

Hardware-Praktikum

Versuch 5

Alle Korrekturen die während des Versuchs durchgeführt wurden sind grün markiert.

1b)

| # | address | Byte 1 | Byte 2 | Byte 3 | Byte 4 | |
|---------------|-------------|------------|-----------|------------|-----------|--|
| CLR R0 | : 0000.0000 | 0000.01.00 | | | | |
| CLR R1 | : 0000.0001 | 0000.01.01 | | | | |
| LD R0,(0xFC): | 0000.0010 | 1111.11.11 | 1111.1100 | 0001.00.00 | | #Lade den Wert an FC in Register 0 |
| LD R1,(0xFD): | 0000.0101 | 1111.11.11 | 1111.1101 | 0001.00.01 | | #Lade den Wert an FD in Register 1 |
| ADD R0, R1 | : 0000.1000 | 0110.01.00 | | | | #R0 = R0 + R1, Ohne Carry! (Carry: 0111.01.00) |
| ST(FF), R0 | : 0000.1001 | 1111.00.00 | | 0001.11.11 | 1111.1111 | #Speichere den Inhalt von Register 0 in FF |
| ST(F0), R0 | : 0000.1100 | 1111.00.00 | | 0001.11.11 | 1111.0000 | #Speichere den Inhalt von Register 0 in DAC1 |
| JMP 0 | : 0000.1111 | 1111.10.11 | 0000.0000 | 0001.00.11 | | #Springe zurück an Adresse 0 |

2a)

```

#! mrasm
;Rampe auf DAC1 ausgeben:

.ORG 0
CLR R0
;Beginne an Adr 0
;Register 0 Loeschen

LOOP:
;Start Schleife
ST (0xFF),R0
;Speichere den Inhalt von Register 0 in FF
ST (0xF0),R0
;Speichere den Inhalt von Register 0 in DAC1
INC R0
;Inkrementiere den Inhalt von R0
JR LOOP
;Sprung zu Schleifenstart

```

2b)

```

#! msram
;Messe die Spannung am
Temperatursensor und gib sie in R0 aus

.ORG 0
;Beginne an Adresse 0

MAIN:
;Mainfunktion
LDSP 0xEF
CALL TEMPERATUR
;Springe zu Subroutine "TEMPERATUR"
ST (0xFF),R0
;Speichere den Inhalt von Register 0 in FF: Gibt
die aktuelle Temp aus
JR MAIN
;Springe zu "MAIN": Startet die Messung erneut

TEMPERATUR:
;Subroutine zur Bestimmung der Temperatur
LD R0,0xFF
;Lade die Konstante FF (255) in R0, "0x"
interpretiert Zahlen in Hexadezimal
LD R2,0x80
;Lade die Konstante 80 (128) in R2
LOOP:
;Schleife fuer die sukzessiver
Approximation
ST (0xF1),R0
;Speichere den Inhalt von Register 0 in F1
LD R1,(0xF1)
;Lade den Inhalt von F1 in R1
LSR R1
;Fuenf mal R1 nach rechts shiften, nun steht im
Carry das Bit 4,
LSR R1
;welches den Zustand des Komperators angibt
LSR R1
LSR R1
LSR R1
JCS HIGHER
;Springe zu HIGHER falls carry flag set (Temp
hoeher)
;Fuehre weiter aus falls nicht (Temp
niedriger)
SUB R0,R2
;R0 = R0 - R2

```

```

        LSR R2                                ;R2 nach rechts shiften, "(R2)/2"
        JZC LOOP                             ;Jump an den Beginn der Loop, falls letztes Bit keine 0
        RET
HIGHER:                                ;Wird nicht immer ausgeführt, nur als Jump
        ADD R0,R2                            ;R0 = R0 + R2
        LSR R2                                ;R2 nach rechts shiften, "(R2)/2"
        JZC LOOP                             ;Jump an den Beginn der Loop, falls letztes Bit
keine 0
        RET

```

2c)

#! mrasm

.ORG 0

;Hauptprogramm zur

Lüftersteuerung

MAIN:

```

        LDSP 0xEF                            ; Stackpointer auf EF setzen
        CALL TEMPERATUR                     ;Springe zu Subroutine "TEMPERATUR"
        LD R1,(0xFE)
        LD R2,(0xFF)
        CMP R0,R1                          ;Vergleiche R0 und R1 (R0 - R1)
        JNS A                               ;Falls negativ Springe zu A
        CMP R0,R2                          ;R0 - R2
        JNS B                               ;Falls negativ Springe zu A
        LD R1,(0xFC)
        JR C
A:
        LD R1,0
        JR C
B:
        LD R1,(0xFD)
        JR C
C:
        ST (0xF0),R1
        ST (0xFF),R1
        ST (0xFE),R0
        JR MAIN                            ;Dauerschleife

```

; Aufgabe 2b

TEMPERATUR:

```

        LD R0,0xFF
        LD R2,0x80
LOOP:
        ST (0xF1),R0
        LD R1,(0xF1)
        LSR R1
        LSR R1
        LSR R1
        LSR R1
        LSR R1
        JCS HIGHER
        SUB R0,R2
        LSR R2
        JZC LOOP
        RET
HIGHER:
        ADD R0,R2
        LSR R2
        JZC LOOP
        RET

```