Gewerbliche Schule Schwäbisch Hall

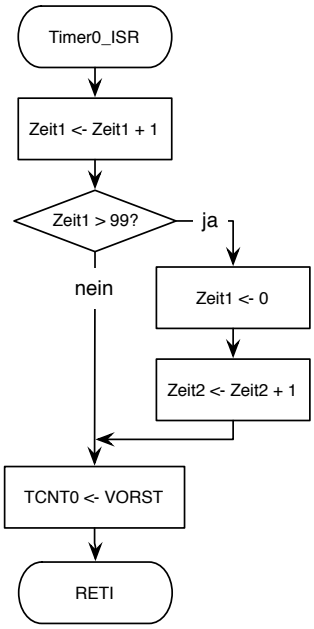
**Klassenarbeit IT 29.06.2016 Zeit 80 min TG12-B Name _____**

# Zeitwächter

Ein ATtiny2313 @ 1MHz soll als Zeitwächter für 500 ms verwendet werden. Der externe Interrupt INT0 startet und stoppt bei negativer Flanke den Timer0. Der Timer steht zunächst, bekommt keinen Takt. Nach dem Start des Timers in der INT0_ISR wird die Timer0_ISR durch einen OverflowInterrupt alle 100 $\mu$s aufgerufen und zählt zwei Register Zeit1 und Zeit2 hoch. Im Hauptprogramm werden die Registerinhalte laufend überprüft und wenn der Wert 500 ms erreicht ist, wird der Ausgang PB0 als Alarm auf 1 gesetzt.

**Wichtig:** Quelltext muss formatiert und sinnvoll kommentiert werden, sonst gibt es Punktabzug!

## 1. Timer-ISR — 25 Punkte




Eine PAP-Skizze und ein wenig Assembler sind gegeben, verwenden Sie zunächst diese Vorgaben!

```
.def Zeit1 = R16
.def Zeit2 = R17
.def tmp   = R18
.def itmp  = R19
.equ VORST = 256-100
```

**a**) Übersetzen Sie den PAP in kommentierten Assembler. **10P**

Zwischen Overflow-Ereignis und Aufruf der ISR vergehen 6 Takte.

**b**) Ermitteln Sie die tatsächlichen Zeiten zwischen den ISR-Aufrufen für die Fälle **Zeit1 <= 99** und **Zeit1 > 99**. **5P**

Der ISR-Entwurf funktioniert nicht optimal und stört das Hauptprogramm.

**c**) Welche Designfehler wurden gemacht? **4P**

**d**) Erstellen Sie eine bessere Assemblerlösung für die Timer0_ISR. **4P**

**e)** Erstellen Sie die Codesequenz für die Initalisierung des Timers **2P**

---

## 2. INT0_ISR — 10 Punkte

Der externe Interrupt INT0 soll bei fallender Flanke ausgelöst werden. Seine ISR setzt bei stehendem Timer0 den Timer auf den Wert VORST und die Register Zeit1, Zeit2 auf 0 und startet den Timer. Sonst stoppt sie den TImer und setzt PB0 auf 0 zurück.

**a)** Erstellen Sie Assembler Quellcode für INT0_ISR. **8P**

**b)** Erstellen Sie die Codesequenz für die Initalisierung des Interrupts. **2P**

---

## 3. Hauptprogramm — 10 Punkte

Im Hauptprogramm wird laufend überprüft, ob die 500 ms erreicht wurden und setzt in diesem Fall PB0 auf 1.

Erstellen Sie den kommentierten Assembler-Quelltext für das Hauptprogramm incl. der ISR-Tabelle und den notwendigen weiteren Initalisierungen (auf die diesbezüglichen Codesequenzen aus Aufgabe 1&2 darf als Kommentar verwiesen werden).

---

## 4. Weitere Aufgaben — 15 Punkte

**a**) Ermitteln Sie das 1er und 2er-Komplement zu 0x0000 und 0xaffe **4P**

**b)** Erstellen Sie den kommentierten Quellcode für ein Unterprogramm **w200**, dass incl. Aufruf und Rücksprung 200 Systemtakte Zeit verbraucht. Lösungen mit zu vielen NOP werden disqualifiziert! **6P**

**c)** Beschreiben Sie den Befehlszyklus bei einem AVR-$\mu$C und zeichnen Sie zur Erläuterung ein Blockschaltbild. **5P**

---

**Total Punkte 60**