

SINUS

;Berechnung des Sinus des Winkels in ersten Quadranten (0°- 90°)

;am Beispiel von 329,3° d.h.sin[30,7*(PI)/180]=-0,510543

;2x8 BIT Register werden für die Ganzzahl des winkels verwendet

;Verwendet wird die Taylor Reihe bis n=3

;4,096 MHZ / Instruction cycle =0,976525 µs

;Lasaros Goumas, Mai 2014

;Assembler directives

list,P16F627

#include <p16F627A.inc> ;Controller header file

_CONFIG_CP_OFF & _PWRTE_OFF & _WDT_OFF & _XT_OSC

ERRORLEVEL -302 ;Supress BanK SELECTION MESSAGES

;Zuordnung der verwendeten Register

count	equ	0x20	;Zählregister
H_multkator	equ	0x21	;Multiplikator High-Register
L_multkator	equ	0x22	;Multiplikator Low-register
LL_multkand	equ	0x23	;Rechtes Multiplikand Low Register
L_multkand	equ	0x24	;Multiplikand Low Register
H_multkand	equ	0x25	;Multiplikand High Register
L_Byte	equ	0x26	;Bogenmas Low bit Register
H_Byte	equ	0x27	;Bogenmas High bit Register
wkomma	equ	0x28	;Winkel Nachkommazahl
wgrad	equ	0x29	;Winkel transformiert in 1. Quadranten
sign	equ	0x2A	;Vorzeichen des transformierten Winkels (i=+/0=-)
H_taylor	equ	0x2B	;Taylor n=3 high bit register
L_taylor	equ	0x2C	;Taylor n=3 low bit register
L_sinus	equ	0x2D	;Low bit Register des sinus des Winkels
H_sinus	equ	0x2E	;High bit Register des Sinus des Winkels

;Ziele der Register Operationen

w equ 0 ;w=Zielregister

f equ 1 ;f=Zielregister

org 0x00 ;Programmbeginn bei Adresse 00h

goto start

;Subroutines

;Transformation des Winkels zum 1. Quadranten und Ermittlung seines Vorzeichens.

Quadrant movlw .3

SINUS

```

movwf    wkomma          ;Winkel Nachkommazahl
movlw    .73
movwf    L_multkator
movlw    .1
movwf    H_multkator    ;Multiplikator laden

bcf      STATUS, 2
bcf      STATUS, 0
movlw    .0
addwf    wkomma, 0
btfss   STATUS, 2        ;Ist Nachkommazahl = 0?
goto    $+02            ;Nein
goto    $+03            ;Ja
movlw    .89
goto    $+02
movlw    .90
subwf    L_multkator, 1
btfss   STATUS, 2        ;Ist L_multkator - 90° = 0?
goto    $+02            ;Nein
goto    loop3           ;Ja
btfsc   STATUS, 0        ;Borrow bit from L_multkator?
goto    $+04            ;Nein. wgrad > 90°
btfss   H_multkator, 0   ;Ja. Ist H_multkator = 1?
goto    loop3           ;Nein. wgrad < 90°
decf    H_multkator, 1   ;Ja. wgrad > 90°

bcf      STATUS, 2
bcf      STATUS, 0
movlw    .90
subwf    L_multkator, 1
btfss   STATUS, 2        ;Ist L_multkator - 180° = 0?
goto    $+02            ;Nein
goto    loop2           ;Ja
btfsc   STATUS, 0        ;Borrow bit from L_multkator?
goto    $+04            ;Nein. wgrad > 180°
btfss   H_multkator, 0   ;Ja. Ist H_multkator = 1?
goto    loop2           ;Nein. wgrad < 180°
decf    H_multkator, 1   ;Ja. wgrad > 180°

bcf      sign, 0
bcf      STATUS, 2
bcf      STATUS, 0
subwf    L_multkator, 1
btfss   STATUS, 2        ;Ist L_multkator - 270° = 0?
goto    $+02            ;Nein
goto    loop1           ;Ja.
btfsc   STATUS, 0        ;Borrow bit from L_multkator?
goto    $+02            ;Nein. wgrad > 270°
goto    loop1           ;Ja. wgrad < 270°

