

# Praktikumsanleitung Maschinennahe Programmierung

Verfasser: R. Hager, Ergänzungen von T. Tempelmeier

## Übungsaufgaben:

- UEB0 Einführung in das CodeWarrior-Entwicklungssystem  
Bauen Sie in das vorgegebene Projekt für das Unterprogramm `uart_putchar` einen Aufruf in Assembler ein.

Schreiben Sie jeweils ein Assembler-Programm, welches die folgenden Aufgaben löst.

- UEB1 Ausgabe eines Strings  
Mit einer Schleife alle Zeichen des Strings einzeln ausgeben
- UEB2 Ausgabe eines Strings (rückwärts)  
Wie Übung 1, aber String rückwärts (beginnend mit dem letzten Zeichen) ausgeben.
- UEB2a Unterschied Zeiger auf String / String als Array in C  
Geben Sie zwei Strings aus (unterschiedlich global in C definiert `char*` oder `char []`).  
Analysieren Sie den Code, der sich bei einer Programmierung in C ergibt.
- UEB3 Ausgeben eines Speicherwortes als Hexadezimalwert  
Ein vorgegebenes Wort im Speicher als ASCII-String ausgeben
- UEB4 Ausgeben eines Speicherwortes als Hexadezimalwert als Unterprogramm  
Umgestaltung von UEB3 zu einem Unterprogramm gemäß Aufrufkonvention unseres Compilers und Aufruf des Unterprogramms aus Assembler
- UEB5 `mymemcopy`  
Erstellung und Test eines Unterprogramms `mymemcopy` (analog zu `memcpy`) und Aufruf des Unterprogramms aus C.  
Schreiben Sie den Funktionskopf in C, erzeugen Sie jedoch den Eingangs-, Ausgangs-, Frame- und Return-Code selbst in Assembler!
- Wh Rückschau auf Übung 2a: Wie sieht der vom C-Compiler erzeugte Code bei lokal definierten Strings aus?
- UEB6 Illegal Instruction  
Testen Sie diesen "Befehl" (einschl. Verbiegen des zugehörigen Vektors)
- UEB7 Trap-Befehl  
Simulieren Sie einen "Betriebssystemaufruf" mit dem Trap-Befehl;  
verwenden Sie eine einfache Stringausgabe als Ersatz für die Betriebssystemfunktion; übergeben Sie die Stringadresse als Parameter an das "Betriebssystem"
- UEB7a Schreiben Sie ein Unterprogramm (keine Stackframes, Parameterübergabe nicht über den Stack, sondern über Register), das den Befehl `BITREV` nachbildet.
- UEB8 Aufbau und Ausgabe einer einfach verketteten, sortierten Liste von Integerzahlen  
Verwenden Sie die Vorgabe `UEB8_Vorgabe.c`

## Übungen zu den Schnittstellen der eingesetzten Controller

- UEB9      Ansteuerung der Leuchtdioden bei Tastendruck

Programmieren Sie den Mikrocontroller so, dass bei Betätigung der Tasten SW1 und/oder SW2 eine Anzeige an den Leuchtdioden erfolgt.

Hilfsmittel (werden in der Vorlesung besprochen):

- M52259DEMO "Schematics" (Prinzipschaltbilder)
- MCF52259RM [MicroRefMan], Kapitel 15
- Project\_Template\_UEB9

Realisieren Sie folgende Varianten:

- a) LED leuchtet, solange der entsprechende Schalter betätigt ist.
- b) LED wechselt (ein-aus-ein- usw.) mit jeder Schalterbetätigung
- c) Die vier LEDs zählen binär mit jeder Schalterbetätigung von SW1
- d) SW1 schaltet eine LED ein, SW2 schaltet sie wieder aus,  
wobei auf SW1 und SW2 per Interrupt reagiert werden soll.

- UEB10      Ausgabe auf die serielle Schnittstelle ohne Verwendung der vorhandenen Routinen (also Ansprechen der nackten Hardware)
- UEB11      Softwaregesteuerte Erzeugung eines "Echos" zu einem externen Interruptsignal; Test, ab wann der Mikrocontroller mit den Interrupts nicht mehr Schritt halten kann.
- UEB12      Steuerung einer Tür unseres Liftmodells (optional, noch nicht sicher)