

RS-232

```
;Hiermit wird eine serielle RS-232 Kommunikation mit einem PC realisiert.  
;Beim Tastendruck wird die Kennung des gedrückten Tasters zum PC übertragen.  
;Wird auf der PC-Tastatur eine Ziffer zwischen 1 und 4 gedrückt,  
;wird eine ein- oder ausgeschaltet.
```

*

```
;RA<3:2> Digitale Eingänge. RA<7:6> Digitale Eingänge  
;RB<5:4> Digitale Ausgänge. RB<3:0> Digitale Eingänge  
;RC<1:0> Digitale Eingänge. RC<7:6> Configured as RX/TX pins  
;Fosc=4MHz /Instruction cyrce=1µS (Osc <500kHz d.h. XT_OSC)  
;Lasaros Goumas: Mai 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  
daten equ 0x22 ;Zwischenspeicher für die seriellen Daten  
counter1 equ 0x23 ;Allgemeines Zählregister  
counter2 equ 0x24 ;Allgemeines Zählregister  
;*****
```

```
;definitionen
```

```
#define LED_1 PORTA, 2 ;LED 1 liegt an Pin RA2  
#define LED_2 PORTA, 3 ;LED 2 liegt an Pin RA3  
#define LED_3 PORTB, 4 ;LED 3 liegt an Pin RB4  
#define LED_4 PORTB, 5 ;LED 4 liegt an Pin RB5  
  
#define TASTER_1 PORTA, 4 ;Taster 1 liegt an Pin RA4  
#define TASTER_2 PORTA, 5 ;Taster 2 liegt an Pin RA5  
#define TASTER_3 PORTC, 0 ;Taster 3 liegt an Pin RC0  
#define TASTER_4 PORTC, 1 ;Taster 4 liegt an Pin RC1  
;*****
```

```
;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
;*****

        org     0x00             ;Programmbeginn bei Adresse 00h
        goto    start
;*****

        org     0x04             ;Interrupt service Adresse

;Interrupt service vector

        bcf     INTCON, 7        ;Global Interrupt disabled
        movwf   w_copy           ;save w in 0x20
        movf    STATUS, 0        ;Verschiebe STATUS in w
        movwf   s_copy           ;save STATUS in 0x21

;Interrupt routine start

;Interrupt routine ende

intrent bcf     INTCON, 1        ;RBO/INT Flag (INTF) cleared

;restore context and return to the main programm

        movf    s_copy, 0        ;Hole STATUS zurück in w
        movwf   STATUS           ;Daten zurück in STATUS
        movf    w_copy, 0        ;Akku daten sichern
        retfie                    ;Interrupt ende und GIE enabled.
;*****

;Subroutines

;Subroutine Initialisierung

init    bank1
        movlw   0x06
        movwf   ADCON1          ;Configur all RA pins as digital

        movlw   B'00110011'
        movwf   TRISA           ;RA<3:2> Ausgänge;. RA<5:4> Eingänge
        movlw   B'00001111'
        movwf   TRISB           ;RB<3:0> Eingänge. RB<5:4> Ausgänge
        movlw   B'11000011'
        movwf   TRISC           ;RC<7:6> RX/TX. RC<1:0> Digitale Eingänge.

```

RS-232

```

bank0
clrf    PORTA
clrf    PORTB
clrf    PORTC
return

```

;Subroutine Intialisierung des RS-232 Interface

```

init_serial    bank1
               movlw    0x0C                ;Baudrate auf 19,2k (19.231 KBAUD)
               movwf    SPBRG
               movlw    B'00000100'        ;8bit/Asynchron/High Speed
               movwf    TXSTA
               bank0
               movlw    B'10010000'        ;Serial port enabled/8 bit/Continuous receipt
               movwf    RCSTA
               return

```

;Subroutine Daten Übertragung

```

transmit       bank1
               bsf     TXSTA, 5            ;Übertragung freischalten
               bank0
               movwf   TXREG
               bank1
               btfss  TXSTA,1             ;Transmit shift register emty?
               goto   $-01                ;Nein.
               bank0
               ;Ja. Alles gesendet
               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

```

;*****

;Hauptprogramm

```

start    call    init
         call    init_serial

mainloop btfsc   PIR1, 5                ;UART receiver Flag "empty"?