```

SINUS

```

movwf    wgrad                ;wgrad auf 90° setzen
movf     L_multkator, 0       ;L_multkator in w
subwf    wgrad, 1            ;90° - L_multkator
movlw    .0
addwf    wkomma, 0
btfss    STATUS, 2           ;Ist Nachkommazahl = 0?
goto     $+02                 ;Nein
goto     $+04                 ;Ja
movf     wkomma, 0           ;Nachkommazahl in w
sublw    .10
movwf    wkomma
goto     $+2E

loop1    movlw    .0
addwf    wkomma, 0
btfss    STATUS, 2           ;Ist Nachkommazahl = 0?
goto     $+02                 ;Nein
goto     $+06                 ;Ja
movf     L_multkator, 0
sublw    .255
sublw    .88
movwf    wgrad
goto     $+05
movf     L_multkator, 0
sublw    .255
sublw    .89
movwf    wgrad
goto     $+1F

loop2    bsf     sign, 0
decf     L_multkator, 1
movlw    .255
movwf    wgrad
movf     L_multkator, 0
subwf    wgrad, 1
movlw    .0
addwf    wkomma, 0
btfss    STATUS, 2           ;Ist Nachkommazahl = 0?
goto     $+02                 ;Nein
goto     $+04                 ;Ja
movf     wkomma, 0           ;Nachkommazahl in w
sublw    .10
movwf    wkomma
goto     $+10

loop3    bsf     sign, 0
movf     L_multkator, 0

```

SINUS

```

sublw    .255
movwf    wgrad
movlw    .0
addwf    wkomma, 0
btfss    STATUS, 2      ;Ist Nachkommazahl = 0?
goto     $+05           ;Nein
movf     wgrad, 0      ;Ja
sublw    .89
movwf    wgrad
goto     $+03
movf     wgrad, 0
sublw    .88
movwf    wgrad
return

```

;Zur Verwendung als Multiplikator wird der Winkelausdrucks in eine
;Ganzzahl umgewandelt (Winkel x 10 + Nachkomma).

```

Winkel   movf     wgrad, 0
          movwf    L_multkator      ;Winkel Ganzzahl laden

          movlw    .0
          addwf    wkomma, 0
          clrf     H_multkator      ;High Multiplikator löschen

          rlf      L_multkator, 1    ;Multiplikator x 2
          movf     L_multkator, 0
          movwf    wgrad            ;L_multkator x 2 in "Winkel"
          bcf      STATUS, 0
          rlf      L_multkator, 1
          btfss    STATUS, 0        ;Übertrag?
          goto     $+02            ;Nein
          rlf      H_multkator, 1   ;Ja Multiplikator x 4
          bcf      STATUS, 0
          rlf      L_multkator, 1
          rlf      H_multkator, 1   ;Multiplikator x 8
          btfss    STATUS, 0        ;Übertrag?
          goto     $+02            ;Nein
          incf     H_multkator, 1   ;Ja: High Multiplikator um 1 erhöhen

          movf     wgrad, 0        ;L_multkator x 2 in w
          bcf      STATUS, 0
          addwf    L_multkator, 1   ;(L_multkator x 8)+ (L_multkator x 2)
          btfss    STATUS, 0        ;Übertrag?
          goto     $+02            ;Nein
          incf     H_multkator, 1   ;Ja: High Multiplikator um 1 erhöhen
          movf     wkomma, 0       ;Nachkommazahl in w
          addwf    L_multkator, 1

