE0        ;SCHLEIFE(1)
call    eread         ; ein Byte lesen
PAGESEL PAGE2
movwf   INDF          ; in Ram speichern
incf    FSR, F        ; Ramaddr+1
incf    data_ee_addr, F ; eepromaddr+1
decfsz  schleife, F  ;
goto    umsch_20m_01 ;ENDE(1) schleife=0
bsf     INTCON, GIE  ;
bsf     bddsneu      ;DDS neu berechnen
bcf     bzfadd       ;ZF subtrahieren
call    TP20_17m     ;Tiefpass schalten
bcf     bgen         ;kein Radioempfang
movlw   .2
movwf   step
return

;-----
umsch_17m:
bcf     INTCON, GIE  ;
movlw   .8
movwf   schleife
movlw   frequenza
movwf   FSR
movlw   LOW(e17m)
movwf   data_ee_addr
umsch_17m_01:
PAGESEL PAGE0
call    eread
PAGESEL PAGE2
movwf   INDF
incf    FSR, F
incf    data_ee_addr, F
decfsz  schleife, F
goto    umsch_17m_01
bsf     INTCON, GIE  ;
bsf     bddsneu
bcf     bzfadd       ;ZF subtrahieren
call    TP20_17m
bcf     bgen         ;kein Radioempfang
movlw   .2
movwf   step
return

;-----
umsch_gen:
bcf     INTCON, GIE  ;
movlw   .8
movwf   schleife
movlw   frequenza
movwf   FSR
movlw   LOW(egen)
movwf   data_ee_addr
umsch_gen_01:
PAGESEL PAGE0
call    eread
PAGESEL PAGE2
movwf   INDF
incf    FSR, F
incf    data_ee_addr, F
decfsz  schleife, F
goto    umsch_gen_01
bsf     INTCON, GIE  ;
bsf     bddsneu
bcf     bzfadd       ;ZF subtrahieren
call    TP20_17m
bsf     bgen         ;Radioempfang
bcf     brit         ;rit aus
bcf     bxit         ;xit aus
return

;-----
maxebene   equ   .16

Menuauswertung:
clrf    impulse      ;Impulse loeschen
PAGESEL PAGE0
call    LCDDisplayClear ;LCD loeschen

```

Sonntag Mär 05, 2006

mc_v1_12.pic

MÄr 05, 06 15:20

mc_v1_12.pic

Seite 38/64

```

PAGESEL PAGE2
mausw01:
call    P2tastaturstatus;SCHLEIFE (1)
SKPNC
goto    mausw01      ;ENDE (1) Taste nicht gedrueckt
movf    ebene, W     ;Anzeige des
call    menuanzeige  ;Menupunktes
;SCHLEIFE (2)
mausw02:
movf    impulse, W   ; ? Impulse angefallen
SKPNZ
goto    mausw04      ;
PAGESEL PAGE0
call    zslsetzen2sek ;
PAGESEL PAGE2
btfsz   impulse, 7   ; 1 ? Vorwaertz
goto    mausw05
incf    ebene, F     ; 1 ebene + 1
movf    ebene, W
xorlw   maxebene+1  ; ? max erreicht
SKPNZ
clrf    ebene       ; 1 von vorn beginnen
goto    mausw06

mausw05:
decf    ebene, F     ; 0 ebene - 1
movf    ebene, W
xorlw   Offh        ; ? unterlauf
SKPZ
goto    mausw06
movlw   maxebene    ; 1 von oben beginnen
movwf   ebene

mausw06:
clrf    impulse      ; Impulse loeschen
PAGESEL PAGE0
call    LCDDisplayClear ; LCD loeschen
PAGESEL PAGE2
movf    ebene, W     ; Menue neu
call    menuanzeige  ; anzeigen
goto    mausw02     ;ENDE (2) keine Impulse

mausw04:
call    P2tastaturstatus;? Taste gedrueckt
SKPNC
goto    mausw07
btfsz   punktpin    ;0 ? Punkt gedrueckt
goto    mausw08
bsf     tonenable    ; 1 Mithoerton ein
LDK1    step, 1     ; Step 10Hz
bsf     blcdneu      ; LCD muss neu eingestellt werden
PAGESEL PAGE0
LCDStr  text17      ; Text anzeigen "spot"
PAGESEL PAGE2

mausw09:
; SCHLEIFE (3)
btfsz   punktpin    ; Punkt gedrueckt
goto    mausw09
call    P2t100mSek  ;
btfsz   punktpin    ; Entprellung
goto    mausw09     ; ENDE (3) keinen Punkt
goto    mausw11

mausw08:
btfsz   strichpin   ; ? Strich gedrueckt
goto    mausw02
PAGESEL PAGE0
LCDStr  text18      ; "tune"
PAGESEL PAGE2

mausw10:
; SCHLEIFE (4)
btfsz   strichpin   ; Strich gedrueckt
goto    mausw10
call    P2t100mSek  ; Entprellung
btfsz   strichpin   ; Strich gedrueckt
goto    mausw10     ; ENDE (4) keinen Strich
PAGESEL PAGE3
call    Dauerton    ; Dauerstrich ausgeben
PAGESEL PAGE2
goto    mausw11

mausw07:
clrf    impulse      ;1 keine Impulse
PAGESEL PAGE0
call    quittungston ; Quittungston
PAGESEL PAGE2
call    quittungston ; Quittung

mausw03:

```

19/32

```

MÄrz 05, 06 15:20          mc_v1_12.pic          Seite 39/64
call    P2tastaturstatus; SCHLEIFE (5)
SKPNC   ;
goto    mausw03            ; ENDE (5) keine Taste gedruickt
movf    ebene,W           ;
call    menuausfuehren    ; Menupunkt ausfuehren
mausw11:
BANKSEL bank1
movlw   0ffh              ;
movwf   ubattmerk+1      ;neuanzeige batt simulieren
movwf   agcmerk+2        ;neuanzeige agc simulieren
BANKSEL bank0
return
;-----
mband1:
call    bandspeichern
movlw   .0
movwf   band
call    umsch_40m
goto    mbreak
;-----
mband2:
call    bandspeichern
movlw   .1
movwf   band
call    umsch_30m
goto    mbreak
;-----
mband3:
call    bandspeichern
movlw   .2
movwf   band
call    umsch_20m
goto    mbreak
;-----
mband4:
call    bandspeichern
movlw   .3
movwf   band
call    umsch_17m
goto    mbreak
;-----
mband5:
call    bandspeichern
movlw   .4
movwf   band
call    umsch_gen
goto    mbreak
;-----
mflagsp:
movf    flag1,W
movwf   data_ee_data
movlw   low(eflag1)
movwf   data_ee_addr
PAGESEL PAGE0
call    ewrite
PAGESEL PAGE2
;-----
mbreak:
PAGESEL PAGE0
call    LCDDisplayClear
PAGESEL PAGE2
bsf     blcdneu
bsf     bddsneu
BANKSEL bank1
movlw   0ffh
movwf   ubattmerk+1      ;neuanzeige batt simulieren
movwf   agcmerk+2        ;neuanzeige agc simulieren
BANKSEL bank0
return
;-----
mvfoab:
btfss   bvfo
goto    mvfoab1
bcf     bvfo
goto    mvfoab2
mvfoab1:
bsf     bvfo
mvfoab2:
goto    mbreak
;-----

```

