

aha

```
#include 8051.h
        .org 0000h
scl      .equ    P3.0           ;Festlegen von SCL
sda      .equ    P3.1           ;Festlegen von SDA
eeprom1 .equ    10100010b      ;Festlegen der adresse eeprom1-4
eeprom2 .equ    10100100b
eeprom3 .equ    10100110b
eeprom4 .equ    10100000b

.org 0h
        ljmp start
start:   ;Startbedingung
        setb scl                ;SCL auf High setzen
        setb sda                ;SDA auf High setzen

        ljmp main                ;Sprung zu main

I2C_start:
        setb sda                ;SDA auf High
        setb scl                ;SCL auf High

        jnb sda, I2C_x40        ;SDA auf High sonst Sprung
        jnb scl, I2C_x40        ;SCL auf High sonst Sprung

        clr sda                 ;SDA auf Low ziehen
        nop
        nop
        clr scl                ;SCL auf Low ziehen
        clr c                   ;löscht das Carry
        ajmp I2C_x41

I2C_x40:
        setb c                 ;Carry(Fehlermeldung) wird gesetzt
I2C_x41:
        ret

I2C_stop:
        clr sda                ;SDA auf Low
        nop
        nop
        setb scl                ;SCL auf High

        nop
        nop
        nop
        nop
        setb sda                ;SDA auf High
        ret

I2C_Byte_senden:
        push B                  ;rettet den Inhalt von B
        mov B, #9b

I2C_x42:
        rlc a                   ;Inhalt des Akku über´s Carry
        mov sda, c              ;Lade SDA mit dem Inhalt Carry
        setb scl                ;SCL auf High

        nop
        nop
        nop
        clr scl                ;SCL auf Low
        djnz B, I2C_x42        ;Sprung zu I2C_x42 wenn B nicht 0
                                ;erhalten des Acknowledge Bit vom

slave
```

```

aha
setB sda ;SDA auf High
nop
setB scl ;SCL auf Low
nop
nop
nop
nop
mov c, sda
clr scl ;SCL auf Low
pop B ;wiederherstellen von B
ret ;return

I2C_Byte_empfangen:
setB sda ;SDA auf High
push B ;rettet den Inhalt von B
mov B, #9b
I2C_x43:
nop
nop
setB scl ;SCL auf High
nop
nop
mov c, sda ;Lade Carry mit dem Inhalt von SDA
rlc a ;Verschiebe Akku nach links über´s
Carry
clr scl ;SCL auf Low
djnz B, I2C_x43 ;Sprung zu I2C_x43 wenn B nicht 0
ist
pop B ;wiederherstellen von B
ret

I2C_ack:
clr sda ; Senden eines Low Acknowledge
nop
setb scl ;SCL auf High
nop
nop
nop
nop
clr scl ;SCL auf Low
ret

I2C_nak:
setB sda ;SDA auf High
nop
setB scl ;SCL auf High
nop
nop
nop
nop
clr scl ;SCL auf Low
ret

Byte_schreiben:
lcall I2C_start ;I2C starten
jc eepromFehler ;Kontrolle ob alles in Ordnung
PUSH ACC ; Akku retten
mov A, eeprom4 ;laden Akku mit Bausteinadresse
lcall I2C_Byte_senden ; Senden der Bausteinadresse

jc eepromFehler ;Kontrolle
lcall I2C_ack ;sende Acknowledge
mov a, r1 ;erstes Byte der Adresse laden
lcall I2C_Byte_senden ;1. AdressByte senden
lcall I2C_ack ;sende Acknowledge
jc eepromFehler ;Kontrolle

```

```

                                aha
mov a,r2                        ;zweites Byte der Adresse laden
lcall I2C_Byte_senden          ;2. Adressbyte senden
lcall I2C_ack                  ;sende Acknowledge
jc eepromFehler               ;Kontrolle
mov a,r3                        ;Datenbyte laden
lcall I2C_Byte_senden          ;Datenbyte senden
lcall I2C_ack                  ;sende Acknowledge
jc eepromFehler               ;Kontrolle
pop ACC                         ;wiederherstellen vom Akku
lcall I2C_stop
ret

;eepromFehler:                ;Stoppen der I2C Routine
LCALL I2C_stop
RET

Byte_lesen:                     ;Lesen eines Bytes
LCALL I2C_start
jc eepromFehler               ;Kontrolle
PUSH ACC                        ;Akku retten
mov A,eeprom4                  ;Lade Akku mit Bausteinadresse
lcall I2C_Byte_senden          ;senden der Bausteinadresse
lcall I2C_ack                  ;sende Acknowledge
jc eepromFehler               ;Kontrolle
mov a,r1                        ;1 Wortadresse
lcall I2C_Byte_senden          ;1 Wortadresse senden
lcall I2C_ack                  ;sende Acknowledge
jc eepromFehler               ;Kontrolle
mov a,r2                        ;2 Wortadresse
lcall I2C_Byte_senden          ;2 Wortadresse senden
lcall I2C_ack                  ;sende Acknowledge
jc eepromFehler               ;Kontrolle
LCALL I2C_start
mov A,eeprom4                  ;Lade Akku mit Bausteinadresse
inc a                           ;Inkrementieren da lesen bei 1
lcall I2C_Byte_senden          ;Datenadresse senden
lcall I2C_ack                  ;sende Acknowledge
jc eepromFehler               ;Kontrolle
lcall I2C_Byte_empfangen       ;Byte zurückgeben
lcall I2C_nack
jc eepromFehler
mov r3,a
pop ACC
lcall I2C_stop
ret

main:
mov r1,#11h
mov r2,#22h
mov r3,#0FFh
acall Byte_schreiben           ;Sprung zu Byte schreiben
mov r1,#33h
mov r2,#44h
mov r3,#22h
acall Byte_schreiben           ;Sprung zu Byte schreiben
mov r1,#55h
mov r2,#66h
mov r3,#44h
acall Byte_schreiben           ;Spring zu Byte schreiben

mov r1,#12h
mov r2,#14h
acall Byte_lesen
mov P0,r3

```

.end

aha