```
.include    "8515def.inc"

; Warning: set up FUSE-Bits ! See Datasheet!

.def        temp    =r16
.def        address =r21
.def        value   =r22
.def        mask    =r23                ; for BitModify
.def        stack   =r24
.def        stack2  =r25
; mcp2515 Instructions:
.equ        WRITE   =0b00000010
.equ        READ    =0b00000011
.equ        RESET   =0b11000000
.equ        BITMODIFY                   =0b00000101

; mcp2515 addresses:
.equ        RXB0D0  =0b01100110
.equ        RXB0D1  =0b01100111
.equ        CANINTF =0b00101100
.equ        CANINTE =0b00101011
.equ        CNF1    =0b00101010
.equ        CNF2    =0b00101001
.equ        CNF3    =0b00101000
.equ        BFPCTRL =0b00001100
.equ        CANCTRL =0b00001111
.equ        TXB0SIDH                    =0b00110001
.equ        TXB0SIDL                    =0b00110010
.equ        TXB0DLC =0b00110101
.equ        TXB0D0  =0b00110110
.equ        TXB0D1  =0b00110111
.equ        TXB0CTRL                    =0b00110000
.equ        TEC     =0b00011100
.equ        REC     =0b00011101
.equ        EFLG    =0b00101101
;Masks
.equ        RXM0SIDH                    =0b00100000
.equ        RXM0SIDL                    =0b00100001
.equ        RXM0EID8                    =0b00100010
.equ        RXM0EID0                    =0b00100011
.equ        RXM1SIDH                    =0b00100100
.equ        RXM1SIDL                    =0b00100101
.equ        RXM1EID8                    =0b00100110
.equ        RXM1EID0                    =0b00100111

.equ        RXB0CTRL                    =0b01100000
.equ        RXB1CTRL                    =0b01110000


.org        0x000

; Stackpointer
            ldi     temp,LOW(RAMEND)
            out     SPL,temp
            ldi     temp,HIGH(RAMEND)
            out     SPH,temp
```

```
; Port A
            ldi     temp,0x00
            out     DDRA,temp           ; EEEEEEEE Input (from PLC) Level

; Port C
            ldi     temp,0x00           ; EE(Sensor Failure)EE(Flowswitch
            out     DDRC,temp

; mcp Startup Delay
            rcall   wait_100ms

; now SPI can be used in mcp2515

            ldi     temp,25             ; 9600baud
            out     UBRR,temp
            sbi     UCR,TXEN            ; TX aktivieren

; SPI Master Init
            ldi     temp,0b10111000     ; Output = SCK & MOSI & /SS & LED
            out     DDRB,temp

            sbi     PortB,4             ; /CS  High

; SPIEnabled, MasterMode, SPI Clock Rate = OSC/128
            ldi     temp,(0<<SPIE)|(1<<SPE)|(0<<DORD)|(1<<MSTR)|(0<<CPOL)
            out     SPCR,temp


; ====================== MCP2515 INIT ==================================

            rcall   mcp_reset
            rcall   wait_100ms

; ===================== MCP2515 CONFIGURATION MODE =====================

            ; CNF1:
            ldi     address,CNF1
            ldi     value,0x01
            rcall   sendbyte

            ; CNF2:
            ldi     address,CNF2
            ldi     value,0xA0
            rcall   sendbyte

            ; CNF3:
            ldi     address,CNF3
            ldi     value,0x02
            rcall   sendbyte

            ; INTERRUPTS
            ldi     address,CANINTE
            ldi     value,0b00000100    ; Transmit Buffer 0 Empty INT Ena
            rcall   sendbyte

; ==================== LOOPBACK MODE ==========================
```

```
                ldi     address,CANCTRL
                ldi     value,0b00000000
                rcall   sendbyte

                ; Receive all messages
                ldi     address,RXB0CTRL
                ldi     value,0b01100000
                rcall   sendbyte

                ldi     address,RXB1CTRL
                ldi     value,0b01100000
                rcall   sendbyte

                ; MASKS
                ldi     address,RXM0SIDH
                ldi     value,0b00000000
                rcall   sendbyte

                ldi     address,RXM0SIDL
                ldi     value,0b01000000
                rcall   sendbyte

                ldi     address,RXM0EID8
                ldi     value,0b00000000
                rcall   sendbyte

                ldi     address,RXM0EID0
