

Musteraufgaben "Mikrocomputer-Technik"

1.) Welchen Wert haben die Flags N, Z und C des Statusregisters (CC-Register) nach Ausführung der folgenden beiden Befehle (0 oder 1 eintragen!)

CLRA
ORAA
ROLA
ROLA

##10000000

N (Negativ)	Z (Zero)	C (Carry)

2.) Was geschieht bei der Ausführung folgender Befehle und welche Register werden dabei ggf. verändert:

a.) **LDD** **0, Y**

b.) **BEQ** **FEHLER**

c.) **RTS**

d.) **CMPB** **#\$0F**

- 3) Entwerfen Sie ein **Unterprogramm** CHSUM, das die Checksumme (Prüfsumme z.B. zur Datensicherung) über einen fortlaufenden Speicherbereich bildet, und zwar ohne Berücksichtigung eines ggf. entstehenden Carry.

Als **Eingangsparameter** werden übergeben:

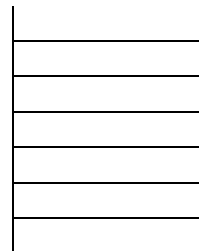
- Im Register **X** die Startadresse des Feldes
- Im Register **B** die Länge des Feldes

Das **Resultat** (die entstandene Checksumme) soll im Akku **A** an das aufrufende Programm übergeben werden, die anderen Register sollen wie vor dem Aufruf sein.

Kommentieren Sie das Programm! Auch ein **Flussdiagramm** wird bewertet!

- 4) Was bewirkt die Befehlsfolge (evtl. mit Skizze!)

PSHX
PULA
PULB



- 5) Ein Taster an Port C, Bit 0, soll über Software *entprellt* werden und zwar soll das Entprellen durch eine Warteschleife realisiert werden. In der Warteschleife wird 20 ms gewartet werden. Ein Programmablauf im zyklisch durchlaufenen Hauptprogramm könnte somit wie folgt aussehen.

- Wenn Taster gedrückt, springe ins Tastatur-Unterprogramm
- Aufruf Warteschleife (ca. 20 ms warten)
- Wenn die Taste immer noch gedrückt ist, wird im Akku **A** \$FF zurückgegeben, andernfalls 0.

- 6) Ein Mikrocontroller soll neben seinen anderen Aufgaben auch noch seine eigenen Betriebsstunden zählen. Dazu werden die entsprechenden Variablen beim Hochlauf (Initialisierung) auf Null gesetzt. Die gesamte Betriebszeit setzt sich aus den Variablen Tage, Stunden, Minuten und Sekunden zusammen. So steht sie jederzeit für beliebige Zwecke bereit (z. B. für ein Ausgabeprogramm).

Ein vorhandener Sekundentakt soll Interrupts auslösen. Die Aufgabe der entsprechenden Interrupt-Serviceroutine besteht in der Aktualisierung der Zeitvariablen.

- a.) Vereinbaren Sie die Variablen (es soll auch nach 10 Jahren noch kein Überlauf der Tage-Variablen auftreten), geben Sie im Kommentar den jeweiligen Wertebereich an, den die Variablen sinnvoller Weise haben können.
- b.) Schreiben Sie die Interrupt-Serviceroutine (eine wie auch immer geartete hardware-bezogene Quittierung des Interrupts ist nicht zu berücksichtigen).

7) Mit einem 68HC11-System soll eine Temperaturregelung realisiert werden. Am parallelen Port C ist ein Temperatursensor mit freilaufenden Analog-Digital-Wandler angeschlossen. Der Wandler liefert einen der Temperatur proportionalen 8-Bit-Wert. Der Temperaturbereich $[-20^{\circ}\text{C} \dots +65^{\circ}\text{C}]$ wird auf den Wertebereich $[0 \dots 255]$ abgebildet; die Auflösung beträgt also $1/3^{\circ}\text{C}$. In regelmäßigen Zeitabständen löst der A/D-Wandler - wenn ein aktueller Analogwert an **Port C** bereitsteht - an STRA über eine Low-High-Flanke einen Interrupt aus. Die Vereinbarungen von Adressen (Code, Daten, I/O-Ports usw.) entsprechen jenen des Praktikums. Sie können dieses als gegeben voraussetzen und müssen sie nicht im Code aufführen. Verwenden Sie aussagekräftige Namen!

An **Port B**, **Bit 0 – 2**, werden drei Ausgangsleitungen verwendet:

- **Bit 0** schaltet über ein geeignetes Interface die Heizung ein und aus (0=aus, 1=ein).
- **Bit 1** schaltet über ein geeignetes Interface die Kühlung ein und aus (0=aus, 1=ein).
- **Bit 2** gibt Alarm bei Leitungsbruch oder Leitungskurzschluss zum Analogsensor.