```

MÄrz 05, 06 15:20          mc_v1_12.pic          Seite 40/64
mrit:
btfsc   bgen              ;? radio
goto    mbreak            ;1 --> break
bcf     bxit              ;xit aus
btfss   brit              ;? rit on
goto    mrit1
bcf     brit              ;1 rit aus
btfsc   bvfo              ; ? VFOa aktiv
goto    mrit4
call    vfob2vfoa         ; 1 VFOa = VFOb(tx-frequenz)
goto    mrit2
mrit4:
call    vfoa2vfob        ; 1 VFOb = VFOa(tx-frequenz)
goto    mrit2
mrit1:
bsf     brit              ;0 rit ein
btfss   bvfo              ; ? VFOb aktiv
goto    mrit3
call    vfob2vfoa         ; 1 VFOb = VFOa
goto    mrit2
mrit3:
call    vfoa2vfob        ; 0 VFOa = VFOb
mrit2:
goto    mbreak
;-----
mkeyer:
btfss   bkeyer            ;? Keyer on
goto    mkeyer1
bcf     bkeyer            ;1 Keyer ausschalten
goto    mkeyer2
mkeyer1:
bsf     bkeyer            ;0 Keyer einschalten
mkeyer2:
goto    mflagsp          ;aenderung in Eeprom speichern
;-----
mlight:
btfsc   blight            ;? Hintergrundbeleuchtung aus
goto    mlight1
bsf     light             ;1 Bel ein
bsf     blight            ; merkbit ein
bcf     blightauto        ; Bel-auto aus
goto    mlight2          ; Daten in Eeprom speichern
mlight1:
bcf     light             ;0 Bel aus
bcf     blight            ; merkbit aus
bsf     blightauto        ; Bel-auto ein
mlight2:
goto    mflagsp          ; Daten in Eeprom speichern
;-----
mmemread:
PAGESEL PAGE3
goto    memread
mmemread01:
goto    mbreak
;-----
mmemwrite:
PAGESEL PAGE3
goto    memwrite
mmemwrite01:
goto    mbreak
;-----
mtune:
PAGESEL PAGE3
goto    Dauerton
mtune01:
goto    mbreak
;-----
mxit1k:
btfsc   bgen              ;? radio
goto    mbreak            ;1 --> break
call    xit1k
goto    mbreak
mxit2k:
btfsc   bgen              ;? radio
goto    mbreak            ;1 --> break
call    xit2k
goto    mbreak
;-----
mlightauto:
btfss   blightauto

```

```

MÄrz 05, 06 15:20          mc_v1_12.pic          Seite 41/64
goto    mlightautol
bcf     blightauto
bcf     blight
bcf     light
goto    mlightauto2
mflightautol:
PAGESEL PAGE0
call    zslsetzen2sek
PAGESEL PAGE2
bsf     blightauto
bsf     light
bcf     blight
mflightauto2:
goto    mflagssp
;-----
smbreak:
PAGESEL PAGE0
call    LCDDisplayClear
goto    start
;-----
smmton:
PAGESEL PAGE3
goto    P3smmton
smmton99:
return
;-----
smdefault:
PAGESEL PAGE3
goto    P3smdefault
;-----
smddskonst:
movlw  low(edds)
bsf     bdds10mhz      ;DDS-Ausgabe 10MHz aktivieren
goto    smedit_hex
;-----
smzf:
bsf     tonenable
movlw  low(ezf)
bsf     bzfdirekt      ;ZF-Direktausgabe aktivieren
goto    smedit_hex_bcd
;-----
smrxdelay:
movlw  .3
movwf  temp+5
movlw  low(erxdelay)
clrf   temp+8
bsf    temp+8,2
CLR4   temp+1
goto   smedit
;-----
;FUNKTION      editieren von 4-Stelligen HEX oder BCD-werten
;EINGANG      temp+7 Adresse im Eeprom
;              temp+8,0 Bit 1 = BCD; Bit 0 = HEX;
;              temp+8,1 BIT 1 = wandlung HEX->BCD und zurueck
;              temp+8,2 BIT 1 = nur 1 Byte bearbeiten
;AUSGANG      temp = 1 4 Byte werden in Eeprom zurueckgespeichert
;              temp = 2 keine Speicherung

smedit_hex:
clrf   temp+8
goto   smedit
smedit_bcd:
clrf   temp+8
bsf    temp+8,0
goto   smedit
smedit_hex_bcd:
clrf   temp+8
bsf    temp+8,0
bsf    temp+8,1
smedit:
movwf  temp+7
movwf  data_ee_addr
movlw  .1
btfs   temp+8,2
movlw  .4
movwf  schleife
clrf   temp
movlw  temp+1
movwf  FSR
PAGESEL PAGE0

```

```

MÄrz 05, 06 15:20          mc_v1_12.pic          Seite 42/64
call    leread
PAGESEL PAGE2
btfs   temp+8,1
call    hex2bcd
btfs   temp+8,2
clrf   temp+5
PAGESEL PAGE3
call    SANzeige
PAGESEL PAGE2
clrf   impulse
smedit01:
btfs   taste          ;SCHLEIFE(1)
goto   smedit03
smedit04:
call    P2tastaturstatus; SCHLEIFE(2)
SKPNC
goto   smedit04      ; ENDE(2) Taste nicht gedrueckt
movf   temp,W        ; ? temp <> 0
SKPNZ
goto   smedit06
movf   temp,W        ; 1 ? temp = 1
xorlw  .1
SKPZ
goto   smedit07
btfs   temp+8,1      ; 1 ? bit 1 gesetzt
call   bcd2hex       ; 1 Wandlung bcd -> hex
movlw  .1
btfs   temp+8,2      ; ? bit 2 gesetzt
movlw  .4            ; 1 4 Byte bearbeite
movwf  schleife     ; Schleife laden
movlw  temp+1
movwf  FSR
movf   temp+7,W
movwf  data_ee_addr
PAGESEL PAGE0
call   lewrite
PAGESEL PAGE2
goto   smedit09
smedit07:
movf   temp,W
xorlw  .2
SKPZ
goto   smedit08
goto   smedit09
smedit08:
smedit06:
incf   temp+5,F
movf   temp+5,W
xorlw  .5
SKPNZ
clrf   temp+5
PAGESEL PAGE3
call   SANzeige
PAGESEL PAGE2
smedit03:
movf   impulse,F
SKPNZ
goto   smedit01
movlw  temp+4
movwf  FSR
movf   temp+5,W
subwf  FSR
btfs   impulse,7
goto   smedit10
movlw  .1
addwf  INDF,F
btfs   temp+8,0
goto   smedit10
;dezimalkorrektur
movf   INDF,W
movwf  temp+10
movlw  6
addwf  temp+10,W
btfs   STATUS,DC    ;? BCD korrektur low durchfuehren
goto   smedit12
movwf  INDF         ;1 ja
smedit12:
movf   INDF,W
movwf  temp+10
movlw  60H         ;? BCD korrektur high durchfuehren

```

```

MÄ€r 05, 06 15:20          mc_v1_12.pic          Seite 43/64
addwf    temp+10,W
btfs    STATUS,C
goto    smedit13
movwf    INDF          ;! ja
smedit13:
smedit10:
    btfs    impulse,7
    goto    smedit11
    movlw    .1
    subwf    INDF,F
    btfs    temp+8,0
    goto    smedit11
;dezimalkorrektur
    rlf    temp+9,F          ;Carry merken
    btfs    STATUS,DC
    goto    smedit14
    movlw    6
    subwf    INDF,F
smedit14:
    btfs    temp+9,0
    goto    smedit15
    movlw    60H
    subwf    INDF,F
smedit15:
smedit11:
    movf    temp,W
    xorlw    .3
    SKPNZ
    clrf    temp
    movf    temp,W
    xorlw    0fh
    SKPZ
    goto    smedit05
    movlw    .2
    movwf    temp
smedit05:
    PAGESEL PAGE3
    call    SAnzeige
    PAGESEL PAGE2
    clrf    impulse
    goto    smedit01          ;
smedit09:
    bcf    tonenable          ;Mithoerton loeschen
    return
;-----
smtbreak:
    PAGESEL PAGE0
    LCDStr stbreak
    PAGESEL PAGE2
    return
;-----
smtddskonst:
    PAGESEL PAGE0
    LCDStr stddskonst
    PAGESEL PAGE2
    return
;-----
smtzf:
    PAGESEL PAGE0
    LCDStr stzf
    PAGESEL PAGE2
    return
;-----
smtsmeter:
    PAGESEL PAGE0
    LCDStr stsmeter
    PAGESEL PAGE2
    return
;-----
smtrxdelay:
    PAGESEL PAGE0
    LCDStr strxdelay
    PAGESEL PAGE2
    return
;-----
smtmton:
    PAGESEL PAGE0
    LCDStr stmton
    BANKSEL tonhoehe
    movf    tonhoehe,W

```

