

### **Aufgabe 33:**

#### **Kurzdarstellung der Aufgabe:**

In der Praktikumsaufgabe sollen Sie zeigen, daß Sie den Mikroprozessor MC6809 in Assembler programmieren können und dabei seine wesentlichen Komponenten (Anzeige, Tastatur, Parallelport, Timer und serielle Schnittstelle) ‚bedienen‘ können. Zwar handelt es sich um eine ‚synthetische‘ Aufgabe, jedoch werden die einzelnen Funktionen der Aufgabenlösung häufig für reale Anwendungen – so oder ähnlich – benötigt. Die Praktikumsaufgabe wird auf den folgenden Seiten in mehrere Teilaufgaben zerlegt, die Sie schrittweise zur Gesamtlösung führen werden.

*Im wesentlichen besteht die Aufgabe darin, den Timer #3 des MC6840 zu starten und danach jederzeit über die Tastatur oder den Parallelport Einfluß auf seine Funktionsweise nehmen zu können. Nach jeder Änderung sollen die Parameter des Timers über die V.24-Schnittstelle zu einem zweiten Rechner übertragen werden und dort den entsprechenden Timer in denselben Funktionsmodus versetzen.*

### **Teilaufgabe 33a):**

*In dieser Teilaufgabe sollen Sie zunächst den Timer #3 des MC6840 so programmieren und starten, daß er eine Rechteckschwingung minimaler Frequenz ausgibt. In einer Interruptroutine zum Timer #3 soll das Segment g der Anzeigestelle S7 zyklisch ein- und ausgeschaltet werden („blinken“). Weiterhin sollen in der Anzeige die Parameter des Timer #3, also sein Startwert und sein Steuerwort dargestellt werden.*

Im einzelnen soll das Programm

1. die Anzeige löschen,
2. den Timer #3 zur Ausgabe des Rechtecksignals minimaler Frequenz starten,
3. die benutzten Parameter des Timers #3 (Startwert und Steuerwort) in einem Pufferbereich P des Speichers ablegen,
4. in einem Unterprogramm die Zähler-Parameter in der Anzeige darstellen und zwar
  - in den Anzeigestellen S5 - S2 den Startwert,
  - in den Anzeigestellen S1 - S0 das Steuerwort,
5. in einer Endlosschleife („Hauptschleife“) verbleiben, in welcher der Pufferbereich P in der Anzeige – wie unter 4. beschrieben – dargestellt wird,
6. in der Interruptroutine zum Timer #3 das Segment g der Anzeigestelle S7 blinken lassen.

Hinweis: Vergessen Sie hier – wie auch in den folgenden Teilaufgaben – nicht das Löschen des Interrupt-Flags.

### **Teilaufgabe 33b):**

*In dieser Aufgabe soll das Programm so ergänzt werden, daß zu jeder Zeit die dargestellten und benutzten Timer-Parameter über die Tastatur geändert werden können.*

Im einzelnen werden die folgenden Programmergänzungen verlangt:

1. In der Hauptschleife soll die Tastatur zyklisch abgefragt werden und nach Drücken der
  - Taste L: („Latch“) in der Anzeigestelle S6 ein ‚L‘ dargestellt, ein neuer Startwert eingelesen, in den Anzeigestellen S5 - S2 ausgegeben und in die Latches des Timer #3 sowie den Puffer P eingetragen werden,
  - Taste S: („Steuerwort“) in der Anzeigestelle S6 ein ‚S‘ dargestellt, ein neues Steuerwort eingelesen, in den Anzeigestellen S1 - S0 ausgegeben und in das Steuerregister des Timer #3 sowie den Puffer P eingetragen werden.
2. Wird ein Startwert kleiner als \$0100 eingegeben, so soll er auf \$0100 erhöht werden.

### Teilaufgabe 33c):

*In dieser Aufgabe soll das bisher entwickelte Programm so ergänzt werden, daß die Timer-Parameter über den Port PB verändert werden können.*

Ändern Sie das Programm dazu so, daß

1. alle Portleitungen PB7,...,PB0 als Eingänge geschaltet sind,
2. daß über CB1 negative Flanken als Interrupts an den Prozessor weitergereicht werden,
3. daß die Leitung CB2 als statische Ausgangsleitung dient,
4. daß in der Hauptschleife die Tastatur auch auf Betätigungen der Taste ‚–‘ abgefragt wird und sich mit jeder Betätigung von ‚–‘ der Zustand ‚ein‘/‚aus‘ der LED an CB2 ändert,
5. daß in der Interruptroutine zu CB1 der Zustand der LED an CB2 getestet wird und der Zustand der Port-Eingänge PB7,...,PB0 beim LED-Zustand
  - ‚ein‘: als H-Byte des Startwerts von Timer #3 interpretiert und in seinen Latches
  - ‚aus‘: als Steuerwort von Timer #3 interpretiert und in seinem Steuerregister sowie an der entsprechenden Position des Speicherpuffers P abgelegt wird.Wird über die Portleitungen das H-Byte des Startwerts auf \$00 gesetzt, so soll der Startwert vom Programm auf den Wert \$0100 korrigiert werden.

