

MPS1-Praktikum - Versuch 3

Die Lösungen aller drei Aufgaben sind für das Testat mit einem in der Vorbereitung erstellten **PAP / Struktogramm** und einer **ausgedruckten Übersetzungsliste (LIST name.LST)** zu dokumentieren!!!

Zuhause vorbereitete Lösungen können entweder mit Diskette oder per **USB-Memory-Stick** (der sollte beim PC-Einschalten bereits eingesteckt sein) überspielt werden.

1. Aufgabe : Zeichenausgabe auf Bildschirm

Es ist ein Programm zu entwerfen, das Zeichen im ASCII-Code (*Skript Seite 74 u. 75*) von den Kippschaltern des Portes P4 einliest, auf den Bildschirm ausgibt und im externen Datenspeicher ablegt.

Der ASCII-Code des gewünschten Zeichens

- werde an den P4-Kippschaltern eingestellt,
- dann mit **P1.0 = 0 einmal eingelesen,**
- mit dem Unterprogramm SENDSE über die serielle Schnittstelle **einmal** zum Terminal gesendet und damit das Zeichen auf dem Bildschirm dargestellt,
- und zuletzt, beginnend ab **Adresse 1000 h** fortlaufend **einmal im externen Datenspeicher abgespeichert.**

Durch P1.3 = 0 wird unabhängig von P1.0 die ab **1000 h abgespeicherte Zeichenkette vollständig** auf den Bildschirm ausgegeben und danach zum Betriebssystem MONITOR 51 (8000 h) zurückgesprungen.

- Für das Erkennen des **Endes der Zeichenkette** muß entweder - eine Zählung durchgeführt ,
- eine Ende-Markierung gesetzt oder
 - das Low-Byte eines Adresszeigers herangezogen werden.

Hinweis: Sie können davon ausgehen, daß bedingt durch die manuelle Eingabe sowieso nicht mehr als 255 Zeichen eingegeben werden -> Beschränkung auf Byte

Ein korrekter Abbruch muß auch dann möglich sein, wenn kein Zeichen eingegeben wurde !!

Für den Test des Programmes beginnen Sie die Eingabe der Zeichenkette mit den Steuerzeichen **Zeilenvorschub (Line Feed LF)** und **Wagenrücklauf (Carriage Return CR)** und erzeugen dann auf einer neuen Zeile den Text:

MPS1-Labor 80535

Hinweis: Das unten angegebene Sendeprogramm für die serielle Schnittstelle SENDSE verlangt beim Aufruf den ASCII-Code des auf den Bildschirm auszugebenden Zeichens im Akku.

Erst nach **abgeschlossenem seriellem Sendevorgang** wird aus dem Unterprogramm zurückgesprungen.

Die Initialisierung der serielle Schnittstelle wird für diesen Versuch vom Betriebssystem MONITOR 51 übernommen (8 Datenbit, 1 Stoppbit, kein Paritätsbit, 9600Bd) *Skript Seite 66 u. 67*

Fügen Sie dieses Unterprogramm hinter Ihr eigenes Programm, aber noch vor die Assembleranweisung END.

```
SENDSE:  MOV  SBUF,A    ;startet Sendevorgang
          CLR  TI        ;löscht Sendebit
WARTE:   JNB  TI,WARTE ;Senden nicht beendet ?
          RET
```

Beispiel: Ausgabe eines Leerzeichens

```
-
-
MOV  A,#20h    ; ASCII-Code von SP
LCALL SENDSE
-
-
```

2. Aufgabe: BCD - Flankenzähler

Es ist ein Programm zu entwerfen, das jede Änderung der Kippschalterstellung von P1.0, d.h. **jede steigende und fallende Signalflanke, in einem**

vierstelligen BCD-Zähler (0000 – 9999)

und in einem

16-Bit-Dualzähler (00000 – 65535 / FFFFh) zählt.

Der Stand des

16-Bit-Dualzählers wird binär im Dualcode auf den LEDs von **Port C (High-Byte) und Port5 (Low-Byte)**

und der

BCD-Zählerstand dezimal auf den **Bildschirm** ausgegeben. *Skript Seite 46 u. 47*

Jeder neue BCD-Zählwert werde durch ein Leerzeichen, gefolgt von den vier Dezimalziffern, auf dem Bildschirm dargestellt.

Das Programm arbeite in Endlosschleife, soll also nur durch RESET abgebrochen werden können.