                ldi     value,0b01000000
                rcall   sendbyte

                ldi     address,RXM1SIDH
                ldi     value,0b00000000
                rcall   sendbyte

                ldi     address,RXM1SIDL
                ldi     value,0b01000000
                rcall   sendbyte

                ldi     address,RXM1EID8
                ldi     value,0b00000000
                rcall   sendbyte

                ldi     address,RXM1EID0
                ldi     value,0b01000000
                rcall   sendbyte

                rcall   wait_100ms



                ; MESSAGE TRANSMISSION (periodic)
send:
                rcall   erroroutput
                rcall   wait_100ms
                rcall   wait_100ms
                rcall   wait_100ms
```

```
            rjmp    send


; clear TXREQ-Flag manually
            ldi     address,TXB0CTRL
            ldi     value,0b00000000
            rcall   sendbyte

; define ID (Std. ID High) (XXXXXXXX)
            ldi     address,TXB0SIDH
            ldi     value,0b00000000
            rcall   sendbyte

; define ID (Std. ID Low) (XXXddddd)
            ldi     address,TXB0SIDL
            ldi     value,0b00000000
            rcall   sendbyte

; define Data Length and Remote/Data Frame
            ldi     address,TXB0DLC
            ldi     value,0b00000010
            rcall   sendbyte

; Data to send
            ldi     address,TXB0D0
            ldi     value,0b11100111
            rcall   sendbyte

; Data to send
            ldi     address,TXB0D1
            ldi     value,0b00011000
            rcall   sendbyte



; wait while send....


; TRANSMIT-LED PB1 on
            sbi     PortB,3

; Send...:
            ldi     address,TXB0CTRL
            ldi     value,0b00001011    ; set TXREQ-Flag, Highest Priorit
            rcall   sendbyte


; Transmission starts, when Bus is available.

; TRANSMIT-LED PB1 off
            cbi     PortB,3

; Interval Time:
            rcall   wait_500ms
            rcall   wait_500ms
```

```
            rcall   wait_500ms
            rcall   wait_500ms

            rjmp    send



;_____________________________________________________________________
getbyte:
            nop
            nop
; /SS low
            cbi     PortB,4
; READ COMMAND
            ldi     temp,READ
            out     SPDR,temp
wait_spi_g1:
            sbis    SPSR,SPIF           ; Transmission complete?
            rjmp    wait_spi_g1
            nop
            nop
            in      temp,SPDR           ; release SPIF here
            rcall   wait_10ms
; SET ADDRESS
            out     SPDR,address
wait_spi_g2:
            sbis    SPSR,SPIF           ; Transmission complete?
            rjmp    wait_spi_g2
            nop
            nop
            in      temp,SPDR           ; release SPIF here
            rcall   wait_10ms
; DUMMY BYTE
            ldi     temp,0b10101010
            out     SPDR,temp
wait_spi_g3:
            sbis    SPSR,SPIF           ; Transmission complete?
            rjmp    wait_spi_g3
; RESULT:
            nop
            nop
            in      temp,SPDR           ; release SPIF here
            rcall   wait_10ms
; /SS high
            sbi     PortB,4
            nop
            nop
            ret
;_____________________________________________________________________
modifybyte:
            nop
            nop
; /SS low
            cbi     PortB,4
; BITMODIFY COMMAND
```

```
            ldi     temp,BITMODIFY
            out     SPDR,temp
wait_spi_b1:
            sbis    SPSR,SPIF           ; Transmission complete?
            rjmp    wait_spi_b1
            nop
            nop
            in      temp,SPDR           ; release SPIF here
            rcall   wait_10ms
; SET ADDRESS
            out     SPDR,address
wait_spi_b2:
            sbis    SPSR,SPIF           ; Transmission complete?
            rjmp    wait_spi_b2
            nop
            nop
            in      temp,SPDR           ; release SPIF here
            rcall   wait_10ms
; MASK BYTE
            out     SPDR,mask
wait_spi_b3:
            sbis    SPSR,SPIF           ; Transmission complete?