Am Eingang PA7 ist ein Not-Aus-Taster angeschlossen, der per **Polling** abgefragt wird (also nicht per Interrupt).

a) Schreiben Sie ein Unterprogramm "PCINIT" für die Initialisierung des Programms so, dass

- die Richtung der Datensignale an Port C als Eingang festgelegt wird,
- eine Tastenbetätigung am Eingang PA7 festgestellt werden kann und
- durch die positive Flanke an STRA ein bedingter Interrupt ausgelöst wird.

b) Programmieren Sie die Interruptbearbeitung für **STRA**: Einlesen des Temperaturwertes, Quittieren des Interrupts.

- Schreiben Sie die für die Festlegung des **Reset** und für die Bearbeitung eine bedingten Unterbrechung an **STRA** notwendigen **Adreßvereinbarungen** auf (→ **Setzen der Vektoren**).
- Schreiben Sie die Interrupt-Serviceroutine für **STRA**, welche den Temperaturwert von **Port C** einliest und in der Variablen **AWERT** speichert. Die Tatsache, dass ein neuer Wert eingelesen wurde, soll in der Variablen **AFLAG** mitgeteilt werden (**AFLAG** = 0: kein neuer Wert, **AFLAG** = \$FF: neuer Wert liegt vor).

c) Schreiben Sie ein Unterprogramm **NOTAUS** für den Not-Aus-Taster an PA7. Sobald dieser Taster gedrückt worden ist, sollen:

- die Bits **0** und **1** von **Port B** auf 0 gesetzt werden
- das Bit **2** von **Port B** blinken (abwechselnd 0 und 1 werden)

Beachten Sie, dass die Routine nicht mehr verlassen werden soll, sondern im Zustand "Blinken Bit 2" verbleiben. Für die Wartezeit zwischen den Wechseln von 0 auf 1 und umgekehrt darf auf ein vordefiniertes Unterprogramm **DELAY** zurückgegriffen werden (Sie müssen **DELAY** also nicht selbst programmieren).

d) Das Hauptprogramm läuft in einer Endlosschleife.

- Wurde die Not-Aus-Taste gedrückt, erfolgt ein Aufruf von **NOTAUS**.
- Ist die Variable **AFLAG** = \$FF, erfolgt die Temperaturregelung wie im Folgenden beschrieben.

Sobald ein neuer Wert vom Analog-Digital-Wandler ansteht, werden abhängig vom neuen Wert folgende Aktionen ausgeführt:

- o Temperatur $< +20^{\circ}\text{C}$: Heizung ein, Kühlung aus

- o Temperatur > +22°C: Heizung aus, Kühlung aus
- o Temperatur > +26°C: Heizung aus, Kühlung ein
- o Temperatur = -20°C: Heizung aus, Kühlung aus, Alarm ein
→ **Ansprung Not-Aus-Unterprogramm**
- o Temperatur = +65°C: Heizung aus, Kühlung aus, Alarm ein
→ **Ansprung Not-Aus-Unterprogramm**

Das Programm wartet, wie gesagt, auf **AFLAG** = \$FF und führt mit dem Wert aus **AWERT** den o. g. Regelalgorithmus durch. Danach wird **AFLAG** auf 0 gesetzt und auf den nächsten Wert gewartet.

Hinweise:

- o Ermitteln Sie zunächst die zu den o. g. Temperaturen gehörenden Eingangswerte.

Temperatur	Wert
-20°C	0 (Kurzschluss)
+20°C	
+22°C	
+26°C	
+65°C	255 (Kabelbruch)

- o Die Fehlerbehandlung (Werte 0 und 255) erfolgt günstigerweise zuerst.

Schreiben Sie nun das Hauptprogramm unter Verwendung der bisher angefertigten Unterprogramme.

8) Entwerfen Sie ein **Unterprogramm MEMCOPY**, das einen fortlaufenden Speicherbereich von einer Stelle im Speicher an eine andere Stelle kopiert. Als **Eingangsparameter** werden die folgenden Werte übergeben:

- o Im Register **X** die Startadresse des Quellbereichs
- o Im Register **Y** die Startadresse des Zielbereichs
- o Im Register **B** die Länge des Feldes

Um beispielsweise 100 Bytes von der Adresse \$8100 bis \$8163 in den Speicher von \$2000 bis \$2063 zu kopieren, wird im Hauptprogramm die folgende Befehlsfolge ausgeführt:

```
LDX  #$8100
LDY  #$2000
LDAB #100
BSR  MEMCOPY
```