

```
; Datei: RFM12_Empfaenger.asm
.include <m8def.inc>
.cseg
```

```
/*
R16:    temp1
R17:    temp2
R18:    Empfangendes High-Byte v. SPI-Schnittstelle
R19:    Empfangendes Low-Byte v. SPI-Schnittstelle
R20:    CMD_HighByte
R21:    CMD_LowByte
R22:    Daten High-Byte
R23:    Daten Low-Byte
*/
```

Init:

```
    ldi r16, LOW(RAMEND)
    out spl, r16
    ldi r16, HIGH(RAMEND)
    out sph, r16
    ; IO-Init + Pull-Ups
    clr r16
    out ddrC, r16          ; PC zur Eingabe
    ser r16
    out portC, r16        ; Pull-Ups der Eingänge
    out ddrD, r16        ; PD zur Ausgabe
    ; SPI-Init
    ldi r16, 0b00101100   ; MOSI=1, SS=1, SCK=1
    out ddrB, r16
    out portB, r16
    ldi r16, 0b01010011   ; SPCR: SPIE(0), SPE(1), DORD(0), MS
TR(1), CPOL(0), CPHA(0), SPR1(1), SPR0(1)
    out sPCR, r16
    ; RFM12-Init
    ; 0x80d8
    ldi r20, 0x80
    ldi r21, 0xd8        //d
    rcall WriteCMD
    ; 0x82d8
    ldi r20, 0x82
    ldi r21, 0xd8
```

```
rcall WriteCMD
;0xA640
ldi r20,0xA6
ldi r21,0x40
rcall WriteCMD
;0xC647
ldi r20,0xC6
ldi r21,0x47
rcall WriteCMD
;0x94A0
ldi r20,0x94
ldi r21,0xA0
rcall WriteCMD
;0xC2AC
ldi r20,0xC2
ldi r21,0xAC
rcall WriteCMD
;0xCA80
ldi r20,0xCA
ldi r21,0x80
rcall WriteCMD
;0xCA83
ldi r20,0xCA
ldi r21,0x83
rcall WriteCMD
;0xC49B
ldi r20,0xC4
ldi r21,0x9B
rcall WriteCMD
;0x9850
ldi r20,0x98
ldi r21,0x50
rcall WriteCMD
;0xE000
ldi r20,0xE0
ldi r21,0x00
rcall WriteCMD
;0xC800
ldi r20,0xC8
ldi r21,0x00
rcall WriteCMD
```

```
; 0xC000
ldi r20, 0xC0
ldi r21, 0x00
rcall WriteCMD
```

Start:

```
nIRQ:
    sbic pin_c, 1 ; nIRQ abfrage
    rjmp start
rcall RF12_RDFIFO
; 0xCA80
ldi r20, 0xCA
ldi r21, 0x81 //80
rcall WriteCMD
; 0xCA83
ldi r20, 0xCA
ldi r21, 0x83
rcall WriteCMD

    rjmp start
```

wort_out:

```
out portd, r18
rcall zeit_1s
rcall zeit_1s
rcall zeit_1s
out portd, r19
rcall zeit_1s
```

RF12_RDFIFO:

```
; 0x0000
ldi r20, 0x00
ldi r21, 0x00
rcall WriteCMD
; 0xB000
ldi r20, 0xB0
ldi r21, 0x00
rcall WriteCMD
rcall wort_out
mov r22, r18 ; Sicherung der Datenbytes
mov r23, r19
ret
```

WriteFSKbyte:

```
push r21
push r20
ldi r20, 0x00
ldi r21, 0x00
rcall WriteCMD
pop r20
pop r21
ldi r20, 0xB8
rcall WriteCMD
ret
```

WriteCMD:

```
cbi portb, 2 ; /SS PB2 ->"low"
out SPDR, r20
Warte_Senden:
    sbis SPSR, SPIF ; warte bis high-Byte übertragen
ist
    rjmp Warte_Senden
in r18, SPDR ; r18 s.o.
out SPDR, r21
Warte_Senden_1:
    sbis SPSR, SPIF ; warte bis low-Byte übertragen i
st
    rjmp Warte_Senden_1
in r19, SPDR ; r19 s.o.
sbi portb, 2
ret
```

zeit_100ms:

```
push r16
push r17
push r18
in r16, sreg
push r16
ldi r16, 0x05 ; 100ms 0x19
zeit_100ms_1:
ldi r17, 0xcd ; 20ms
```

```
zeit_100ms_2:
    ldi r18, 0x1d    ; 100µs
zeit_100ms_3:
    dec r18
    brne zeit_100ms_3
    dec r17
    brne zeit_100ms_2
    dec r16
    brne zeit_100ms_1
    pop r16
    out sreg, r16
    pop r18
    pop r17
    pop r16
    ret
```

```
zeit_1s:
    push r16
    push r17
    push r18
    in r16, sreg
    push r16
    ldi r16, 0x32    ; 1s
zeit_1s_1:
    ldi r17, 0xcd    ; 20ms
zeit_1s_2:
    ldi r18, 0x1d    ; 100µs
zeit_1s_3:
    dec r18
    brne zeit_1s_3
    dec r17
    brne zeit_1s_2
    dec r16
    brne zeit_1s_1
    pop r16
    out sreg, r16
    pop r18
    pop r17
    pop r16
    ret
```