```

SINUS

```

rrf      wgrad, 1      ;"Winkel x 2" zurück in "Winkel"

return

```

; Umwandlung der Winkel Ganzzahl in Bogenmass [(Winkel x 10 + Nachkomma) x PI/1800]

```

bogenmass      movlw    .10
                movwf   count
                bcf     STATUS, 0
                movlw   .26
                subwf   wgrad, 0
                btfsc   STATUS, 0      ;Borrow bit?
                goto    $+03          ;Nein.Bleibt bei 10 Durchläufen
                movlw   .2            ;Ja
                subwf   count, 1      ;Count auf 8 Durchläufen reduziert
                clrf    L_Byte
                clrf    H_Byte        ;Ergebnisregister löschen
                bcf     STATUS, 0      ;Carry bit löschen
                rrf     L_multkator, 1 ;Low Multiplikator rechts rotieren
                btfss   STATUS, 0      ;Überlauf?
                goto    $+0A          ;Nein. Counter prüfen
                bcf     STATUS, 0      ;Ja
                movf    L_multkand, 0
                addwf   L_Byte, 1
                btfss   STATUS, 0      ;Überlauf?
                goto    $+03          ;Nein
                bcf     STATUS, 0      ;Ja
                incf    H_Byte, 1      ;H_Byte um 1 erhöhen
                movf    H_multkand, 0
                addwf   H_Byte, 1
                decfsz  count, 1      ;Sind 10 bzw 8 Durchläufe vorbei?
                goto    $+02          ;Nein:
                goto    $+13          ;Ja: Ende der Multiplikation
                rlf     H_multkand, 1  ;High Multiplikand Register um 1 nach
links
                ;rotieren
                bcf     STATUS, 0      ;Carry bit löschen
                rlf     L_multkand, 1  ;Low Multiplikand Register nach links
rotieren
                btfss   STATUS, 0      ;Übertrag?
                goto    $+03          ;Nein
                incf    H_multkand, 1  ;Ja:High Multiplikand im 1 erhöhen
                bcf     STATUS, 0      ;Carry bit löschen

                rlf     LL_multkand, 1 ;Rechtes low Multiplikand Register nach
links
                ;rotieren
                btfss   STATUS, 0      ;Übertrag?

```

SINUS

```

goto    $+02           ;Nein
incf    L_multkand, 1  ;Low Multiplikand Register um 1 erhöhen
bcf     STATUS, 0     ;Carry bit löschen
rrf     H_multkator, 1 ;High multiplikator rechts rotieren
btfss   STATUS, 0     ;Überlauf?
goto    $-1E          ;Nein. Neuer Durchlauf
movlw   .128           ;Ja
addwf   L_multkator, 1
goto    $-21           ;Neuer Durchlauf
return

```

;Taylor Term Ermittlung für $n = 3$ ($x^2 \cdot x/6$)

```

taylor      clrf      L_taylor
            clrf      H_taylor           ;Taylor Ergebnisregister löschen
            movlw     .9
            movwf     count
            bcf       STATUS, 0
            movlw     .58
            subwf     wgrad, 0
            btfsc    STATUS, 0           ;Borrow bit?
            goto     $+02                ;Nein. Bleibt bei 9 Durchläufen
            decf     count, 1            ;Ja. Count auf 8 Durchläufen reduziert
            clrf     H_multkand
            movf     H_Byte, 0
            movwf    H_multkator         ;H_Byte an high Multiplikator Register
            movwf    L_multkand         ;H_Byte an low Multiplikand Register
            movf     L_Byte, 0
            movwf    L_multkator        ;L_Byte an low Multiplikator Register
            movwf    LL_multkand        ;L_Byte an rechten low Multiplikand

Register    bcf       STATUS, 0         ;Carry bit löschen
            rrf       L_multkator, 1     ;Low multiplikator rechts rotieren
            btfss   STATUS, 0           ;Überlauf?
            goto    $+0A                ;Nein. Counter prüfen
            bcf     STATUS, 0           ;Ja
            movf    L_multkand, 0
            addwf   L_taylor, 1
            btfss   STATUS, 0           ;Überlauf?
            goto    $+03                ;Nein
            bcf     STATUS, 0           ;Ja
            incf    H_taylor, 1         ;H_Byte um 1 erhöhen
            movf    H_multkand, 0
            addwf   H_taylor, 1
            decfsz  count, 1            ;Sind 9 bzw 8 Durchläufe vorbei?
            goto    $+02                ;Nein:
            goto    $+13                ;Ja: Ende der Ermittlung von x2
            rlf     H_multkand, 1       ;High Multiplikand Register um 1 nach

