

```

.include      "m8515def.inc"

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

.def         temp      =r16
.def         temp2     =r21
.equ         CLOCK     =4000000
.equ         BAUD      =2400
.equ         UBRRVAL   =CLOCK/(BAUD*16)-1

.org         0x0000
rjmp        main

.org         0x0001
rjmp        interrupt0

interrupt0:
; Message on RX0B:

        sbi      PortB,4          ; release /CS

        cbi      PortB,4          ; /CS pull down
        ldi      temp,0b00000011  ; READ-Instruction
        rcall    spiout
        ldi      temp,0b01100110  ; RXB0D0
        rcall    spiout
        ldi      temp,0b10101010  ; Dummy Byte
        rcall    spiout
        sbi      PortB,4          ; release /CS

; SPDR = temp2 now holds the received data

; USART Transmission

serout:
        sbis     UCSRA,UDRE        ; wait UDR
        rjmp     serout
        out      UDR,temp2         ; SPI-Data Register to UDR (sendi

; clear RX0BF Flag:

        cbi      PortB,4          ; /CS pull down
        ldi      temp,0b00000101  ; BIT-Modify-Instruction
        rcall    spiout
        ldi      temp,0b00101100  ; Register CANINTF
        rcall    spiout
        ldi      temp,0b00000001  ; MASK (Bits to be changed)
        rcall    spiout
        ldi      temp,0b00000000  ; Release Interrupt
        rcall    spiout
        sbi      PortB,4          ; release /CS

        reti

main:
; Stackpointer init

```

```

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

; Port A output
        ldi    temp,0xFF
        out    DDRA,temp

; Port D input
        ldi    temp,0x00
        out    DDRD,temp

; enable Interrupts on INT0-Pin (PD2)
        ldi    temp,0b00000010    ; INT0 = falling Edge
        in     temp,GICR
        ori    temp,0b01000000    ; INT0 INT enable
        out    GICR,temp          ; General INT Control Register

; mcp Startup Delay
        rcall  wait_500ms
        rcall  wait_500ms

; now SPI can be used in mcp2515

; Baudrate einstellen
        ldi    temp,LOW(UBRRVAL)
        out    UBRRL,temp
        ldi    temp,HIGH(UBRRVAL)
        out    UBRRH,temp

; Frame-Format: 8 Bit
        ldi    temp,(1<<URSEL)|(3<<UCSZ0)
        out    UCSRC,temp

        sbi    UCSRB,TXEN          ; activate TX

; SPI Master Init
        ldi    temp,0b10110000    ; Output = SCK & MOSI & /SS
        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 =====

```

```

;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
        sbi        PortA,0                ; LED PA0 on            MAIN LED

; 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

; INTERRUPTS

; Receive Buffer 0 & 1 INT Enable
        cbi        PortB,4                ; /CS pull down
        ldi        temp,0b00000010        ; WRITE-Instruction
        rcall      spiout
        ldi        temp,0b00101011        ; Register CANINTE
        rcall      spiout
        ldi        temp,0b10100001        ; Set RX0IE = 1 & Errors
        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,0b00100000       ; Receive messages with standard
    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,0b01000000       ; Receive messages with ext. ID (
    rcall  spiout
    sbi    PortB,4                ; release /CS

; define Filter 0
    cbi    PortB,4                ; /CS pull down
    ldi    temp,0b00000010       ; WRITE-Instruction
    rcall  spiout
    ldi    temp,0b00000000       ; Register RXF0SIDH
    rcall  spiout
    ldi    temp,0b000000100      ; Receive messages with ID_std_hi
    rcall  spiout
    sbi    PortB,4                ; release /CS

; MASKS (each Receive Buffer has 4 Masks (ID_std_high, ID_std_low, ID_ext

; MASKs
    cbi    PortB,4                ; /CS pull down
    ldi    temp,0b00000010       ; WRITE-Instruction
    rcall  spiout
    ldi    temp,0b00100000       ; Register RXM0SIDH
    rcall  spiout
    ldi    temp,0b11111111       ;
    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       ; Receive all
    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

    sbi    PortA,1

; 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:

; RXnBF-Pins
        cbi      PortB,4            ; /CS pull down
        ldi      temp,0b00000010    ; WRITE-Instruction
        rcall    spiout
        ldi      temp,0b00001100    ; Register BFPCTRL
        rcall    spiout
        ldi      temp,0b00000101    ; /RX0BF Pin = Interrupt Pin
        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    ; Inputs
        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,0b11100000    ; MASK (Bits to be changed)
        rcall    spiout
        ldi      temp,0b00000000    ; Set Normal Mode
        rcall    spiout
        sbi      PortB,4            ; release /CS

        sbi      PortA,2

        sei                        ; Global INT enable

loop:    rjmp     loop              ; wait for Interrupt.....

; ===== 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..

; =====

; ===== 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
; =====

```