            rjmp    wait_spi_b3
            nop
            nop
            in      temp,SPDR           ; release SPIF here
            rcall   wait_10ms
; BITS TO BE CHANGED
            out     SPDR,value
wait_spi_b4:
            sbis    SPSR,SPIF           ; Transmission complete?
            rjmp    wait_spi_b4
            nop
            nop
            in      temp,SPDR           ; release SPIF here
            rcall   wait_10ms
; /SS high
            sbi     PortB,4
            nop
            nop
            ret
;_____________________________________________________________________
mcp_reset:
            nop
            nop
; /SS low
            cbi     PortB,4
            ldi     temp,0b11000000     ; RESET-Instruction
            out     SPDR,temp
wait_spi_r:
            sbis    SPSR,SPIF           ; Transmission complete?
            rjmp    wait_spi_r
            nop
            nop
            in      temp,SPDR           ; release SPIF here
            rcall   wait_10ms
; /SS high
```

```
            sbi     PortB,4
            nop
            nop
            ret
;_____________________________________________________________________
sendbyte:
            nop
            nop
; /SS low
            cbi     PortB,4
; WRITE COMMAND
            ldi     temp,WRITE
            out     SPDR,temp
wait_spi_w1:
            sbis    SPSR,SPIF           ; Transmission complete?
            rjmp    wait_spi_w1
            nop
            nop
            in      temp,SPDR           ; release SPIF here
            rcall   wait_10ms
; SET ADDRESS
            out     SPDR,address
wait_spi_w2:
            sbis    SPSR,SPIF           ; Transmission complete?
            rjmp    wait_spi_w2
            nop
            nop
            in      temp,SPDR           ; release SPIF here
            rcall   wait_10ms
; DATA BYTE
            out     SPDR,value
wait_spi_w3:
            sbis    SPSR,SPIF           ; Transmission complete?
            rjmp    wait_spi_w3
            nop
            nop
            in      temp,SPDR           ; release SPIF here
            rcall   wait_10ms
; /SS high
            sbi     PortB,4
            nop
            nop
            ret

;_____________________________________________________________________
erroroutput:
            ldi     address,TEC
            rcall   getbyte
            rcall   serout

            ldi     address,REC
            rcall   getbyte
            rcall   serout

            ldi     address,EFLG
            rcall   getbyte
            rcall   serout
```

```
            ret

;_____________________________________________________________________
wait_500ms:
;     2000000 Zyklen:
; ----------------------------
; warte 1999998 Zyklen:
            ldi     R17,$12
WGLOOP0v:   ldi     R18,$BC
WGLOOP1v:   ldi     R19,$C4
WGLOOP2v:   dec     R19
            brne    WGLOOP2v
            dec     R18
            brne    WGLOOP1v
            dec     R17
            brne    WGLOOP0v
; ----------------------------
; warte 2 Zyklen:
            nop
            nop
; ============================
            ret


;_____________________________________________________________________
wait_100ms:
;     400000 Zyklen:
; ----------------------------
; warte 399999 Zyklen:
            ldi     R17,$97
WGLOOP0s:   ldi     R18,$06
WGLOOP1s:   ldi     R19,$92
WGLOOP2s:   dec     R19
            brne    WGLOOP2s
            dec     R18
            brne    WGLOOP1s
            dec     R17
            brne    WGLOOP0s
; ----------------------------
; warte 1 Zyklus:
            nop
; ============================
            ret

;_____________________________________________________________________
wait_10ms:

;     40000 Zyklen:
; ----------------------------
; warte 39999 Zyklen:
            ldi     R17,$43
WGLOOP0:    ldi     R18,$C6
WGLOOP1:    dec     R18
            brne    WGLOOP1
            dec     R17
            brne    WGLOOP0
```

```
; ----------------------------
; warte 1 Zyklus:
            nop
; ============================
            ret

;_____________________________________________________________________
serout:
wait_ser:
            sbis    USR,UDRE            ; wait UDR
            rjmp    wait_ser
            out     UDR,temp            ; SPI-Data Register to UDR (sendi

            ret
```