

```

;- DL8MCG ----- HAM-Radio-Ruftonfilter mit Görtzel-Algorithmus --- auf UniDsp56-Board von DL9GFA ----
COS_OMEGA_CT      equ    @COS(2*PI*1750/SAMPLE_FREQ)    ; Filter-Frequenz -> CT_cosw
dftCallTone_TH    equ    0.0019                        ; Default-Schwelle = 800 Hz
dftCallTone_Avg   equ    1000                          ; Default-Wert Anzahl Mittelungen

                org x:
CT_N_cnt          ds     1
CT_Avg            ds     1
CT_cosw           ds     1
CT_Q1             ds     1
CT_Lev           ds     1

                org y:
CT_Q2             ds     1

                org p:

;----- Init Ruftonfilter -----
CallTone_init     move    #>COS_OMEGA_CT,x0            ; Defaultwert Ruftonfrequenz
                  move    x0,x:CT_cosw
                  move    #>dftCallTone_TH,x0         ; Defaultwert Ruftonschwelle
                  move    x0,x:CT_Lev
                  move    #>dftCallTone_Avg,x0        ; Defaultwert Anzahl der Mittelungen
                  move    x0,x:CT_Avg
                  bclr    #DSW_CT1,x:DspStatusWord     ; Ruftonbit im Dsp-Status-Wort löschen
                  jmp     CallTone_init2

;----- Aufruf Ruftonfilter -----
CallTone_filt     move    #>-1,m2
                  move    #>-1,m3
                  move    #>-1,m4
                  move    #>CT_cosw,r2                ; cos(w)
                  move    #>CT_Q1,r3                  ; Q1
                  move    #>CT_Q2,r4                  ; Q2

                  move    x:CT_N_cnt,a                ; Anzahl der Mittelungen erreicht?
                  sub     #1,a
                  jmi    CallTone_Power
                  move    a,x:CT_N_cnt

                  move    x:Tx2_AfDacVal,a
                  asr     #8,a,a                       ; AF-Sample durch 2^8 teilen, x(i) -> a

                  move    x:(r3),x0 y:(r4),y0          ; Q1 -> x0 , Q2 -> y0
                  sub     y0,a x:(r2)+,y0             ; x(i) - Q2 -> a , cos(w) -> y0
                  mac     x0,y0,a x0,y:(r4)+          ; Q1 * cos(w) + x(i) - Q2 -> a , Q1 -> Q2
                  mac     x0,y0,a                     ; 2 * Q1 * cos(w) + x(i) - Q2 -> a
                  nop
                  move    a,x:(r3)+                   ; 2 * Q1 * cos(w) + x(i) - Q2 -> Q1
                  rts

;----- Bestimmung der Signalleistung -----
CallTone_Power    move    x:(r3)+,x0 y:(r4)+,y0        ; Q1 -> x0 , Q2 -> y0
                  mpy     x0,y0,b x:(r2)+,x1          ; Q1*Q2 -> b , cos(w) -> x1
                  mpy     y0,y0,a                     ; Q2^2 -> a
                  mac     x0,x0,a b,y1                ; Q2^2 + Q1^2 -> a , Q1*Q2 -> y1
                  mac     -x1,y1,a                    ; Q2^2 + Q1^2 -cos(w)*Q1*Q2 -> a
                  mac     -x1,y1,a                    ; Q2^2 + Q1^2 -2*cos(w)*Q1*Q2 -> a
                  ; in a steht die Leistung (quadriert)

;----- Auswertung, ob Ruftonschwelle erreicht -----
                  move    x:DspStatusWord,x0         ; bisheriger Zustand
                  move    x:CT_Lev,x1
                  cmp     x1,a
                  move    #>1<<DSW_CT1,a
                  jge     CallTone_Power2
                  move    #>~(1<<DSW_CT1),a
CallTone_Power2   eor     x0,a
                  and     #>1<<DSW_CT1,a
                  jeq     CallTone_init2              ;CallTone_Power3
                  bchg    #DSW_CT1,x:DspStatusWord
                  jsr     SendStatusToUc              ; Zustand an uC2 senden

;----- Initialisierung für den nächsten Mittelungsdurchgang -----
CallTone_init2    move    x:CT_Avg,x0
                  move    x0,x:CT_N_cnt              ; N = Anzahl Mittelungen
                  move    #0,x0
                  move    x0,x:CT_Q1                 ; Q1 = 0
                  move    x0,y:CT_Q2                 ; Q2 = 0
                  rts

```