

```

.include      "m8515def.inc"

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

.def          temp      =r16
.def          address    =r21
.def          value      =r22
.def          mask       =r23                                ; for BitModify

; 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

.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_500ms
rcall        wait_500ms
rcall        wait_500ms

```

```

; now SPI can be used in mcp2515

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

    ldi    temp,0b01010001    ; SPIEnabled, MasterMode, SPI Clo
    out    SPCR,temp

    sbi    PortB,4            ; /CS High
    nop
    nop
    nop
    nop
    nop

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

    rcall  mcp_reset
    rcall  wait_500ms

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

    ; CNF1:
    ldi    address,CNF1
    ldi    value,0x04
    rcall  sendbyte

    ; CNF2:
    ldi    address,CNF2
    ldi    value,0xB8
    rcall  sendbyte

    ; CNF3:
    ldi    address,CNF3
    ldi    value,0x05
    rcall  sendbyte

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

; ===== MCP2515 NORMAL MODE =====

    ; Normal Mode:
    ldi    address,CANCTRL
    ldi    value,0b00001000    ; OSMODE
    rcall  sendbyte

; MESSAGE TRANSMISSION (periodic)
send:

```

```

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

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

; define ID (Std. ID Low)
    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,0b11001100
    rcall  sendbyte

; Data to send:
    ldi    address,TXB0D1
    ldi    value,0b00110011
    rcall  sendbyte

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

; Transmission starts, when Bus is available.

; TRANSMIT-LED PB1 on
    sbi    PortB,3

; wait while send....
    rcall  wait_100ms

    cbi    PortB,3            ; LED PB1 off

; Interval Time:
    rcall  wait_500ms
    rcall  wait_500ms
    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
    in      temp, SPDR            ; release SPIF here

; SET ADDRESS
    out    SPDR, address

wait_spi_g2:
    sbis    SPSR, SPIF            ; Transmission complete?
    rjmp    wait_spi_g2
    in      temp, SPDR            ; release SPIF here

; DUMMY BYTE
    ldi    temp, 0b10101010
    out    SPDR, temp

wait_spi_g3:
    sbis    SPSR, SPIF            ; Transmission complete?
    rjmp    wait_spi_g3

; RESULT:
    in      temp, SPDR            ; release SPIF here

; /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
    in      temp, SPDR            ; release SPIF here

; SET ADDRESS
    out    SPDR, address

wait_spi_b2:
    sbis    SPSR, SPIF            ; Transmission complete?
    rjmp    wait_spi_b2
    in      temp, SPDR            ; release SPIF here

; MASK BYTE

```

```

        out        SPDR,mask
wait_spi_b3:
        sbis       SPSR,SPIF           ; Transmission complete?
        rjmp       wait_spi_b3
        in         temp,SPDR           ; release SPIF here
; BITS TO BE CHANGED
        out        SPDR,value
wait_spi_b4:
        sbis       SPSR,SPIF           ; Transmission complete?
        rjmp       wait_spi_b4
        in         temp,SPDR           ; release SPIF here
; /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
        in         temp,SPDR           ; release SPIF here
; /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
        in         temp,SPDR           ; release SPIF here
; SET ADDRESS
        out        SPDR,address
wait_spi_w2:
        sbis       SPSR,SPIF           ; Transmission complete?
        rjmp       wait_spi_w2
        in         temp,SPDR           ; release SPIF here
; DATA BYTE
        out        SPDR,value
wait_spi_w3:
        sbis       SPSR,SPIF           ; Transmission complete?
        rjmp       wait_spi_w3

```

```

        in      temp, SPDR          ; release SPIF here
; /SS high
        sbi     PortB, 4
        nop
        nop
        ret

; -----
wait_500ms:
;      2000000 Zyklen:
; -----
; warte 1999998 Zyklen:
        ldi     R17, $12
WGLOOP0: ldi     R18, $BC
WGLOOP1: ldi     R19, $C4
WGLOOP2: dec     R19
        brne    WGLOOP2
        dec     R18
        brne    WGLOOP1
        dec     R17
        brne    WGLOOP0
; -----
; 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

```

