

## SPI\_Hello World

```
;"Hello World" Dot Matrix LCD Anzeige mit "Serial Peripheral Interface"
;RA<7:0> Outputs
;RB<7.0> Outputs
;Fosc=4 /Instruction cyrc=1µS (Osc <500kHz d.h. XT_OSC)
;Lasaros Goumas: April 2020
;*****

;Assembler directives
    list,P16F876A
    #include <p16F876A.inc>           ;Controller header file
    __CONFIG_CP_OFF & _PWRTE_OFF & _WDT_OFF & _XT_OSC & _LVP_OFF
    ERRORLEVEL      -302           ;Supress Bank SELECTION MESSAGES
;*****

;Variablen

w_copy      equ    0x20           ;saved w register
s_copy      equ    0x21           ;saved STATUS register
counter1    equ    0x22           ;Allgemeines zählregister
counter2    equ    0x23           ;Allgemeines zählregister
disp_store  equ    0x24           ;Display daten Zwischenspeicher
;*****

;definitionen

#define DISP_CSB      PORTA, 4      ;Display Aktivierung
#define DISP_CLK      PORTB, 6      ;Display clock
#define DISP_SI       PORTB, 7      ;Display Datenleitung
#define DISP_RS       PORTA, 3      ;Command/Data selector
#define DISP_LED      PORTA, 2      ;Display activ
;*****

;Ziele der Register Operationen

w            equ    0              ;w Zielregister
f            equ    1              ;f Zielregister
;*****

;Macro Definitionen

bank0        macro
    bcf      STATUS, 5
    bcf      STATUS, 6
endm

bank1        macro
    bsf      STATUS, 5
endm

display_init macro
    movlw   0x38                  ;"Function set"
    call    write_command
```

SPI\_Hello World

```

movlw    0x39                ;"Function set"
call     write_command
movlw    0x14                ;"Bias set"
call     write_command
movlw    0x78                ;"Contrast set"
call     write_command
movlw    0x52                ;"Power/ICON/Contrast control"
call     write_command
movlw    0x69                ;"Follower control"
call     write_command
movlw    0xC8
call     delay                ;200msec warten
movlw    0x0C                ;"Display /ON/OFF control"
call     write_command
movlw    0x01                ;"Clear display"
call     write_command
NOP
NOP                            ;2msec warten
movlw    0x06                ;Entry mode set"
call     write_command
endm

```

\*\*\*\*\*

```

org      0x00                ;Programmbeginn bei Adresse 00h
goto    start

```

\*\*\*\*\*

;Subroutines

;Subroutine Initialisierung

```

init          bank1
movlw        B'00000001'
movwf        TRISA                ;RA<7:1> Outputs/RA0 Input
clrf         TRISB                ;RB<7:0> configurer as outputs
bcf          OPTION_REG, 7        ;Pull-ups "ON"
bank0
clrf         PORTA
clrf         PORTB
clrf         disp_store
bsf          DISP_CSB              ;Display disabled
bcf          DISP_CLK              ;Clock line auf "NULL"
bcf          DISP_SI               ;No data
bcf          DISP_RS               ;Command selected
clrf         counter1
clrf         counter2
bcf          INTCON, 7             ;GIE disabled
return

```

;Subroutine für 30µsec delay

## SPI\_Hello World

```

delay03      movlw    0x0F
             movwf    counter2
             decfsz   counter2, 1      ;30µsec vorbei?
             goto     $-01             ;Nein.
             return
    
```

;Subroutine für eine Verzögerung zwischen 1msec und 200msec

```

delay        movwf    counter1        ;Übergabeparameter für Delay sichern
             movlw    .250            ;Bei Verwendung eines 4-MHz-Taktes
             movwf    counter2
             nop
             decfsz   counter2, 1     ;1msec vorbei?
             goto     $-02            ;Nein
             decfsz   counter1, 1     ;Ja. Delay abgelaufen?
             goto     $-06            ;Nein
             return
    
```