```

MÄ€r 05, 06 15:20          mc_v1_12.pic          Seite 44/64
    BANKSEL 0
    call    LCDHEX
    PAGESEL PAGE2
    return
;-----
smtdefault:
    PAGESEL PAGE0
    LCDStr stdefault
    PAGESEL PAGE2
    return
;-----
mton:
    PAGESEL PAGE0
    LCDStr texton
    PAGESEL PAGE2
    return
;-----
mtoff:
    PAGESEL PAGE0
    LCDStr textoff
    PAGESEL PAGE2
    return
;-----
mtband1:
    PAGESEL PAGE0
    LCDStr tband1
    LCDStr text40m
    PAGESEL PAGE2
    return
;-----
mtband2:
    PAGESEL PAGE0
    LCDStr tband2
    LCDStr text30m
    PAGESEL PAGE2
    return
;-----
mtband3:
    PAGESEL PAGE0
    LCDStr tband3
    LCDStr text20m
    PAGESEL PAGE2
    return
;-----
mtband4:
    PAGESEL PAGE0
    LCDStr tband4
    LCDStr text17m
    PAGESEL PAGE2
    return
;-----
mtband5:
    PAGESEL PAGE0
    LCDStr tband5
    LCDStr textgen
    PAGESEL PAGE2
    return
;-----
mtbreak:
    PAGESEL PAGE0
    LCDStr tbreak
    PAGESEL PAGE2
    return
;-----
mtsetup:
    PAGESEL PAGE0
    LCDStr tsetup
    PAGESEL PAGE2
    return
;-----
mtkeyer:
    btfs    bkeyer
    goto    mtkeyer1
    call    mtoff
    goto    mtkeyer2
mtkeyer1:
    call    mton
mtkeyer2:
    PAGESEL PAGE0
    LCDStr tkeyer

```

MÄr 05, 06 15:20

mc_v1_12.pic

Seite 45/64

```

PAGESEL PAGE2
return
;-----
mtrit:
    btfss    brit
    goto    mtrit1
    call    mtoff
    goto    mtrit2
mtrit1:
    call    mton
mtrit2:
    PAGESEL PAGE0
    LCDStr  trit
    PAGESEL PAGE2
    return
;-----
mtvfoab:
    PAGESEL PAGE0
    LCDStr  tvfoab
    PAGESEL PAGE2
    return
;-----
mtlight:
    btfss    blight
    goto    mtlight1
    call    mtoff
    goto    mtlight2
mtlight1:
    call    mton
mtlight2:
    PAGESEL PAGE0
    LCDStr  tlight
    PAGESEL PAGE2
    return
;-----
mtlightauto:
    btfss    blightauto
    goto    mtlightauto1
    call    mtoff
    goto    mtlightauto2
mtlightauto1:
    call    mton
mtlightauto2:
    PAGESEL PAGE0
    LCDStr  tlightauto
    PAGESEL PAGE2
    return
;-----
mttune:
    PAGESEL PAGE0
    LCDStr  ttune
    PAGESEL PAGE2
    return
;-----
mtmemread:
    PAGESEL PAGE0
    LCDStr  tmemread
    PAGESEL PAGE2
    return
;-----
mtmemwrite:
    PAGESEL PAGE0
    LCDStr  tmemwrite
    PAGESEL PAGE2
    return
;-----
mtxit1k:
    btfss    bxit
    goto    mtxit1k1
    call    mtoff
    goto    mtxit1k2
mtxit1k1:
    call    mton
mtxit1k2:
    PAGESEL PAGE0
    LCDStr  txit1k
    PAGESEL PAGE2
    return
;-----
mtxit2k:

```

MÄr 05, 06 15:20

mc_v1_12.pic

Seite 46/64

```

    btfss    bxit
    goto    mtxit2k1
    call    mtoff
    goto    mtxit2k2
mtxit2k1:
    call    mton
mtxit2k2:
    PAGESEL PAGE0
    LCDStr  txit2k
    PAGESEL PAGE2
    return
;=====
AnzeigeAGC:
    PAGESEL PAGE0
    call    Uagc ;AGC-Spannung messen
    PAGESEL PAGE2
    call    getswert ;swert errechnen
    BANKSEL bank1
    movwf   agcmerk+1 ;swert merken
    subwf   agcmerk,W ;vergleichen mit dem letzten wert
    BANKSEL bank0
    SKPNC
    goto    AAGC01
    movlw   .100 ;1 Timer 1 Sek neu starten
    movwf   zs2
    BANKSEL bank1
    movf   agcmerk+1,W ; neuer wert wird groesster wert
    movwf   agcmerk
    BANKSEL bank0
    goto    AAGC02
AAGC01:
    movf   zs2,F ;0 ? Timer 1 Sek abgelaufen
    SKPZ
    goto    AAGC02
    movf   zs4,F ; 1 ? timer 500 mSek abgelaufen
    SKPZ
    goto    AAGC04
    BANKSEL bank1
    movf   agcmerk,F ; 1 ? AGC != 0
    SKPZ
    decf   agcmerk,F ; 1 AGC -- 1
    BANKSEL bank0
    movlw   .50 ; timer 500 mSek neu starten
    movwf   zs4
AAGC04:
AAGC02:
    BANKSEL bank1
    movf   agcmerk,W ;? letzter angezeigte wert != neuester wert
    subwf   agcmerk+2,W
    BANKSEL bank0
    SKPNZ
    goto    AAGC03 ;0 break --> zum Ende
    PAGESEL PAGE0 ;1 AGC neu anzeigen
    movlw   .14 | 40h
    call    LCDPos
    movlw   'S'
    call    LCDChar
    PAGESEL PAGE2
    BANKSEL bank1
    movf   agcmerk,W
    movwf   agcmerk+2
    BANKSEL bank0
    call    uagc2s
    PAGESEL PAGE0
    call    LCDChar
    PAGESEL PAGE2
AAGC03:
    return
;=====
AnzeigeBatt:
    PAGESEL PAGE0
    call    UBatt ;Batteriespannung messen
    call    ubatt_mul ;umrechnen
    PAGESEL PAGE2
    BANKSEL bank1
    movwf   ubattmerk
    subwf   ubattmerk+1,W ;? alter wert != neuer wert
    BANKSEL bank0
    SKPNZ
    goto    az03 ;0 --> break zum ende

```

```

MÄr 05, 06 15:20          mc_v1_12.pic          Seite 47/64
PAGESEL PAGE0             ;1 neu anzeigen
movlw .11                  ; pos in LCD
call LCDPos                ;
BANKSEL bank1             ;
movf ubattmerk,W          ; neuen wert merken
movwf ubattmerk+1         ;
BANKSEL bank0             ;
call HEX2BCD              ; in BCD wandeln
PAGESEL PAGE2             ;
movf temp+7,W             ;
andlw 0fh                 ; ? erste stelle 0
SKPZ                      ;
goto az01                 ;
PAGESEL PAGE0             ; 1 Leerzeichen anstelle 0
call LCDSpace             ;
PAGESEL PAGE2             ;
goto az02                 ;
az01:
movf temp+7,W             ; 0 wert anzeigen
PAGESEL PAGE0             ;
call BCDToASCII           ;
call LCDChar              ;
PAGESEL PAGE2             ;
az02:
PAGESEL PAGE0             ;
movf temp+6,W             ; naechste stelle
movwf LCDByte2            ;
swapf LCDByte2,W          ;
call BCDToASCII           ;
call LCDChar              ;
movlw '.'                 ; Komma
call LCDChar              ;
movf LCDByte2,W           ; naechste stelle
call BCDToASCII           ;
call LCDChar              ;
movlw 'v'                 ; "v"
call LCDChar              ;
PAGESEL PAGE2             ;
az03:
return                    ;fertig
;=====
vfoa2vfob:
LD4 frequenzb,frequenza
return
;=====
vfob2vfoa:
LD4 frequenza,frequenzb
return
;=====
xitlk:
btfsc bxit                ;? VFob aktiv
goto xitlk03              ;
btfsf bvfo                 ;
goto xitlk01              ;
call vfob2vfoa            ;1 VFOa=VFOb
goto xitlk02              ;
xitlk01:
call vfoa2vfob           ;0 VFOb=VFOa
xitlk02:
bcf brit                   ;rit aus
bsf bxit                   ;xit ein
movlw temp                ;
movwf pointer2            ;
movlw frequenza           ;
btfsf bvfo                 ;
movlw frequenzb           ;
movwf pointer1            ;
PAGESEL PAGE0             ;
call cleartemp4           ;
movlw 10h                 ;
movwf temp+1              ;
call bcdadd4              ;
PAGESEL PAGE2             ;
goto xitlk04              ;
xitlk03:
bcf bxit                   ;
xitlk04:
return
;=====
xit2k:

```

