

```

.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
    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

; ===== 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 =====

; CNF1:
    cbi        PortB,4            ; /CS pull down
    ldi        temp,0b00000010    ; WRITE-Instruction
    rcall      spiout

```

```

        ldi        temp,0b00101010        ; Register CNF1
        rcall      spiout
        ldi        temp,0x04                ; 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,0xB8                ; 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,0x05                ; calculated value for CNF3
        rcall      spiout
        sbi        PortB,4                ; release /CS

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

; Normal Mode:
        cbi        PortB,4                ; /CS pull down
        ldi        temp,0b00000010        ; WRITE-Instruction
        rcall      spiout
        ldi        temp,0x0F                ; Register CANCTRL
        rcall      spiout
        ldi        temp,0b00001000        ; OSM
        rcall      spiout
        sbi        PortB,4                ; release /CS

; Interrupts
        cbi        PortB,4                ; /CS pull down
        ldi        temp,0b00000010        ; WRITE-Instruction
        rcall      spiout
        ldi        temp,0x2b                ; Register CANINTE
        rcall      spiout
        ldi        temp,0b00000100        ; Transmit Buffer 0 Empty Flag En
        rcall      spiout
        sbi        PortB,4                ; release /CS

; MESSAGE TRANSMISSION (periodic)
send:

; clear TXREQ-Flag manually

```

```

        cbi      PortB,4                ; /CS pull down
        ldi      temp,0b00000010      ; WRITE-Instruction
        rcall    spiout
        ldi      temp,0x2C             ; Register CANINTF
        rcall    spiout
        ldi      temp,0b00000000      ; clear Transmit Buffer 0 Empty F
        rcall    spiout
        sbi      PortB,4                ; release /CS

; 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,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 (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,0b00000010      ; Data-Frame, 2 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

; Data to send:
        cbi      PortB,4                ; /CS pull down
        ldi      temp,0b00000010      ; WRITE-Instruction
        rcall    spiout
        ldi      temp,0b00110111      ; Register TXB0D1
        rcall    spiout
        ldi      temp,0b00110011      ; 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,3                ; LED PB1 on

; 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

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

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
; 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

```