```
.include    "m8515def.inc"

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

.def        temp    =r16
.def        temp2   =r21
.equ        CLOCK   =4000000
.equ        BAUD    =9600
.equ        UBRRVAL =CLOCK/(BAUD*16)-1

.org        0x0000




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 C output
            ldi     temp,0xFF
            out     DDRC,temp

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

; mcp Startup Delay
            rcall   wait_500ms
            rcall   wait_500ms

; now SPI can be used in mcp2515

; Set up baud rate
            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 ==================================
start:

; 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

; INTERRUPTS

; Receive Buffer INT Enable
            cbi     PortB,4             ; /CS pull down
            ldi     temp,0b00000010     ; WRITE-Instruction
            rcall   spiout
            ldi     temp,0b00101011     ; Register CANINTE
            rcall   spiout
```

```
            ldi     temp,0b00000001     ; Set RX0IE = 1
            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


; ===================== 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,0b00000000     ; Normal
            rcall   spiout
            sbi     PortB,4             ; release /CS

poll_RX:


; 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



; Poll on Receive Flag:

; Reading Flag
            cbi     PortB,4             ; /CS pull down
            ldi     temp,0b00000011     ; READ-Instruction
            rcall   spiout
            ldi     temp,0b00101100     ; CANINTF
            rcall   spiout
;                   ldi temp, 0b01010101        ; CPHA = 1 (Sampling on f
```

```
;                   out SPCR, temp
                ldi     temp,0b10101010     ; Dummy Byte
                rcall   spiout
;                   ldi temp, 0b01010001        ; CPHA = 0
;                   out SPCR, temp
                sbi     PortB,4             ; release /CS

                sbrs    temp2,0             ; RX0IF set?
                rjmp    poll_RX             ; no, keep on waiting...


; Message on RX0B:

; Receive-LED
                sbi     PortA,0             ; LED On

                sbi     PortB,4             ; release /CS

; Reading RXB0D0
                cbi     PortB,4             ; /CS pull down
                ldi     temp,0b00000011     ; READ-Instruction
                rcall   spiout
                ldi     temp,0b01100110     ; RXB0D0
                rcall   spiout
;                   ldi temp, 0b01010101        ; CPHA = 1 (Sampling on f
;                   out SPCR, temp
                ldi     temp,0b10101010     ; Dummy Byte
                rcall   spiout
;                   ldi temp, 0b01010001        ; CPHA = 0
;                   out SPCR, temp
                sbi     PortB,4             ; release /CS

; SPDR = temp2 now holds the received data
                rcall   serout

; Reading RXB0D1
                cbi     PortB,4             ; /CS pull down
                ldi     temp,0b00000011     ; READ-Instruction
                rcall   spiout
                ldi     temp,0b01100111     ; RXB0D1
                rcall   spiout
;                   ldi temp, 0b01010101        ; CPHA = 1 (Sampling on f
;                   out SPCR, temp
                ldi     temp,0b10101010     ; Dummy Byte
                rcall   spiout
;                   ldi temp, 0b01010001        ; CPHA = 0
;                   out SPCR, temp
                sbi     PortB,4             ; release /CS

; SPDR = temp2 now holds the received data

                rcall   serout

                rcall   wait_100ms

                cbi     PortA,0             ; LED Off
```

```
            rjmp    poll_RX



; ========================= 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 Registe

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


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


; USART Transmission

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