```

SINUS

links

```

bcf     STATUS, 0      ;rotieren
rlf     L_multkand, 1  ;Carry bit löschen
                    ;Low Multiplikand Register nach links

```

rotieren

```

btfss  STATUS, 0      ;Übertrag?
goto   $+03           ;Nein
incf   H_multkand, 1  ;Ja:High Multiplikand im 1 erhöhen
bcf    STATUS, 0      ;Carry bit löschen

```

links

```

rlf     LL_multkand, 1 ;Rechtes low Multiplikand Register nach

```

```

                    ;rotieren
btfss  STATUS, 0      ;Übertrag?
goto   $+02           ;Nein
incf   L_multkand, 1  ;Low Multiplikand Register um 1 erhöhen
bcf    STATUS, 0      ;Carry bit löschen
rrf    H_multkator, 1 ;High multiplikator rechts rotieren
btfss  STATUS, 0      ;Überlauf?
goto   $-1E           ;Nein. Neuer Durchlauf
movlw  .128            ;Ja
addwf  L_multkator, 1
goto   $-21           ;Neuer Durchlauf

```

```

movlw  .10             ;Beginn der Ermittlung von (x2*x=x3)

```

```

movwf  count
movlw  .58
subwf  wgrad, 0
btfsc  STATUS, 0      ;Borrow bit?
goto   $+03           ;Nein.Bleibt bei 10 Durchläufen
movlw  .2
subwf  count, 1       ;Count auf 8 Durchläufen reduziert
movf   H_taylor, 0
movwf  H_multkator    ;H_taylor an high Multiplikator Register
movf   L_taylor, 0
movwf  L_multkator    ;L_taylor an low Multiplikator Register
clrf   H_multkand
movf   H_Byte, 0
movwf  L_multkand     ;H_Byte an low Multiplikand Register
movf   L_Byte, 0
movwf  LL_multkand    ;L_Byte an rechten low Multiplikand

```

Register

```

clrf   L_taylor
clrf   H_taylor       ;Taylor Ergebnisregister löschen
bcf    STATUS, 0      ;Carry bit löschen
rrf    L_multkator, 1 ;Low multiplikator rechts rotieren
btfss  STATUS, 0      ;Überlauf?
goto   $+0A           ;Nein. Counter prüfen
bcf    STATUS, 0      ;Ja

```

SINUS

```

movf    L_multkand, 0
addwf   L_taylor, 1
btfss   STATUS, 0      ;Überlauf?
goto    $+03           ;Nein
bcf     STATUS, 0      ;Ja
incf    H_taylor, 1    ;H_Byte um 1 erhöhen
movf    H_multkand, 0
addwf   H_taylor, 1
decfsz  count, 1      ;Sind 10 bzw. 8 Durchläufe vorbei?
goto    $+02           ;Nein:
goto    $+13           ;Ja: Ende der Ermittlung von x3
rlf     H_multkand, 1  ;High Multiplikand Register um 1 nach
links
                                ;rotieren
bcf     STATUS, 0      ;Carry bit löschen
rlf     L_multkand, 1  ;Low Multiplikand Register nach links
rotieren
btfss   STATUS, 0      ;Übertrag?
goto    $+03           ;Nein
incf    H_multkand, 1;Ja:High Multiplikand im 1 erhöhen
bcf     STATUS, 0      ;Carry bit löschen
rlf     LL_multkand, 1 ;Rechtes low Multiplikand Register nach
links
                                ;rotieren
btfss   STATUS, 0      ;Übertrag?
goto    $+02           ;Nein
incf    L_multkand, 1  ;Low Multiplikand Register um 1 erhöhen
bcf     STATUS, 0      ;Carry bit löschen
rrf     H_multkator, 1 ;High multiplikator rechts rotieren
btfss   STATUS, 0      ;Überlauf?
goto    $-1E           ;Nein. Neuer Durchlauf
movlw   .128           ;Ja
addwf   L_multkator, 1
goto    $-21           ;Neuer Durchlauf
movlw   .10            ;Beginn der Ermittlung von x3/6 =
x3*0,166667
movwf   count
movlw   .58
subwf   wgrad, 0
btfsc   STATUS, 0      ;Borrow bit?
goto    $+03           ;Nein.Bleibt bei 10 Durchläufen
movlw   .2             ;Ja
subwf   count, 1      ;Count auf 8 Durchläufen reduziert
movf    H_taylor, 0
movwf   H_multkator   ;H_taylor an high Multiplikator Register
movf    L_taylor, 0
movwf   L_multkator   ;L_taylor an low Multiplikator Register