### Teilaufgabe 33d):

*In dieser Aufgabe soll ein Unterprogramm ‚V24‘ entwickelt werden, daß den Inhalt des Speicherpuffers P über die serielle V.24-Schnittstelle des R6551 zu einem Empfänger überträgt. Die Übertragung soll im ASCII-Code geschehen, der in der folgenden Tabelle ausschnittsweise angegeben ist:*

**Ausschnitt aus dem ASCII-Code**

ASCII	Hex.	ASCII	Hex.	ASCII	Hex.	ASCII	Hex.
\$30	0	\$34	4	\$38	8	\$43	C
\$31	1	\$35	5	\$39	9	\$44	D
\$32	2	\$36	6	\$41	A	\$45	E
\$33	3	\$37	7	\$42	B	\$46	F

Im einzelnen soll das Unterprogramm folgende Bedingungen erfüllen:

1. Die V.24-Schnittstelle soll mit den folgenden Parametern initialisiert werden:  
*Paritätsprüfung deaktiviert, Sende- und Empfangs-Interrupt deaktiviert, Echo Mode disabled, DTR=1, Wortlänge 8 bit, 2 Stopbits, interner Sendetakt, Übertragungsrate 1800 Baud.*
2. Vor der Übertragung jedes Zeichens über die V.24-Schnittstelle soll das Statusregister des R6551 daraufhin überprüft werden, ob der Sender des R6551 zur Übertragung bereit ist.
3. Zu Beginn jeder Übertragung des Speicherpuffers P soll ASCII-Zeichen \$0D (CR - *Carriage Return*, ‚Wagenrücklauf‘) ausgesendet werden.
4. Das Unterprogramm ‚V24‘ soll vom Hauptprogramm immer dann aufgerufen werden, wenn der Zähler gestartet oder einer der Timer-Parameter geändert wurde.

### Teilaufgabe 33e):

*In dieser Aufgabe sollen Sie nun ein Programm schreiben, das auf einem zweiten Rechner (Empfänger,) läuft. Bei diesem zweiten Rechner kann es sich wahlweise um einen PC (mit 6809-Simulator) oder einen zweiten Praktikumsrechner handeln, der über die V.24-Schnittstelle angebunden ist. Das Programm soll die über die V.24-Schnittstelle empfangenen Parameter des Timers #3 in die Register des eigenen Timers eintragen. Außerdem sollen sie – wie im Sender – in der Anzeige des Empfängers dargestellt werden. Als Ergebnis sollen die Timer in beiden Rechner stets im selben Modus arbeiten.*

Das Programm soll – soweit wie möglich – aus Einzelteilen des Programms aus den Teilaufgabe 1 - 4 zusammengesetzt werden, wobei leichte Anpassung nötig sein können. Im einzelnen soll es:

1. die V.24-Schnittstelle wie in Teilaufgabe 4 initialisieren,
  2. den Timer #3 mit dem Startwert \$0000 und dem Steuerwort \$00 starten,
  3. beide Parameter in einem Speicherpuffer P' ablegen,
  4. den Startwert in den Anzeigestellen S5 - S2, das Steuerwort in den Stellen S1, S0 darstellen,
- danach in einer Hauptschleife
5. die V.24-Schnittstelle auf das Vorliegen eines empfangenen Zeichens solange abfragen, bis das Steuerzeichen \$0D empfangen wurde,
  6. die übertragenen Timer-Parameter empfangen, sie im Puffer P' ablegen und in der Anzeige – wie unter 4. angegeben – darstellen.

In der Interruptroutine zum Timer #3 sollen – wie in Aufgabe 1 – nur das Blinken des Segments g der Anzeigestelle S7 veranlaßt und natürlich die Interruptanforderung von Timer #3 gelöscht werden.

Für den Empfang der Parameter soll ein Unterprogramm ,V24E' geschrieben werden, das

1. jeweils zwei ASCII-Zeichen einliest,
2. diese in ihre entsprechenden Hexadezimalzeichen (0,...,9,A,...,F) zurückwandelt,
3. zu einem Byte (2 Zeichen) geeignet zusammenfaßt.