;Subroutine für das Schreiben eines Kommandos

```

write_command movwf    disp_store      ;Daten im Speicherregister
             bcf     DISP_CSB          ;Display aktivieren
             bcf     DISP_RS           ;Kommando empfangen
             bcf     DISP_CLK
             movlw   0x08
             movwf   counter1          ;Es werden 8 Bits übertragen
             rlf     disp_store, 1     ;Display Zeichenspeicher nach links.
             btfsc   STATUS, 0         ;Carry Bit "NULL"
             goto    $+03              ;Nein. Datum übernehmen
             bcf     DISP_SI           ;Ja. Datum nicht übernehmen
             goto    $+02
             bsf     DISP_SI
             NOP
             bsf     DISP_CLK
             NOP
             bcf     DISP_CLK
             decfsz  counter1, 1       ;Display Ausgabezähler "NULL" ?
             goto    $-0B              ;Nein
             bcf     DISP_SI
             bcf     DISP_RS
             call    delay03           ;30µsec warten
             return
    
```

;Subroutine für das Schreiben eines Zeichens

```

write_data    movwf    disp_store      ;Daten im Speicherregister
             bcf     DISP_CSB          ;Display aktivieren
             bsf     DISP_RS           ;Data empfangen
             bcf     DISP_CLK
    
```

## SPI\_Hello World

```

movlw    0x08
movwf    counter1        ;Es werden 8 Bits übertragen
rlf      disp_store, 1   ;Display Zeichenspeicher nach links.
btfsc    STATUS, 0      ;Carry Bit "NULL"?
goto     $+03           ;Nein. Datum übernehmen
bcf      DISP_SI        ;Ja. Datum nicht übernehmen
goto     $+02
bsf      DISP_SI
NOP
bsf      DISP_CLK
NOP
bcf      DISP_CLK
decfsz   counter1, 1    ;Display Ausgabezähler "NULL" ?
goto     $-0B          ;Nein
bcf      DISP_SI
bcf      DISP_RS
call     delay03        ;Warte 30µsec
return

```

;\*\*\*\*\*

;Hauptprogramm

```

start      call    init

            movlw   0x28
            call    delay            ;40msec warten
            display_init

mainloop    bsf     PORTA, 2        ;Übernahme von Daten beginnt
            movlw   0x80
            call    write_command    ;Position 1 der Zeile 1 (=0x80+0x00)
            movlw   0x20
            call    write_data       ;Leerzeichen 1 in Zeile 1
            movlw   0x20
            call    write_data
            movlw   0x20
            call    write_data
            movlw   A'.'
            call    write_data       ;"." Zeichen 4 in Zeile 1
            movlw   A'.'
            call    write_data
            movlw   A'H'
            call    write_data       ;"H" Zeichen 6 in Zeile 1
            movlw   A'E'
            call    write_data
            movlw   A'L'
            call    write_data
            movlw   A'L'
            call    write_data
            movlw   A'O'
            call    write_data
            movlw   A'.'

```

SPI\_Hello World

```
call    write_data
movlw  A'.'
call    write_data
movlw  0xC3
call    write_command    ;Position 4 der Zeile 2 (=0x80+0x43)
movlw  A'.'
call    write_data        ;"." Zeichen 4 in Zeile 2
movlw  A'.'
call    write_data
movlw  A'W'
call    write_data        ;"W" Zeichen 6 in Zeile 2)
movlw  A'O'
call    write_data
movlw  A'R'
call    write_data
movlw  A'L'
call    write_data
movlw  A'D'
call    write_data
movlw  A'.'
call    write_data
movlw  A'.'
call    write_data
movlw  .250
call    delay
movlw  .250
call    delay
movlw  .250
call    delay
movlw  .250
call    delay            ;Warte 750msec
bcf    PORTA, 2        ;Übernahme von Daten beendet
movlw  .250
call    delay
movlw  .250
call    delay
movlw  .250
call    delay            ;Warte 750msec
goto   mainloop
```

end

;\*\*\*\*\*