```

RS-232

```

goto    read_data    ;Nein.
btfss  TASTER_1     ;Ja. Taster 1 losgelassen?
goto    taster1      ;Nein. Taster 1 Gedrückt
btfss  TASTER_2     ;Taster 2 losgelassen?
goto    taster2
btfss  TASTER_3     ;Taster 3 losgelassen?
goto    taster3
btfss  TASTER_4     ;Taster 4 losgelassen?
goto    taster4
goto    mainloop     ;Taster und UART receiver erneut abfragen

read_data  movf    RCREG, 0    ;UART receiver data.
           movwf   daten      ;Copy of UART receiver data
           movf    daten, 0
           xorlw   A'1'       ;Receiver Daten mit 1 vergleichen
           btfsc  STATUS, 2   ;Flag "Z" 0?
           goto    led1       ;Nein. Es wurde eine 1 übertragen.
           movf    daten, 0
           xorlw   A'2'       ;Receiver Daten mit 2 vergleichen
           btfsc  STATUS, 2   ;Flag "Z" 0?
           goto    led2       ;Nein. Es wurde eine 2 übertragen.
           movf    daten, 0
           xorlw   A'3'       ;Receiver Daten mit 3 vergleichen
           btfsc  STATUS, 2   ;Flag "Z" 0?
           goto    led3       ;Nein. Es wurde eine 3 übertragen.
           movf    daten, 0
           xorlw   A'4'       ;Receiver Daten mit 4 vergleichen
           btfsc  STATUS, 2   ;Flag "Z" 0?
           goto    led4       ;Nein Es wurde eine 4 übertragen.
           goto    mainloop

led1      btfsc  PORTB, 0     ;LED an?
           goto    $+03       ;Nein
           bcf    LED_1       ;Ja. LED ausschalten
           goto    $+02
           bsf    LED_1       ;LED einschalten
           goto    mainloop

led2      btfsc  PORTB, 1     ;LED an?
           goto    $+03       ;Nein
           bcf    LED_2       ;Ja. LED ausschalten
           goto    $+02
           bsf    LED_2       ;LED einschalten
           goto    mainloop

led3      btfsc  PORTB, 3     ;LED an?
           goto    $+03       ;Nein
           bcf    LED_3       ;Ja. LED ausschalten
           goto    $+02
           bsf    LED_3       ;LED einschalten
           goto    mainloop

```

```

                                RS-232
led4      btfsc    PORTB, 2      ;LED an?
          goto    $+03          ;Nein
          bcf     LED_4         ;Ja. LED ausschalten
          goto    $+02
          bsf     LED_4         ;LED einschalten
          goto    mainloop

taster1   movlw   A'S'
          call    transmit
          movlw   A'1'
          call    transmit      ;Es wird "S1" mit Return gesendet
          movlw   A'\n'        ;Steht für LF = Linefeed
          call    transmit
          movlw   A'\r'        ;Steht für CR = Carrige Return
          call    transmit
          btfss   TASTER_1     ;Taster 1 losgelassen?
          goto    $-1          ;Nein
          movlw   0x14
          call    delay         ;20msec zum enprellen des Tasters abwarten
          goto    mainloop

taster2   movlw   A'S'
          call    transmit
          movlw   A'2'
          call    transmit      ;Es wird "S2" mit Return gesendet
          movlw   A' '        ;Leerzeichen
          call    transmit
          movlw   A'\r'        ;Steht für CR = Carrige Return
          call    transmit
          btfss   TASTER_2     ;Taster 2 losgelassen?
          goto    $-1          ;Nein
          movlw   0x14
          call    delay         ;20msec zum enprellen des Tasters abwarten
          goto    mainloop

taster3   movlw   A'S'
          call    transmit
          movlw   A'3'
          call    transmit      ;Es wird "S3" mit Return gesendet
          movlw   0x0A        ;Steht für LF = Linefeed
          call    transmit
          movlw   0x0D        ;Steht für CR = Carrige Return
          call    transmit
          btfss   TASTER_3     ;Taster 3 losgelassen?
          goto    $-1          ;Nein
          movlw   0x14
          call    delay         ;20msec zum enprellen des Tasters abwarten
          goto    mainloop

taster4   movlw   A'S'
          call    transmit
          movlw   A'4'

```

```

                                RS-232
call    transmit                ;Es wird "S4" mit Return gesendet
movlw   A' '
call    transmit                ;Leerzeichen
movlw   0x07
call    transmit                ;PC gibt einen signalton aus
btfss   TASTER_4               ;Taster 3 losgelassen?
goto    $-1                    ;Nein
movlw   0x14
call    delay                   ;20msec zum enprellen des Tasters abwarten
goto    mainloop
```

```
end
```

```
;*****
```