```

MÄr 05, 06 15:20          mc_v1_12.pic          Seite 48/64
btfsc bxit                ;
goto xit2k03              ;
btfsf bvfo                 ;
goto xit2k01              ;
call vfob2vfoa            ;
goto xit2k02              ;
xit2k01:
call vfoa2vfob           ;
xit2k02:
bcf brit                   ;
bsf bxit                   ;
movlw temp                ;
movwf pointer2            ;
movlw frequenza           ;
btfsf bvfo                 ;
movlw frequenzb           ;
movwf pointer1            ;
PAGESEL PAGE0             ;
call cleartemp4           ;
movlw 20h                 ;
movwf temp+1              ;
call bcdadd4              ;
PAGESEL PAGE2             ;
goto xit2k04              ;
xit2k03:
bcf bxit                   ;
xit2k04:
return
;=====
;Funktion Ausrechnen SWR aus uvor und urueck
;Register temp, temp+1, temp+2, temp+3, temp+4, temp+5, uvor, urueck, swr x3
;Eingang uvor, urueck
;Ausgang swr = Bargraphlaenge, swr+1-swr+2 SWR in Dezimalform
getswr:
PAGESEL PAGE0             ;
call cleartemp            ;temp loeschen
PAGESEL PAGE2             ;
movf uvor,W               ;temp = Uvor - Urueck
btfsc STATUS,Z           ;? test auf Uvor = 0
goto getswr4              ;1 break zur extremwertkorrektur
movwf temp                ;
movf urueck,W            ;
movwf temp+2              ;
SUB2 temp,temp+2          ;
btfsf STATUS,C           ;? Uvor < Urueck
goto getswr3              ;1 break zur Extremwertkorrektur
movf temp,F               ;? Uvor - Urueck = 0
btfsc STATUS,Z           ;
goto getswr3              ;1 break zur Extremwertkorrektur
movf uvor,W               ;
movwf temp+4              ;
ADD2 temp+2,temp+4        ;temp+2 = Uvor + Urueck
clrf temp+5               ;
decf temp+5,F             ;
getswr1:
incf temp+5,F             ;SCHLEIFE(1)
; Ziffer vor dem Komma durch Subtraktion
SUB2 temp+2,temp          ; errechnen
btfsc STATUS,C           ;
goto getswr1              ;ENDE(1) temp+2 < 0 in temp+5 = Ergebnis vor dem Komma
ADD2 temp+2,temp          ;0 wurde unterschritten wieder korrigieren
;Rest mit 10 multiplizieren
BMUL10 temp+2             ;
clrf temp+4               ;Nachkommastelle ermitteln
;Schleifenvorbereitung
decf temp+4,F             ;SCHLEIFE(2)
getswr2:
incf temp+4,F             ; und nachkommastelle errechnen
SUB2 temp+2,temp          ;
btfsc STATUS,C           ;
goto getswr2              ;ENDE(1) temp+2 < 0 in temp+4 = Ergebnis
ADD2 temp+2,temp          ;Korrektur wenn < 0
movlw 0ah                 ;? test ob SWR > 9,9
subwf temp+5,W            ;
btfsf STATUS,C           ;
goto getswr5              ;
getswr3:
movlw 9                    ;1 Anzeige auf 9,9 einstellen
movwf temp+4              ;
movwf temp+5              ;
goto getswr5              ;
getswr4:

```


MÄr 05, 06 15:20	mc_v1_12.pic	Seite 49/64
	clrf temp+4 ;Extremwert 0,0 einstellen	
	clrf temp+5	
	clrf temp+3 ;BarGraph = 0	
	goto getswr8	
getswr5:	clrf temp+3 ;BarGraph ermitteln	
	movlw 2 ;BarGraph = 0	
	subwf temp+5,W ;? SWR < 2.0	
	btfs STATUS,C ;	
	goto getswr6 ;	
	decf temp+5,W ;0 BarGraph = 9 + (SWR(vorm Komma) - 1)	
	movwf temp+3 ;	
	movlw 9 ;	
	addwf temp+3,F ;	
	goto getswr7 ;	
getswr6:	movf temp+4,W ;! BarGraph = SWR(Nachkomma)	
	movwf temp+3 ;	
getswr7:	movlw .18 ;? BarGraph > 18	
	subwf temp+3,W ;	
	btfs STATUS,C ;	
	decf temp+3,F ;	
getswr8:	movf temp+3,W	
	BANKSEL bank1	
	movwf swr ;Bargraph	
	BANKSEL bank0	
	movf temp+4,W	
	BANKSEL bank1	
	movwf swr+1 ;SWR	
	BANKSEL bank0	
	movf temp+5,W	
	BANKSEL bank1	
	movwf swr+2	
	BANKSEL bank0	
	return	
;=====		
mscan:	btfsc blightauto	
	bcf light	
	LDK1 step,2	
	bcf bvfo ;VFO A	
	call AnzeigeAGC ;S_Wert anzeigen	
	PAGESEL PAGE0	
	LCDStr text6 ;"scanning" auf Zeile 2	
	PAGESEL PAGE2	
	BANKSEL bank1	
	LD2 agc, wert ;Startwert einstellen	
	BANKSEL bank0	
mscan01:	BANKSEL bank1	
	clrf anst ;Anstieg = 0	
	LD4 utemp,fscan ;Startfrequenz in fscan	
	BANKSEL bank0	
	LD4 frequenza,utemp ;Startfrequenz holen	
scan01:	PAGESEL PAGE0 ;SCHLEIFE(1)	
	call LCDAnzeigeZ1 ; LCD neu anzeigen Zeile1	
	call stepadd ; step addieren	
	call ddsbinausrechnen: DDS neu ausrechnen	
	call rx_to_dds ; Empfangsfrequenz laden	
	PAGESEL PAGE2	
	call AnzeigeAGC ; S_Wert anzeigen	
	BANKSEL bank1	
	CMP2 agc, wert ; ? wert > agc	
	BANKSEL bank0	
	SKPNC	
	goto scan02	
	BANKSEL bank1	
	incf anst,F ; 1 anstieg+1	
	BANKSEL bank0	
	goto scan03	
scan02:	BANKSEL bank1 ; 0 anstieg=0	
	clrf anst	
	BANKSEL bank0	
scan03:	BANKSEL bank1	
	CMPK1 anst,2 ; ? anstieg > 2	

MÄr 05, 06 15:20	mc_v1_12.pic	Seite 50/64
	BANKSEL bank0	
	SKPC	
	goto scan04	
	btfs blightauto ; 1 ? Lichtautomatik	
	bsf light ; 1 Licht ein	
	movlw .40	
	movwf schleife+2	
scan05:	call AnzeigeAGC ; SCHLEIFE(2)	
	BANKSEL bank1	
	LD2 agc, wert ; s-wert merken	
	BANKSEL bank0	
	PAGESEL PAGE0	
	call t100mSek ;	
	PAGESEL PAGE2	
	btfs punktpin ; ? Punktpaddle	
	goto scan98 ; 1 --> break	
	btfs bkeyer ; ? keyer ein	
	goto scan05_1 ;	
	btfs strichpin ; 1 ? Strichpaddle	
	goto scan98 ; 1 --> break	
scan05_1:	btfs taste ; ? Taste	
	goto scan98 ; 1 --> break	
	movf impulse,F ; ? Drehgeber-Impulse	
	btfs STATUS,Z ;	
	goto scan98 ; 1 --> break	
	decfsz schleife+2,F ; ENDE(2) nach 4 Sek oder break	
	goto scan05	
	BANKSEL bank1	
	clrf anst ; anstieg = 0	
	BANKSEL bank0	
	btfs blightauto ; ? Lichtautomatik	
	bcf light ; 1 Licht aus	
scan04:	BANKSEL bank1	
	LD2 agc, wert ; s-wert merken	
	BANKSEL bank0	
	btfs punktpin ; ? Punktpaddle	
	goto scan98 ; 1 --> break	
	btfs bkeyer ; ? keyer ein	
	goto scan04_1 ;	
	btfs strichpin ; 1 ? Strichpaddle	
	goto scan98 ; 1 --> break	
scan04_1:	btfs taste ; ? Taste	
	goto scan98 ; 1 --> break	
	movf impulse,F ; ? Drehgeber-Impulse	
	btfs STATUS,Z ;	
	goto scan98 ; 1 --> break	
	CMP4 frequenza, frequenzb	
	SKPC ;	
	goto scan01 ;ENDE(1) frequenza > frequenzb oder break	
	bsf bddsneu	
	goto mscan01 ;	
scan98:	btfs blightauto ;--> break	
	bsf light ;? Lichtautomatik	
	PAGESEL PAGE0 ;! Licht ein	
