

# Lösungshinweise zu Übungs- und Prüfungsaufgaben

(Kurzfassung, ohne Gewähr)

Bitte Email an Werner.Zimmermann@hs-esslingen.de, wenn Sie einen Fehler finden!

## Aufgabe Lauflicht

Siehe CodeWarrior-Projekte Lauflicht.mcp, LauflichtAsm.mcp

## Aufgabe Addieren1

Siehe CodeWarrior-Projekt Addieren1.mcp

### 2./3. Globale Variable verschiedener Größe

<pre> char a, b, c;  char main(void) 28: { 29:     EnableInterrupts; 0000 10ef          CLI 30: 31:     a = 10; 0002 c60a          LDAB #10 0004 7b0000        STAB a  32:     b = 20; 0007 58            ASLB 0008 7b0000        STAB b  33:     c = a+b; 000b f60000        LDAB a 000e fb0000        ADDB b 0011 7b0000        STAB c  34:     return c;  35: } 0014 3d            RTS </pre>	<pre> int a, b, c; 26: 27: int main(void) 28: { 29:     EnableInterrupts; 0000 10ef          CLI 30: 31:     a = 10; 0002 c60a          LDAB #10 0004 87            CLRA 0005 7c0000        STD a  32:     b = 20; 0008 58            ASLB 0009 7c0000        STD b  33:     c = a+b; 000c fc0000        LDD a 000f f30000        ADDD b 0012 7c0000        STD c  34:     return c;  35: } 0015 3d            RTS </pre>	<pre> 25: long a, b, c; 26: 27: long main(void) 28: { 29:     EnableInterrupts; 0000 10ef          CLI 30: 31:     a = 10; 0002 c60a          LDAB #10 0004 87            CLRA 0005 7c0000        STD a:2 0008 c7            CLRB 0009 7c0000        STD a  32:     b = 20; 000c c614          LDAB #20 000e 7c0000        STD b:2 0011 c7            CLRB 0012 7c0000        STD b  33:     c = a+b; 0015 fc0000        LDD a:2 0018 f30000        ADDD b:2 001b 7c0000        STD c:2 001e fc0000        LDD a 0021 f90000        ADCB b:1 0024 b90000        ADCA b 0027 7c0000        STD c  34:     return c; 002a fc0000        LDD c:2 002d fe0000        LDX c 35: } 0030 3d            RTS </pre>
--	---	--

### 4. Lokale Variable auf dem Stack

<pre> char a, b, c;  41: TYPE main(void) 42: { TYPE a, b, c; 0000 1b9d          LEAS -3,SP 43: 44:     EnableInterrupts; 0002 10ef          CLI 45: 46:     a = 10; 0004 c60a          LDAB #10 0006 6b82          STAB 2,SP 47:     b = 20; 0008 58            ASLB 0009 6b80          STAB 0,SP 48:     c = a+b; 000b 860a          LDAA #10 000d ab80          ADDA 0,SP 000f 6a81          STAA 1,SP 49:     return c; 0011 b701          TFR A,B 50: } 0013 1b83          LEAS 3,SP 0015 3d            RTS </pre>
--

5. -

**Aufgabe Addieren2**

Siehe CodeWarrior-Projekt Addieren2.mcp

**Aufgabe Blinkende Leuchtdioden mit RTI-Interrupt**

Siehe CodeWarrior-Projekte RtiInterrupt.mcp, RtiInterruptAsm.mcp

Minimale Interruptfrequenz 3,8Hz → Periodendauer 262ms:  $x=7, y=15 \rightarrow RTICTL=\$7F$

Maximale Interruptfrequenz 3,9kHz → Periodendauer 256µs:  $x=1, y=0 \rightarrow RTICTL=\$10$

**Aufgabe Blinkende Leuchtdioden mit dem Modulus-Down-Counter-Interrupt**

Siehe CodeWarrior-Projekte MdcInterrupt.mcp, MdcInterruptAsm.mcp

Minimale Interruptfrequenz  $24\text{MHz} / (16 \cdot 65536) = 22,9\text{Hz} \rightarrow \text{Periodendauer } 43,7\text{ms}$

Maximale Interruptfrequenz  $24\text{MHz} / (1 \cdot 65536) = 366\text{Hz} \rightarrow \text{Periodendauer } 2,7\text{ms}$

**Aufgabe Ansteuerung des 7-Segment-Displays**

Siehe CodeWarrior-Projekt SevenSegmentDisplay.mcp

**Aufgabe Musik1**

Siehe CodeWarrior-Projekt Musik1.mcp

**Aufgabe Musik2**

Siehe CodeWarrior-Projekt Musik1.mcp

**Aufgabe PWM mit Tastensteuerung**

Siehe CodeWarrior-Projekt PWM1.mcp

**Aufgabe Signalerzeugung für Periodendauermessung**

Siehe CodeWarrior-Projekt Timer3.mcp

**Aufgabe StrCpy****Aufgabe MemCpy****Aufgabe Buffer**

Siehe CodeWarrior-Projekt AssemblerUnterprogramme.mcp