

```

.include      "m8515def.inc"

.def          temp      =r16

; DEFINE INPUTs, OUTPUTs

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

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

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

; mcp Startup Delay
        rcall    wait_500ms

; now SPI can be used in mcp2515

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

; SPI Master Init
        ldi      temp,0b10100000    ; Output = SCK & MOSI
        out      DDRB,temp

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

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

; mcp Reset
        cbi      PortB,4            ; /CS pull down
        ldi      temp,0b11000000    ; RESET-Instruction
        rcall    spiout
        sbi      PortB,4            ; release /CS

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

;BIT TIMING:

; OSC = 16MHz
; Target CAN-Speed = 125.000 kbps
; TQ = 1ms
; #TQ = 8
; % Error = 0%
; Propagation Delay = 1 TQ
; Phase Segment1 = 3 TQ
; Phase Segment1 = 3 TQ

```

```

; SJW = 1 TQ

; CNF1:
    cbi    PortB,4           ; /CS pull down
    ldi    temp,0b00000010  ; WRITE-Instruction
    rcall  spiout
    ldi    temp,0b00101010  ; Register CNF1
    rcall  spiout
    ldi    temp,0b00000111  ; calculated value for CNF1
    rcall  spiout
    sbi    PortB,4           ; release /CS

; CNF2:
    cbi    PortB,4           ; /CS pull down
    ldi    temp,0b00000010  ; WRITE-Instruction
    rcall  spiout
    ldi    temp,0b00101001  ; Register CNF2
    rcall  spiout
    ldi    temp,0b10010000  ; calculated value for CNF2
    rcall  spiout
    sbi    PortB,4           ; release /CS

; CNF3:
    cbi    PortB,4           ; /CS pull down
    ldi    temp,0b00000010  ; WRITE-Instruction
    rcall  spiout
    ldi    temp,0b00101000  ; Register CNF3
    rcall  spiout
    ldi    temp,0b00000010  ; calculated value for CNF3
    rcall  spiout
    sbi    PortB,4           ; release /CS

; FILTER:

; Receive Buffer 0
    cbi    PortB,4           ; /CS pull down
    ldi    temp,0b00000010  ; WRITE-Instruction
    rcall  spiout
    ldi    temp,0b01100000  ; Register RXB0CTRL
    rcall  spiout
    ldi    temp,0b01100000  ; Receive all messages
    rcall  spiout
    sbi    PortB,4           ; release /CS

; Receive Buffer 1
    cbi    PortB,4           ; /CS pull down
    ldi    temp,0b00000010  ; WRITE-Instruction
    rcall  spiout
    ldi    temp,0b01100000  ; Register RXB1CTRL
    rcall  spiout
    ldi    temp,0b01100000  ; Receive all messages
    rcall  spiout
    sbi    PortB,4           ; release /CS

; MASK:

```

```

; MASKs
    cbi    PortB,4                ; /CS pull down
    ldi    temp,0b00000010       ; WRITE-Instruction
    rcall  spiout
    ldi    temp,0b00100000       ; Register RXM0SIDH
    rcall  spiout
    ldi    temp,0b00000000       ; Delete Mask, Receive all messag
    rcall  spiout
    sbi    PortB,4                ; release /CS

; MASKs
    cbi    PortB,4                ; /CS pull down
    ldi    temp,0b00000010       ; WRITE-Instruction
    rcall  spiout
    ldi    temp,0b00100001       ; Register RXM0SIDL
    rcall  spiout
    ldi    temp,0b00000000       ; Delete Mask, Receive all messag
    rcall  spiout
    sbi    PortB,4                ; release /CS

; MASKs
    cbi    PortB,4                ; /CS pull down
    ldi    temp,0b00000010       ; WRITE-Instruction
    rcall  spiout
    ldi    temp,0b00100010       ; Register RXM0EID8
    rcall  spiout
    ldi    temp,0b00000000       ; Delete Mask, Receive all messag
    rcall  spiout
    sbi    PortB,4                ; release /CS

; MASKs
    cbi    PortB,4                ; /CS pull down
    ldi    temp,0b00000010       ; WRITE-Instruction
    rcall  spiout
    ldi    temp,0b00100011       ; Register RXM0EID0
    rcall  spiout
    ldi    temp,0b00000000       ; Delete Mask, Receive all messag
    rcall  spiout
    sbi    PortB,4                ; release /CS

; MASKs
    cbi    PortB,4                ; /CS pull down
    ldi    temp,0b00000010       ; WRITE-Instruction
    rcall  spiout
    ldi    temp,0b00100100       ; Register RXM1SIDH
    rcall  spiout
    ldi    temp,0b00000000       ; Delete Mask, Receive all messag
    rcall  spiout
    sbi    PortB,4                ; release /CS

; MASKs
    cbi    PortB,4                ; /CS pull down
    ldi    temp,0b00000010       ; WRITE-Instruction
    rcall  spiout
    ldi    temp,0b00100101       ; Register RXM1SIDL
    rcall  spiout
    ldi    temp,0b00000000       ; Delete Mask, Receive all messag