	LCDStr text7 ;"break" auf Zeile 2	
	PAGESEL PAGE2	
scan99:	PAGESEL PAGE2 ;SCHLEIFE(3)	
	btfs bkeyer ;	
	goto scan99_1 ;	
	btfs strichpin ; bei Keyer auch Strichpin mit beachten	
	goto scan99 ;	
scan99_1:	btfs punktpin ; bei Handkeyer nur Reaktion	
	goto scan99 ; bei Punkpin und Taste	
	btfs taste ;	
	goto scan99 ;ENDE(3) punkt, strich oder taste nicht gedruickt	
	btfs blightauto ;? Lichtautomatik	
	bcf light ;! Licht aus	
	goto mbreak	
;=====		
msetup:	smaxe bene equ .6	

MÃ€r 05, 06 15:20	mc_v1_12.pic	Seite 51/64
	<pre> clrf ebene clrf impulse PAGESEL PAGE0 call LCDDisplayClear PAGESEL PAGE2 msausw01: call P2tastaturstatus SKPNC goto msausw01 movf ebene,W call smenuanzeige msausw02: movf impulse,W ;? Impulse angefallen SKPNZ goto msausw04 PAGESEL PAGE0 call zslsetzen2sek PAGESEL PAGE2 btfsc impulse,7 ;! ? Vorwaertz goto msausw05 incf ebene,F ; 1 movf ebene,W xorlw smaxeebene+1 ; SKPNZ clrf ebene goto msausw06 msausw05: decf ebene,F movf ebene,W xorlw 0ffh SKPZ goto msausw06 movlw smaxeebene movwf ebene msausw06: PAGESEL PAGE0 call LCDDisplayClear PAGESEL PAGE2 movf ebene,W call smenuanzeige clrf impulse msausw04: call P2tastaturstatus SKPC goto msausw02 msausw03: call P2tastaturstatus SKPNC goto msausw03 movf ebene,W call smenuausfuehren goto msetup ;===== ;Funktion HEX zu BCD wandeln hex2bcd: LD4 tempkonst1, temp+1 movlw .8 * 4 movwf schleife CLR4 temp+1 hextbl1: movlw temp+1 movwf FSR movlw 4 movwf schleife+1 hextb4: movf INDF,W andlw B'00001111' movwf temp+5 movlw .5 subwf temp+5,W btfss STATUS,C goto hextb2 movlw .3 addwf INDF,F hextb2: movf INDF,W andlw B'11110000' movwf temp+5 movlw 50h </pre>	

MÃ€r 05, 06 15:20	mc_v1_12.pic	Seite 52/64
	<pre> subwf temp+5,W btfss STATUS,C goto hextb3 movlw 30h addwf INDF,F hextb3: incf FSR,F decfsz schleife+1 goto hextb4 bcf STATUS,C RL4 tempkonst1 RL4 temp+1 decfsz schleife,F goto hextb1 return ;===== bcd2hex: LD4 tempkonst1, temp+1 CLR4 temp+1 movlw .8 * 4 ;32 bit bcd in bin umwandeln movwf schleife+1 bcd2hex1: bcf STATUS,C RR4 tempkonst1 RR4 temp+1 movlw tempkonst1 movwf FSR movlw 4 movwf schleife bcd2hex3: btfss INDF,7 goto bcd2hex4 movlw 30H subwf INDF,F bcd2hex4: btfss INDF,3 goto bcd2hex5 movlw 3 subwf INDF,F bcd2hex5: incf FSR,F decfsz schleife,F goto bcd2hex3 decfsz schleife+1,F goto bcd2hex1 return ;===== ;Funktion dividieren von 2 2Byte-Werten ;Eingang divisor, dividend ;Ausgang ergebnis, rest im dividend ;Register schleife, divisor:2, dividend:2 divisor equ temp+4 divident equ temp+2 ergebnis equ temp divb2: clrf schleife CLR2 ergebnis divb21: incf schleife clrc btfsc divisor+1,7 goto divb23 RL2 divisor CMP2 dividend,divisor SKPNC goto divb21 CLR2 ergebnis divb22: clrc RR2 divisor divb23: CMP2 dividend,divisor SKPC goto divb24 SUB2 dividend,divisor divb24: RL2 ergebnis decfsz schleife </pre>	

MÄr 05, 06 15:20	mc_v1_12.pic	Seite 53/64
goto	divb22	
return		

;Funktion	Multplikation von 2 Faktoren 2 Byte lang	
;Eingang	faktor1:2, faktor2:2	
;Ausgang	ergebnis:2	
;Register	pointer1, pointer2, schleife, schleife+1, zw, ueberlauf	
faktor1:	equ temp+2	
faktor2:	equ temp+4	
ergebnis:	equ temp	
zw	equ temp+6	
ueberlauf	equ temp+7	
mulb2:		
CLR2	ergebnis	
movlw	.16	;Schleife laden mit 32 bit
movwf	schleife	
mulb21:		
movlw	faktor2	;schleife (1)
movwf	pointer1	; pointer1 = faktor2
movlw	ergebnis	
movwf	pointer2	; pointer2 = faktor1
clrf	ueberlauf	; Ueberlauf loeschen
bcf	STATUS,C	; Carry loeschen
RR2	faktor1	
btfs	STATUS,C	; ? Carry = 1 nach RR
goto	mulb22	
movlw	2	; 1 Schleife laden mit 8
movwf	schleife+1	
mulb25:		
movf	pointer1,W	; schleife (2)
movwf	FSR	; 1 Pointer laden
movf	INDF,W	; wert1 holen
movwf	zw	; zwischenspeichern
movf	pointer2,W	; Pointer2 laden
movwf	FSR	; ins indirekte Register
movf	ueberlauf,W	; gemerketen Ueberlauf holen
clrf	ueberlauf	; und loeschen
addwf	INDF,F	; zum Ergebnis addieren
btfs	STATUS,C	; ? Ueberlauf
goto	mulb24	
movlw	1	; 1 neuen Ueberlauf
movwf	ueberlauf	; merken
mulb24:		
movf	zw,W	; Wert2 = Wert2 + Wert1
addwf	INDF,F	; zum Ergebnis addieren
btfs	STATUS,C	; ? Ueberlauf
goto	mulb23	
movlw	1	; 1 neuen Ueberlauf
movwf	ueberlauf	; merken
mulb23:		
incf	pointer1	
incf	pointer2	
decfsz	schleife+1	; ende (2)
goto	mulb25	
mulb22:		
bcf	STATUS,C	; Carry loeschen
RL2	faktor2	
decfsz	schleife,F	;ende (1) 32 mal durchlaufen
goto	mulb21	
return		

getswert:		
PAGESEL	PAGE0	
call	Uagc	;AGC Spannung messen
PAGESEL	PAGE2	
BANKSEL	bank1	
LD2	utemp,wert	;Wert zum Rechnen holen
BANKSEL	bank0	
LD2	temp+2,utemp	
BANKSEL	bank1	
LD2	utemp,s_konst1	;s_konst1 subtrahieren
BANKSEL	bank0	
LD2	temp+4,utemp	
CMP2	temp+2,temp+4	;? wert > konst1
SKPC		
goto	gsw01	
SUB2	temp+2,temp+4	;1
LDK2	temp+4,0,.10	; Ergebnis mit 10 multiplizieren

MÄr 05, 06 15:20	mc_v1_12.pic	Seite 54/64
call	mulb2	
LD2	temp+2,temp	
BANKSEL	bank1	
LD2	utemp,s_konst2	
BANKSEL	bank0	
LD2	temp+4,utemp	
movf	temp+5,F	; ? s_konst2 = 0
btfs	STATUS,Z	
goto	gsw03	
movf	temp+4,F	
btfs	STATUS,Z	
goto	gsw02	; 1 --> break Ergebnis = 0fh = unendlich
gsw03:		
call	divb2	; 0 Ergebnis durch s_konst2 dividieren
movf	temp,W	
andlw	0xf0	; ? Ergebnis groesser 0fh
SKPZ		
goto	gsw02	
movf	temp,W	
movf	temp+1,F	; 0 ? Ergebnis groesser 0ffh
SKPZ		
gsw02:		
movlw	0x0f	; 1 Ergebnis = 0fh = max
return		
gsw01:		
clrw		;0 Ergebnis = 0 = min
return		
=====		
smsmeter:		
PAGESEL	PAGE3	
call	P3smsmeter	
PAGESEL	PAGE2	
return		
=====		
TP40_30m:		
bcf	relgem	
bcf	rel1	
bcf	rel2	
bsf	rel1	
call	P2t10mSek	
bcf	rel1	
call	P2t1mSek	
bsf	rel2	
call	P2t10mSek	
bcf	rel2	
call	P2t1mSek	
return		

TP20_17m:		
bsf	relgem	
bsf	rel1	
bsf	rel2	
bcf	rel1	
call	P2t10mSek	
bsf	rel1	
call	P2t1mSek	
bcf	rel2	
call	P2t10mSek	
bsf	rel2	
call	P2t1mSek	
bcf	relgem	
bcf	rel1	
bcf	rel2	
return		

P2t100mSek:		
movlw	.100	
goto	P2t01	
P2t1mSek:		
movlw	1	
goto	P2t01	
P2t10mSek:		
movlw	.10	
P2t01:		
PAGESEL	PAGE0	
call	DELAY	
PAGESEL	PAGE2	
return		

P2tastaturstatus:		

MÄr 05, 06 15:20

mc_v1_12.pic

Seite 55/64

