

```

.include      "m8515def.inc"

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

.def          temp      =r16
.def          temp2     =r21

.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                ; EEEEEEEEE Input (from PLC) Level

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

; mcp Startup Delay
    rcall      wait_500ms

; now SPI can be used in mcp2515

; SPI Master Init
    ldi        temp,0b10110011          ; Output = SCK & MOSI & /SS & LED
    out        DDRB,temp

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

    sbi        PortB,4                  ; /CS auf High (kein Slave angewä

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

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

    rcall      wait_500ms

; ===== 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,0x07              ; 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,0x90              ; 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,0x02              ; 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,0b01110000       ; 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,0b00000010       ; WRITE-Instruction
        rcall    spiout
        ldi      temp,0b00001111       ; Register CANCTRL
        rcall    spiout
        ldi      temp,0b00001000       ; Set Normal Mode
        rcall    spiout
        sbi      PortB,4                ; release /CS

```

```

        sbi      PortB,0                ; LED PB0 on          MAIN LED

; MESSAGE TRANSMISSION (periodic)
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 (XXXXXXXX)
        rcall    spiout
        ldi      temp,0b00000100        ; however here 100
        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 (XXXdddddd)
        rcall    spiout
        ldi      temp,0b00000000        ; however here 0
        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
        in       temp,PinA              ; 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.

        sbi      PortB,1                ; LED PB1 on          MAIN LED

; wait while send....
        rcall    wait_100ms            ; LED On Time

```

```

; clear TX0IF Flag (Buffer 0 Empty Flag):
;
;         cbi PortB, 4                ; /CS pull down
;         ldi temp, 0b00000101        ; BIT-Modify-Instruction
;         rcall spiout
;         ldi temp, 0b00101100        ; Register CANINTF
;         rcall spiout
;         ldi temp, 0b00000100        ; MASK (Bits to be change
;         rcall spiout
;         ldi temp, 0b00000000        ; Release Interrupt
;         rcall spiout
;         sbi PortB, 4                ; release /CS

```

wait_TX_empty:

; Verify that Message was send successfully:

; If TXB0CTRL.TXREQ clear, then success

```

        cbi    PortB,4                ; /CS pull down
        ldi    temp,0b00000011        ; READ-Instruction
        rcall   spiout
        ldi    temp,0b00110000        ; TXB0CTRL
        rcall   spiout
        ldi    temp,0b10011111        ; Dummy Byte
        rcall   spiout
        sbi    PortB,4                ; release /CS

```

;temp has content of TXB0CTRL

```

        sbrc    temp,3                ; Test on TXREQ
        rjmp    wait_TX_empty

```

```

        cbi    PortB,1                ; LED PB1 off      MAIN LED

```

; Interval Time:

```

        rcall   wait_500ms
        rcall   wait_500ms
        rcall   wait_500ms

        rjmp    loop_send

```

; ===== Delay =====

wait_500ms:

; =====

; Warteschleifen-Generator

; 800000 Zyklen:

; -----

; warte 799995 Zyklen:

```

        ldi    R17,$5F
WGLOOP0: ldi    R18,$17
WGLOOP1: ldi    R19,$79
WGLOOP2: dec    R19
        brne   WGLOOP2
        dec    R18
        brne   WGLOOP1
        dec    R17

```

```

                brne      WGLOOP0
; -----
; warte 3 Zyklen:
                ldi       R17,$01
WGLOOP3:        dec       R17
                brne      WGLOOP3
; -----
; warte 2 Zyklen:
                nop
                nop
; =====
                ret

; ===== SPI Transmit =====
spiout:

                out       SPDR,temp

wait_spi:       sbis      SPSR,SPIF          ; Transmission complete?
                rjmp      wait_spi
; SPIF is set

                in        temp2,SPSR
                in        temp2,SPDR          ; release SPIF by reading Register
                ret                          ; back to programm..

wait_100ms:
; =====
; Warteschleifen-Generator
; 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

```