Hinweis: Für die Ausgabe auf den Bildschirm müssen die vier BCD- Ziffern nacheinander in den ASCII-Code gewandelt und mit dem seriellen Sendeprogramm SENDSE auf den Bildschirm dargestellt werden (Übergabe im Akku)

Entwerfen Sie einen **inkrementalen BCD-Zähler**, denn die alternative Lösung einer 16 Bit Dual-BCD-Wandlung ist schwieriger zu programmieren (16 Bit- Division !!, aber Sie dürfen natürlich , wenn Sie sich da heranwagen wollen).

Verwenden Sie für die **Dezimalkorrektur** (Überspringen der Pseudotetraden) im BCD-Zähler den **DA A – Befehl !!**

Die positive und negative Flankendetektion kann entweder mittels **Polling** durch **Pegelabfrage von P1.0** oder durch Abfrage des positiv/negativ-flankengetriggerten **Interrupt-Anzeige-Flags IEX3** erfolgen. Eine Lösung mittels **Interrupt bzw Interrupt-Service-Routine** ist zwar durch die Aufgabenstellung nicht erforderlich, darf aber falls unbedingt gewünscht und in der Vorlesung bereits behandelt, auch realisiert werden.

Testen: Um die beiden Zähler über den gesamten Wertebereich testen zu können, wird ein Impulsgenerator benötigt. Durch das Voranstellen der folgenden Programmzeilen an den Anfang Ihres Programmes wird mit dem Timer 2 im Compare-Modus 0 ein 50 Hz Rechtecksignal mit Puls/Pause 1:1 am Portpin P1.3 erzeugt. Durch Verbinden der Buchsen P1.3 mit P1.0 (beide Kippschalter geöffnet !!) kann dann der Zähler getestet werden.

```

ORG    ....
MOV   CCL3,#0F0H
MOV   CCH3,#0D8H
MOV   CRCL,#0E0H
MOV   CRCH,#0B1H
MOV   CCEN,#80H
MOV   T2CON,#11H
- - - ;eigenes Programm
  
```

3. Aufgabe: 16-Bit Arithmetik mit vorzeichenbehafteten Dualzahlen *Skript Seite 47*

Es ist ein Programm zu entwerfen, das zwei **vorzeichenbehaftete 16-Bit Dualzahlen addiert oder subtrahiert**.

Dazu werden die 16 Bit-Operanden mit den Kippschaltern auf der Anzeigebox wie folgt eingestellt:

	HighByte	LowByte		
OP1	Port A	Port 1	(Eingabe)	
+/- OP2	Port B	Port 4	(Eingabe)	
ERG	Port C	Port 5	(Ausgabe)	

Negative Operanden müssen im 2er-Komplement eingestellt werden !!

Bei **P3.2 = 1** soll eine **Addition** und bei **P3.2 = 0** eine **Subtraktion (OP1 - OP2)** der beiden 16-Bit-Operanden durchgeführt werden.

Mit **P3.3 = 0** soll das Programm abgebrochen und in das Betriebssystem **MONITOR 51** zurückgekehrt werden.

Das Carry-Bit CY des Ergebnisses werde an P3.4 und das Overflow-Bit OV an P3.5 angezeigt.

Skript Seite 45 und 37

Das **Überschreiten des Zahlenbereiches** soll auf dem Bildschirm (alternativ LCD-Display) in konstanter Position (Endlosschleife!) durch Ausgabe eines **Leerzeichens (kein Überschreiten)** oder beim **Überlauf durch +** bzw beim **Unterlauf durch -** angezeigt werden.

Beachten Sie, daß im vorzeichenbehafteten Dualsystem bei beiden Operationen ein Über- oder Unterlauf auftreten kann. (Unterscheidung Über-/Unterlauf -> Skript !! oder Herleitung aus Zahlenkreis)

Vervollständigen Sie die Rechnungen in der **Tabelle (in der Vorbereitung !!!)** und testen Sie damit Ihr Programm :

Op1	Op2	Erg	Op1	Op2	Erg	P3.5 OV	P3.4 CY	Über/Unterlauf +/-
+32767 (+)	+255 =	+33022	7FFF (+)	00FF =	80FE	1	0	+
+32767 (+)	-255 =				=			
+32767 (-)	+255 =				=			
+32767 (-)	-255 =				=			
-32767 (+)	+255 =				=			
-32767 (+)	-255 =				=			
-32767 (-)	+255 =				=			
-32767 (-)	-255 =				=			

Hinweis: Beachten Sie die unterschiedlichen Funktionen der **Arithmetik-Flags CY und OV** für das Signalisieren einer Zahlenbereichsüberschreitung und der Realisierung des internen Ergebnis-Übertrags vom Low- zum High-Byte. *Skript Seite 47*