```

PAGESEL PAGE0
call tastaturstatus
PAGESEL PAGE2
return
;-----
P2LCDHEX:
PAGESEL PAGE0
call LCDHEX
PAGESEL PAGE2
return
;#####
endepage2:
if endepage2 > 1800h
error "PAGE2 ueberschritten"
endif
org 1800h
PAGE3:
;#####
Sanzeige:
LD4 utemp,temp
BANKSEL bank1
LD4 tempbank1,utemp
BANKSEL bank0
LD4 utemp,temp+4
BANKSEL bank1
LD4 tempbank1+4,utemp
BANKSEL bank0
LD4 utemp,temp+8
BANKSEL bank1
LD4 tempbank1+8,utemp
BANKSEL bank0
btfs bdds10mhz ;? DDS-Ausgabe 10MHz
goto sanz01_3
LD4 ddskonst,temp+1 ;! geaenderte Konstante laden

ifdef europa
LDK4 frequenza,6h,07h,50h,0 ; 6,075 Mhz "Deutsche Welle" in Europa
endif

ifndef europa
LDK4 frequenza,10h,0h,0h,0h ; 10 Mhz Zeitzeichensender in USA
endif

goto sanz01_2
sanz01_3:
btfs bzfdirekt ;? ZF-Ausgabe
goto sanz01_1
LD4 frequenza,temp+1 ;! Frequenz A mit ZF laden
sanz01_2:
bsf bddsohnezf ; keine ZF-Berechnung
bcf bvfo ; VFO A
PAGESEL PAGE0
call ddsbinausrechnen ; DDS neu ausrechnen
call rx_to_dds ; Empfangsfrequenz laden
PAGESEL PAGE3
sanz01_1:
BANKSEL bank1
LD4 utemp,tempbank1
BANKSEL bank0
LD4 temp,utemp
BANKSEL bank1
LD4 utemp,tempbank1+4
BANKSEL bank0
LD4 temp+4,utemp
BANKSEL bank1
LD4 utemp,tempbank1+8
BANKSEL bank0
LD4 temp+8,utemp

movlw temp+4
movwf FSR
movlw .4
movwf schleife
PAGESEL PAGE0
call LCDDisplayClear
movlw B'00001111'
call LCDCOM
PAGESEL PAGE3
sanz01:

```

MÄr 05, 06 15:20

mc_v1_12.pic

Seite 56/64

```

movf INDF,W
PAGESEL PAGE0
call LCDHEX
PAGESEL PAGE3
decf FSR,F
decfsz schleife
goto sanz01
movf temp,W
SKPZ
goto sanz02
PAGESEL PAGE0
LCDStr text12
PAGESEL PAGE3
sanz02:
movf temp,W
xorlw 1
SKPZ
goto sanz03
PAGESEL PAGE0
LCDStr text8
PAGESEL PAGE3
sanz03:
movf temp,W
xorlw 2
SKPZ
goto sanz04
PAGESEL PAGE0
LCDStr text9
PAGESEL PAGE3
sanz04:
LDI temp+6, temp+5
clrc
rlf temp+6,F
incf temp+6,W
iorlw 40h
PAGESEL PAGE0
call LCDPos
PAGESEL PAGE3

return
;-----
P3smsmeter:
PAGESEL PAGE0
LCDStr text10
PAGESEL PAGE3
smsm01:
PAGESEL PAGE0 ;SCHLEIFE(1)
call tastaturstatus ;
PAGESEL PAGE3
SKPNC ;
goto smsm01 ;ENDE(1) Taste nicht mehr gedrueckt
smsm02:
PAGESEL PAGE0 ;SCHLEIFE(2)
call Uagc ; AGC Spannung messen
PAGESEL PAGE3
BANKSEL bank1
LD2 utemp,wert ; Wert holen
BANKSEL bank0
LD2 temp,utemp
movlw 40h
PAGESEL PAGE0
call LCDPos
PAGESEL PAGE3
movf temp+1,W ; und Anzeigen
PAGESEL PAGE0
call LCDHEX
PAGESEL PAGE3
movf temp,W
PAGESEL PAGE0
call LCDHEX
call tastaturstatus
PAGESEL PAGE3
SKPC
goto smsm02 ;ENDE(2) Taste gedrueckt
BANKSEL bank1
LD2 utemp,wert ;Wert holen in Bank0
LD2 s_konst1,wert ;Wert auch in den Ram speichern
BANKSEL bank0
movlw .2
movwf schleife

```

MÄr 05, 06 15:20

mc_v1_12.pic

Seite 57/64

```

movlw   utemp
movwf   FSR
movlw   low(es_konst1) ;s_konst1 in Eeprom speichern
movwf   data_ee_addr
PAGESEL PAGE0
call    lewrite        ;und in den Eeprom speichern
LCDStr  text11
PAGESEL PAGE3
smsm03:
PAGESEL PAGE0
call    tastaturstatus
PAGESEL PAGE3
SKPNC
goto    smsm03        ;ENDE(3) Taste nicht gedrueckt
smsm04:
PAGESEL PAGE0        ;SCHLEIFE(4)
call    Uagc         ; AGC-Spannung messen
PAGESEL PAGE3
BANKSEL bank1
LD2     utemp,wert
BANKSEL bank0
LD2     temp,utemp
movlw   40h
PAGESEL PAGE0
call    LCDPos
PAGESEL PAGE3
movf    temp+1,W
PAGESEL PAGE0
call    LCDHEX        ; und Anzeigen
PAGESEL PAGE3
movf    temp,W
PAGESEL PAGE0
call    LCDHEX
call    tastaturstatus
PAGESEL PAGE3
SKPNC
goto    smsm04        ;ENDE(4) Taste gedrueckt
BANKSEL bank1
LD2     utemp,wert    ;Wert holen
LD2     utemp+2,s_konst1;Konstandel holen
BANKSEL bank0
LD2     temp+2,utemp
LD2     temp+4,utemp+2
SUB2    temp+2,temp+4 ;Konstande2 = Wert - Konstandel
LDK2    temp+4,0,.10
PAGESEL PAGE2
call    mulb2         ;Konstande2 = Konstande2 * 10
PAGESEL PAGE3
LD2     temp+2,temp
LDK2    temp+4,0,.9
PAGESEL PAGE2
call    divb2         ;Konstande2 = Konstande2 / 9
PAGESEL PAGE3
movlw   .2
movwf   schleife
movlw   temp
movwf   FSR
movlw   low(es_konst2)
movwf   data_ee_addr
PAGESEL PAGE0
call    lewrite        ;Konstande2 in Eeprom speichern
PAGESEL PAGE3
return
;-----
memread:
call    eeauswahl     ;Speicherstelle auswaehlen
movf    eeindex,F     ;? index <> 0
SKPNZ
goto    memread01
call    eeaddrrechnung ;1 daten aus eeprom
movwf   data_ee_addr  ; laden
movlw   .9            ; 9 Byte
movwf   schleife
movlw   temp          ; erst einmal
movwf   FSR          ; in temp laden
PAGESEL PAGE0
call    lread
PAGESEL PAGE3
movlw   0fh
subwf   temp+8,W      ;

```

MÄr 05, 06 15:20

mc_v1_12.pic

Seite 58/64