```

```

rcall    spiout
sbi      PortB,4           ; release /CS

; MASKs
cbi      PortB,4           ; /CS pull down
ldi      temp,0b00000010   ; WRITE-Instruction
rcall    spiout
ldi      temp,0b00100110   ; Register RXM1EID8
rcall    spiout
ldi      temp,0b00000000   ; Delete Mask, Receive all messag
rcall    spiout
sbi      PortB,4           ; release /CS

; MASKs
cbi      PortB,4           ; /CS pull down
ldi      temp,0b00000010   ; WRITE-Instruction
rcall    spiout
ldi      temp,0b00100111   ; Register RXM1EID0
rcall    spiout
ldi      temp,0b00000000   ; Delete Mask, Receive all messag
rcall    spiout
sbi      PortB,4           ; release /CS

; PIN FUNCTIONS:

; High Impedance on RXnBF-Pins
cbi      PortB,4           ; /CS pull down
ldi      temp,0b00000010   ; WRITE-Instruction
rcall    spiout
ldi      temp,0b00001100   ; Register BFPCTRL
rcall    spiout
ldi      temp,0b00000000   ; Delete Mask, Receive all messag
rcall    spiout
sbi      PortB,4           ; release /CS

; RXnRTS-Pins = Input
cbi      PortB,4           ; /CS pull down
ldi      temp,0b00000010   ; WRITE-Instruction
rcall    spiout
ldi      temp,0b00001101   ; Register TXRTSCTRL
rcall    spiout
ldi      temp,0b00000000   ; Delete Mask, Receive all messag
rcall    spiout
sbi      PortB,4           ; release /CS

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

; activate NORMAL MODE
cbi      PortB,4           ; /CS pull down
ldi      temp,0b00000101   ; BIT-Modify-Instruction
rcall    spiout
ldi      temp,0b00001111   ; Register CANCTRL
rcall    spiout
ldi      temp,0b11101000   ; MASK (Bits to be changed)
rcall    spiout

```

```

        ldi        temp,0b00001000        ; Set Normal Mode, One Shot Mode
        rcall     spiout
        sbi        PortB,4                ; release /CS

; MESSAGE TRANSMISSION (always send 0b11001100), One Shot Mode
loop_send:
; define ID (Std. ID High)
        cbi        PortB,4                ; /CS pull down
        ldi        temp,0b00000010       ; WRITE-Instruction
        rcall     spiout
        ldi        temp,0b00110001       ; Register TXB0SIDH
        rcall     spiout
        ldi        temp,0b00000000       ; however here 0
        rcall     spiout
        sbi        PortB,4                ; release /CS

; define ID (Std. ID Low)
        cbi        PortB,4                ; /CS pull down
        ldi        temp,0b00000010       ; WRITE-Instruction
        rcall     spiout
        ldi        temp,0b00110010       ; Register TXB0SIDL
        rcall     spiout
        ldi        temp,0b00001000       ; value greater than 0 (not most
        rcall     spiout
        sbi        PortB,4                ; release /CS

; define Data Length and Remote/Data Frame
        cbi        PortB,4                ; /CS pull down
        ldi        temp,0b00000010       ; WRITE-Instruction
        rcall     spiout
        ldi        temp,0b00110101       ; Register TXB0DLC
        rcall     spiout
        ldi        temp,0b00000001       ; Data-Frame, 1 Data-Byte
        rcall     spiout
        sbi        PortB,4                ; release /CS

; Data to send:
        cbi        PortB,4                ; /CS pull down
        ldi        temp,0b00000010       ; WRITE-Instruction
        rcall     spiout
        ldi        temp,0b00110110       ; Register TXB0D0
        rcall     spiout
        ldi        temp,0b11001100       ; Data to send <<-----
        rcall     spiout
        sbi        PortB,4                ; release /CS

; Send...:
        cbi        PortB,4                ; /CS pull down
        ldi        temp,0b10000001       ; Send, Buffer 0 RTS Signal
        rcall     spiout
        sbi        PortB,4                ; release /CS

; Transmission starts, when Bus is available.

; wait while send....
        rcall     wait_500ms

```

```
    rjmp    loop_send
```

```
wait_500ms:
; =====
;   Warteschleifen-Generator
;   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
```

```
; SPI Transmit
```

```
spiout:    out    SPDR,temp
wait_spi:  sbis    SPSR,SPIF          ; Transmission complete?
          rjmp   wait_spi
          ret                ; back...
```