```


SINUS

```

    clrf    H_multkand
    clrf    L_multkand
    movlw   .40
    movwfm LL_multkand    ;1/6 im rechten low Multiplikand Register

laden

    clrf    L_taylor
    clrf    H_taylor      ;Taylor Ergebnisregister löschen
    bcf     STATUS, 0
    rrf     L_multkator, 1 ;Low multiplikator rechts rotieren
    btfss   STATUS, 0     ;Überlauf?
    goto    $+0A         ;Nein. Counter prüfen
    bcf     STATUS, 0     ;Ja
    movf    L_multkand, 0
    addwfm L_taylor, 1
    btfss   STATUS, 0     ;Überlauf?
    goto    $+03         ;Nein
    bcf     STATUS, 0     ;Ja
    incf    H_taylor, 1   ;H_Byte um 1 erhöhen
    movf    H_multkand, 0
    addwfm H_taylor, 1
    decfsz  count, 1     ;Sind 10 bzw. 8 Durchläufe vorbei?
    goto    $+02         ;Nein:
    goto    $+13         ;Ja: Ende der Ermittlung von x3
    rlf     H_multkand, 1 ;High Multiplikand Register um 1 nach

links

                                ;rotieren
    bcf     STATUS, 0     ;Carry bit löschen
    rlf     L_multkand, 1 ;Low Multiplikand Register nach links

rotieren

    btfss   STATUS, 0     ;Übertrag?
    goto    $+03         ;Nein
    incf    H_multkand, 1 ;Ja:High Multiplikand im 1 erhöhen
    bcf     STATUS, 0     ;Carry bit löschen

links

    rlf     LL_multkand, 1 ;Rechtes low Multiplikand Register nach

                                ;rotieren
    btfss   STATUS, 0     ;Übertrag?
    goto    $+02         ;Nein
    incf    L_multkand, 1 ;Low Multiplikand Register um 1 erhöhen
    bcf     STATUS, 0     ;Carry bit löschen
    rrf     H_multkator, 1 ;High multiplikator rechts rotieren
    btfss   STATUS, 0     ;Überlauf?
    goto    $-1E         ;Nein. Neuer Durchlauf
    movlw   .128         ;Ja
    addwfm L_multkator, 1
    goto    $-21         ;Neuer Durchlauf
    return

```

SINUS

;Hauptprogramm

```
start      call    Quadrant
           call    Winkel
           movlw   B'01110010'
           movwf  LL_multkand      ;Multiplikand rechtes Low-register mit
                                   ;0,0001740 laden
           clrf   L_multkand      ;Multiplikand Low-register mit 0,0 laden
                                   ;entsprechend (PI)/1800 = 0,01740
           clrf   H_multkand      ;Multiplikand High-register löschen
           call   bogenmass
           call   taylor
           bcf    STATUS, 0
           clrf   H_sinus
           clrf   L_sinus        ;Sinus Register löschen

           movf   L_taylor, 0
           subwf  L_Byte, 0
           movwf  L_sinus
           btfsc  STATUS, 0      ;Übertrag?
           goto   $+04          ;Nein
           incf   H_sinus, 0     ;Ja. High bit von Sinus um 1 erhöhen
           subwf  H_Byte, 0
           movwf  H_sinus
           goto   $

           end
```

;*****