```

btfsc   STATUS,Z      ;
goto    memread99     ; ---> break da FF
movf    temp+8,W      ; Bandinformation laden
PAGESEL PAGE2
call    eebandumsch   ; auf entsprechendes band schalten
PAGESEL PAGE3
LD4     frequenza,temp ; Frequenz A
LD4     frequenzb,temp+4; Frequenz B
LD4     utemp,temp     ; Frequenz A nach Bank1
BANKSEL bank1
LD4     fscan,utemp   ; in fscan speichern (scanstart merken)
BANKSEL bank0
btfss   bscan        ; ? Scannen
goto    memread01
PAGESEL PAGE0
LCDStr  text15        ; 1 text "read + scanning"
PAGESEL PAGE3
memread01:
PAGESEL PAGE0        ;SCHLEIFE(1)
call    tastaturstatus ;
PAGESEL PAGE3
SKPNC
goto    memread01     ;ENDE(1) bis Taste losgelassen
PAGESEL PAGE0
call    LCDDisplayClear ;LCD loeschen
PAGESEL PAGE2        ;weitere funktionen auf Seite 2
btfsc   bscan        ;? Scannen
goto    mscan        ;1 --> scanning
memread99:
PAGESEL PAGE2
goto    mmemread01    ;0 --> normale Returnfunktion
;-----
memwrite:
call    eeauswahl
movf    eeindex,F     ;? index <> 0
SKPNZ
goto    memw03
call    eeaddrrechnung
movwf   data_ee_addr
movlw   .9
movwf   schleife
movlw   frequenza
movwf   FSR
memw02:
movf    INDF,W
movwf   data_ee_data
PAGESEL PAGE0
call    ewrite
PAGESEL PAGE3
incf    FSR,F
incf    data_ee_addr,F
decfsz  schleife,F
goto    memw02
PAGESEL PAGE0
call    LCDDisplayClear
LCDStr  text14
PAGESEL PAGE3
memw01:
PAGESEL PAGE0
call    tastaturstatus
PAGESEL PAGE3
SKPNC
goto    memw01
memw03:
PAGESEL PAGE2
goto    mbreak
;-----
eemaxindex equ .20
eeauswahl:
clrf   impulse
PAGESEL PAGE0
call   LCDDisplayClear
PAGESEL PAGE3
eeausw01:
PAGESEL PAGE0
call   tastaturstatus
PAGESEL PAGE3
SKPNC
goto   eeausw01

```

MÃ€r 05, 06 15:20	mc_v1_12.pic	Seite 59/64
eeausw02:	<pre> movf eeindex,W call eeanzeige movf impulse,W ;? Impulse angefallen SKPNZ goto eeausw04 PAGESEL PAGE0 call zslsetzen2sek PAGESEL PAGE3 btfsc impulse,7 ;1 ? Vorwaertz goto eeausw05 incf eeindex,F ; 1 movf eeindex,W xorlw eemaxindex+1 ; SKPNZ clrf eeindex goto eeausw06 </pre>	
eeausw05:	<pre> decf eeindex,F movf eeindex,W xorlw 0ffh SKPZ goto eeausw06 movlw eemaxindex movwf eeindex </pre>	
eeausw06:	<pre> PAGESEL PAGE0 call LCDDisplayClear PAGESEL PAGE3 movf eeindex,W call eeanzeige clrf impulse </pre>	
eeausw04:	<pre> PAGESEL PAGE0 call tastaturstatus PAGESEL PAGE3 SKPC goto eeausw02 call Tasteeingabe bcf bscan movf tastennummer,W xorlw 2 SKPNZ bsf bscan return </pre>	
eeanzeige:	<pre> movf eeindex,F ;? index <> 0 SKPNZ goto eeanz01 clrw ;1 Frequenzen ausgeben PAGESEL PAGE0 call LCDPos ; Position 0 Zeile1 PAGESEL PAGE3 CLR4 temp+1 ; 4 Zellen loeschen movf eeindex,W ; movwf temp+1 ; PAGESEL PAGE2 call hex2bcd ; index in BCD wandeln PAGESEL PAGE3 movf temp+1,W ; und PAGESEL PAGE0 call LCDHEX ; anzeigen PAGESEL PAGE3 call eeaddrrechnung ; Adressen im Eeprom ausrechnen movwf data_ee_addr ; movlw .9 ; 9 Byte movwf schleife ; vom Eeprom movlw temp ; in temp laden movwf FSR PAGESEL PAGE0 call leread ; Leerzeichen ausgeben call LCDSpace ; Frequenz A call LCDChar ; call LCDSpace ; Leerzeichen movf temp+3,W ; call LCDHEX ; Mhz ausgeben movlw '.' ; komma call LCDChar ; </pre>	

MÃ€r 05, 06 15:20	mc_v1_12.pic	Seite 60/64
eeanz01:	<pre> movf temp+2,W ; Nachkommastellen ausgeben call LCDHEX movf temp+1,W call LCDHEX movf temp,W call LCDHEX movlw 40h ; Zeile 2 ausgeben call LCDPos ; LDI temp,temp+8 ; Bandinformation movf temp,W call LCDHEX ; ausgeben call LCDSpace movlw 'B' ; Frequenz B call LCDChar call LCDSpace movf temp+7,W call LCDHEX movlw '.' call LCDChar movf temp+6,W call LCDHEX movf temp+5,W call LCDHEX movf temp+4,W call LCDHEX PAGESEL PAGE3 goto eeanz02 </pre>	
eeanz02:	<pre> PAGESEL PAGE0 ;0 cancel-anzeige LCDStr text13 PAGESEL PAGE3 return </pre>	
eeaddrrechnung:	<pre> movf eeindex,W movwf schleife+2 movlw emem movwf temp </pre>	
eadr03:	<pre> decfsz schleife+2 goto eadr02 goto eadr01 </pre>	
eadr02:	<pre> movlw .9 addwf temp goto eadr03 </pre>	
eadr01:	<pre> movf temp,W return </pre>	
Tasteeingabe:	<pre> PAGESEL PAGE0 ; call quittungston ; PAGESEL PAGE3 ; clrf impulse ;Impulse loeschen movlw 1 ;Tastennummer 1 voreinstellen movwf tastennummer ; movlw .10 ;Zeitschleife aufbauen 500 mSek movwf schleife ; </pre>	
Tasteeing1:	<pre> PAGESEL PAGE0 ; call Tastegedrueckt ;SCHLEIFE(1) PAGESEL PAGE3 ; SKPC ; goto Tasteeing2 ; break --> Taste nicht gedrueckt Tastennummer 1 PAGESEL PAGE0 ; call t50mSek ; Display unveraendert PAGESEL PAGE3 ; decfsz schleife,F ;ENDE(1) nach 500 msek Taste gedrueckt goto Tasteeing1 ; PAGESEL PAGE0 ; call quittungston ; call t50mSek ; call quittungston ; PAGESEL PAGE3 ; </pre>	
Tasteeing3:	<pre> movlw 2 ;nach 500 mSek Tastennummer 2 movwf tastennummer ; </pre>	

