Versuchsbeschreibung Nr. 9 und Nr. 10

Sie erhalten die Versuchsbeschreibung Nr.9 und Nr. 10 in einer Ausgabe. Die Versuche bestehen aus insgesamt 2 Aufgaben. Da die 2. Aufgabe recht anspruchsvoll ist, können Sie am Versuchstag des 9. Versuches damit beginnen, und diese am 10. Versuchstag beenden.

Verwendete Geräte: 1 PC 1 PIC-Entwicklungssystem

Literatur:
1. Assembler-Handbuch-Auszüge
2. PIC-Datenbuch-Auszüge

Inhalt: Hardwareanbindung an den PIC17C43

Anmerkung:

Für die einzelnen Aufgaben stehen Testkästen zur Verfügung. Diese Testkästen sind sehr einfach aufgebaut, deshalb werden Sie nicht näher erläutert. Auf jedem Testkasten befindet sich ein Sockel, an dem das Kabel der Emulator-Probe angeschlossen wird. Beim Anschluss des Kabels ist darauf zu achten, daß das Pin 1 des Kabels (gekennzeichnet durch einen runden Punkt) auch an Pin 1 beim Sockel angeschlossen wird.

Auf jedem Testkasten befinden sich die Ein- und Ausgabebauteile, die schon an gewissen Ports des PIC17C43 angeschlossen sind.

Falls sich auf dem Board Buchsen für eine Versorgungsspannung befinden, schliessen Sie bitte das Board an die entsprechende Spannung an. **ACHTUNG !!!!!** Achten Sie beim Anschluß der Versorgungsspannung auf die Reihenfolge des Einschaltens der Geräte, wie Sie im Anhang von Versuch 8 angegeben wurde.

Aufgaben:

1.1. Schreiben Sie in Programm, das an Port C 8 Schalter einliest, und diese auf 8 LEDs am Augang (PORT D) abbildet.
Die Schalter sollen solange eingelesen werden, bis alle Schalter auf Low = 0 gestellt sind.

1.2. Schreiben Sie ein Programm, das einen Leuchtpunkt auf den 8 LEDs an Port D frei durchlaufen lässt.
Mit den 8 Schaltern an Port C soll die Durchlauffrequenz des Leuchtpunktes verändert werden können.
Die Schalter sollen solange eingelesen werden, bis alle Schalter auf Low = 0 gestellt sind.

- Speichern Sie ihre Assembler-Files unter **c:\labor\vers9** als **led1.asm** und **led2.asm** ab.
- Dokumentieren Sie Ihr Assemblerprogramm ausreichend !
- Steppen Sie jeweils Ihre Programme mit dem Emulator durch
- Kontrollieren Sie das Ergebnis.

Als Eingabe-Möglichkeit befindet sich auf dem Testboard ein DIP-Schalter. Wird ein Schalter auf "ON" gestellt, so wird eine "0" an den entsprechenden Port gelegt. Um eine LED leuchten zu lassen, muß an den entsprechenden Port eine "1" gelegt werden.

Die Schalter sind an den PORT C und die LEDs sind von links nach rechts gesehen an den PORT D wie folgt angeschlossen::

Schalter 1	Port C0	LED 1	Port D0
Schalter 2	Port C1	LED 2	Port D1
Schalter 3	Port C2	LED 3	Port D2
Schalter 4	Port C3	LED 4	Port D3
Schalter 5	Port C4	LED 5	Port D4
Schalter 6	Port C5	LED 6	Port D5
Schalter 7	Port C6	LED 7	Port D6
Schalter 8	Port C7	LED 8	Port D7

2.1. Mit einem Taster soll eine 7-Segmentanzeige (7-SA)an Port D jeweils um eine Ziffer hochgezählt werden.
Der Taster soll sich am Interrupt-Eingang INT befinden (T3).
Lassen Sie die 7-SA in einem definierten Zustand zu zählen beginnen.

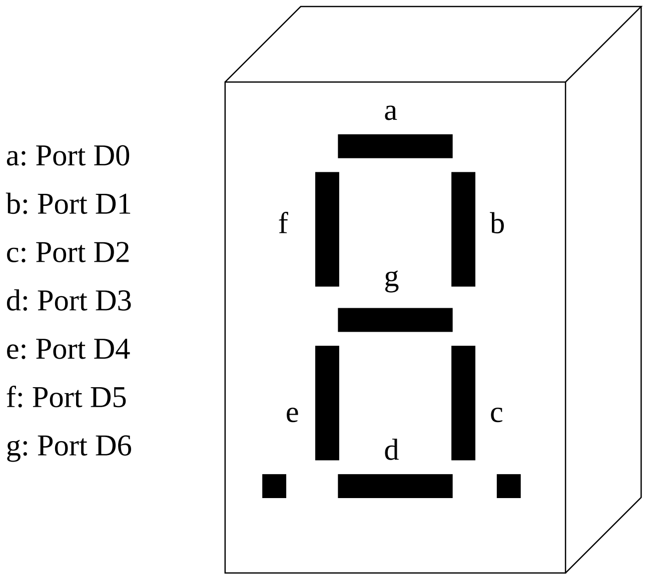
2.2. Es stehen für diesen Versuchsteil, folgende Dinge zur Verfügung:
- 7-Segmentanzeige zur Ausgabe an Port D
- 3 Taster zur Eingabe (T1,T2,T3)

T1 soll das Hochzählen der 7-Segmentanzeige in einem definierten Zustand starten (Port B0 - Eingang). Mit T2 soll die 7-Segmentanzeige angehalten werden können (RT-Eingang). T3 soll die 7-Segmentanzeige zu jedem beliebigen Zeitpunkt auf 0 zurücksetzen (INT-Eingang).

- Speichern Sie ihre Assembler-Files unter **c:\labor\vers9** als **ssa1d.asm** und **ssa2.asm** ab.
- Dokumentieren Sie Ihr Assemblerprogramm ausreichend !
- Steppen Sie jeweils Ihre Programme mit dem Emulator durch.
- Kontrollieren Sie das Ergebnis.

Als Eingabe-Möglichkeit befinden sich auf dem Testboard die Taster T1 - T3. Beim Drücken der Taste liegt ein Low-Signal "0" an dem entsprechenden Eingang, ansonsten ein High-Pegel "1".

Als Ausgabe dient die 7-Segmentanzeige. Um ein Segment aufleuchten zu lassen, muß eine "0" am entsprechenden Port ausgegeben werden. Die Siebensegmentanzeige ist wie unten abgebildet am Port D angeschlossen.



Anmerkungen:

- Die Tasten sind nicht entprellt, d.h. beim Betätigen der Taster entsteht eine ganze Reihe von Impulsen. Nehmen Sie deshalb in Ihren Assemblerprogrammen eine softwaremässige Entprellung vor.