```

MÄ€ 05, 06 15:20          mc_v1_12.pic          Seite 61/64

Tasteeing4:
    PAGESEL PAGE0          ;
    call Tastegedrueckt    ;
    PAGESEL PAGE3          ;
    SKPNC                   ;
    goto Tasteeing4
Tasteeing2:
    return
;=====
P3smmton:
    PAGESEL PAGE0          ;
    call Tastegedrueckt    ;
    PAGESEL PAGE3          ;
    SKPNC                   ;
    goto P3smmton
smmton02:
    bsf tonenable          ;
    PAGESEL PAGE0          ;
    movlw 40h               ;
    call LCDPos             ;
    call UmessKeyerPoti    ;
    movwf temp              ;
    clrc                    ;
    rrf temp,W              ;
    BANKSEL tonhoehe       ;
    movwf tonhoehe         ;
    BANKSEL bank0          ;
    call LCDHEX             ;
    call Tastegedrueckt    ;
    PAGESEL PAGE3          ;
    SKPC                     ;
    goto smmton02
smmton01:
    PAGESEL PAGE0          ;
    call Tastegedrueckt    ;
    PAGESEL PAGE3          ;
    SKPNC                   ;
    goto smmton01
    BANKSEL tonhoehe       ;
    movf tonhoehe,W        ;
    BANKSEL bank0          ;
    movwf data_ee_data     ;
    movlw low(epitch)      ;
    movwf data_ee_addr    ;
    PAGESEL PAGE0          ;
    call ewrite             ;
    PAGESEL PAGE2          ;
    bcf tonenable          ;
    goto smmton99
;=====
seflag1      de    b'00000011'      ;flag1 lightauto=on keyer=on
sedds        de    0b8h,063h,05eh,05h ;DDS konstante
sezf         de    38h,0ffh,4ah,0    ;ZF binar (4,915 MHz)
seband1      de    01h               ;welches Band auf Speicherstelle 1
seband2      de    02h               ;welches Band auf Speicherstelle 2
seband3      de    03h               ;welches Band auf Speicherstelle 3
seband4      de    04h               ;welches Band auf Speicherstelle 4
seband5      de    05h               ;welches Band auf Speicherstelle 4
serxdelay    de    .4                 ;default 5 mSek Verzoegerung TX -> RX
sepitch      de    3dh               ;Tonhoehe

se80m:       de    0,0,56h,03h
              de    0,0,56h,03h
se40m:       de    0,0,03h,07h
              de    0,0,03h,07h
se30m:       de    0,60h,11h,10h
              de    0,60h,11h,10h
se20m:       de    0,0,06h,14h
              de    0,0,06h,14h
se17m:       de    0,60h,09h,18h
              de    0,60h,09h,18h
segen:       de    0,50h,07h,06h
              de    0,50h,08h,06h
ses_konst1:  de    25h,0                ;S-Meter konstante 1
ses_konst2:  de    95h,0                ;S_meter konstante 2
sende:

P3smdefault:
    movlw sende-seflag1      ;Anzahl der Bytes
    movwf temp

```

```

MÄ€ 05, 06 15:20          mc_v1_12.pic          Seite 62/64

    clrfsf data_ee_addr      ;im Eeprom mit Adr 0 beginnend
    movlw LOW seflag1       ;Beginn im Rom
    movwf temp+1            ;LOW merken
    BANKSEL EEADR           ;
    movwf EEADR              ;in Register laden
    movlw HIGH seflag1      ;HIGH
    BANKSEL EEADRH          ;
    movwf EEADRH            ;in Register laden
eepromneul:
    BANKSEL EECON1          ;SCHLEIFE(1)
    bsf EECON1, EEPGD       ; lesen aus dem Programmspeicher
    bcf INTCON,GIE          ; Interrupt sperren
    btfsc INTCON,GIE        ; warten bis gesperrt
    goto eepromneul
    bsf EECON1, RD          ; lesen im Rom
    nop
    nop
    bsf INTCON,GIE          ; Interrupt wieder freigeben
    BANKSEL EEDATA          ;
    movf EEDATA, W          ; daten in W
    BANKSEL data_ee_data    ;
    movwf data_ee_data      ; in Eepromdaten laden
    BANKSEL PORTA          ;
    PAGESEL PAGE0          ;
    call ewrite              ; und abspeichern
    PAGESEL PAGE3          ;
    BANKSEL data_ee_addr    ;
    incf data_ee_addr,F      ; Eeprom addr + 1
    BANKSEL temp            ;
    incfsz temp+1,F         ; naechste Adresse im Rom LOW
    goto eepromneul2
    BANKSEL EEADRH          ; LOW Ueberlauf HIGH Addr + 1
    incf EEADRH,F          ;
    BANKSEL temp            ;
eepromneul2:
    movf temp+1,W          ;
    BANKSEL EEADR           ;
    movwf EEADR              ;
    BANKSEL temp            ;
    decfsz temp,F          ;
    goto eepromneul
    PAGESEL PAGE0          ;
    goto start
;=====
Dauerton:
    btfsc bgen              ;? Radioempfang
    goto Dauer06            ;
    bsf tonenable           ;0 Mithoerton ein
    PAGESEL PAGE0          ;
    call tx_to_dds           ; Sendefrequenz ein
    bsf senderein           ; Sender ein
    call LCDDisplayClear    ; Display loeschen
    PAGESEL PAGE3          ;
Dauer01:
    PAGESEL PAGE0          ; SCHLEIFE(1)
    call tastaturstatus    ;
    PAGESEL PAGE3          ; SCHLEIFE(2)
    SKPNC                   ;
    goto Dauer02            ; ENDE(2) Taste gedrueckt --> break
    PAGESEL PAGE0          ;
    call UPowerVor           ; Messen Vorlauf
    movwf uvor              ; und speichern
    call UPowerRueck        ; Messen Ruecklauf
    movwf urueck            ; und speichern
    PAGESEL PAGE2          ;
    call getswr              ; SWR errechnen
    PAGESEL PAGE0          ;
    LCDStr text3             ; "SWR: "
    PAGESEL PAGE3          ;
    BANKSEL bank1           ;
    movf swr+2,W            ; Ganzzahliges Ergebnis holen
    BANKSEL bank0          ;
    iorlw 30h                ;
    PAGESEL PAGE0          ;
    call LCDChar             ; und anzeigen
    movlw '.'                ; Dezimalpunk anzeigen
    call LCDChar            ;
    BANKSEL bank1           ;
    movf swr+1,W            ; Nachkommastelle anzeigen
    BANKSEL bank0          ;

```

```

MÄ€r 05, 06 15:20          mc_v1_12.pic          Seite 63/64
iorlw  30h          ;
call   LCDChar     ;
call   LCDSpace    ;
movlw  'P'         ; Leistung anzeigen
call   LCDChar     ;
movlw  ':'         ;
call   LCDChar     ;
PAGESEL PAGE3     ;
movf   uvor,W      ; aus Vorlauf errechnen
movwf  temp        ;
clrc                    ;
rrf    temp,F      ;
rrf    temp,W      ;
PAGESEL PAGE1     ;
call   u2pwr       ;
movwf  temp        ;
swapf  temp,W      ;
andlw  0fh         ;
iorlw  30h         ;
PAGESEL PAGE0     ;
call   LCDChar     ;
movlw  '.'         ;
call   LCDChar     ;
movf   temp,W      ;
andlw  0fh         ;
iorlw  30h         ;
call   LCDChar     ;
movlw  'W'         ; P:X.X W
call   LCDChar     ;
PAGESEL PAGE3     ;
Dauer03:
movlw  40h         ; Bargraph anzeigen in Zeile 2
PAGESEL PAGE0     ;
call   LCDPos      ;
PAGESEL PAGE3     ;
BANKSEL bank1     ;
movf   swr,W       ;
BANKSEL bank0     ;
movwf  schleife   ; Balken anzeigen
movlw  .16         ;
movwf  schleife+1 ;
Dauer04:
; SCHLEIFE(3)
movlw  0ffh        ; volles schwarzes Zeichen
decf   schleife,F  ;
btfsc  schleife,7  ; ? noch Balken anzeigen
movlw  ' '         ; 0 Leerzeichen
PAGESEL PAGE0     ;
call   LCDChar     ; anzeigen
PAGESEL PAGE3     ;
decfsz schleife+1,F ;
goto   Dauer04    ; ENDE(3) temp+3 = 0
btfss  strichpin   ;
goto   Dauer02    ;
btfss  punktpin   ;
goto   Dauer02    ;
goto   Dauer01    ; ENDE(1) Taste gedruickt oder Keyer betaetigt
Dauer02:
PAGESEL PAGE0     ;
call   LCDDisplayClear ;
LCDStr  text16     ; "break"
PAGESEL PAGE3     ;
Dauer05:
PAGESEL PAGE0     ;
call   tastaturstatus ; SCHLEIFE(1)
PAGESEL PAGE3     ;
SKPNC
goto   Dauer05    ;
btfss  strichpin   ;
goto   Dauer05    ;
btfss  punktpin   ;
goto   Dauer05    ; ENDE(1) keine Taste gedruickt oder kein Keyer
Dauer06:
bcf    senderein   ;1 Sender aus
bcf    tonenable   ; Mithoerton aus
PAGESEL PAGE0     ;
call   rx_to_dds   ; RX Frequenz einstellen
PAGESEL PAGE2     ;
goto   mtune01    ;
;=====
nop

```

```

MÄ€r 05, 06 15:20          mc_v1_12.pic          Seite 64/64
END
;=====

```