# Praktikum Mikrocomputertechnik (EIT - B3, MEC - B5)

# Versuch 04 –

# Assembler: Digitaler Würfel, Lauflicht 2.0

Version: 2024-10-31a

# Inhaltsverzeichnis

1	Auf	fgabenstellung	2
2		fgabe 01: Digitaler Würfel	
	2.1	Aufgabenstellung	
	2.2	Erzeugen eines Assembler Projekt in CCS	3
	2.3	Informationssammlung	4
	2.4	Ablaufdiagramm	4
	2.5	Programmierung	5
3	Auf	fgabe 02: Lauflicht mit Drehrichtungsumkehr	6
	3.1	Aufgabenstellung	6
	3.2	Informationssammlung	6
	3.2	.1 Das Kommando RLC	e
	3.3	Ablaufdiagramm	7
	3.4	ASM Programm Musterlösung	7
	3.5	Probleme durch Kontaktprellen	8

## 1 Aufgabenstellung

In diesem Praktikum werden wir uns noch einmal mit der Programmierung in Assembler beschäftigen. Es sind zwei Aufgaben zu erledigen

- 1. Erzeugen eines digitalen Würfels, der Werte von 0 bis 4 ausgeben kann
- Erweiterung des Lauflichts aus Versuch 03:
   Die Laufrichtung der LEDs soll abhängig vom Zustand des Tasters S1 werden

### 2 Aufgabe 01: Digitaler Würfel

#### 2.1 Aufgabenstellung

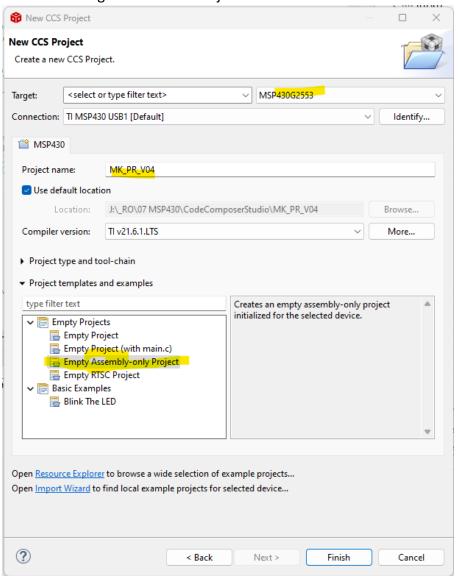
Wir wollen einen digitalen Würfel erzeugen.

Solange der Taster S2 gedrückt ist soll ein Zähler kontinuierlich von 0 bis 4 hochzählen. Wenn die Taste nicht mehr gedrückt wird soll der Zählerstand an den LEDs ausgeben werden.

Bei Zählerstand 0 keine LED, bei Zählerstand 1 eine LED, bei Zählerstand 2 zwei LEDs usw. Verwenden Sie R5 zum Aufbewahren des Zählerstands.

#### 2.2 Erzeugen eines Assembler Projekt in CCS

Über das Dialogfeld New CCS Project ...



.. erzeugen wir uns ein vorkonfiguriertes Projekt.

Benennen Sie die Datei main.asm in main\_04\_01.asm um.

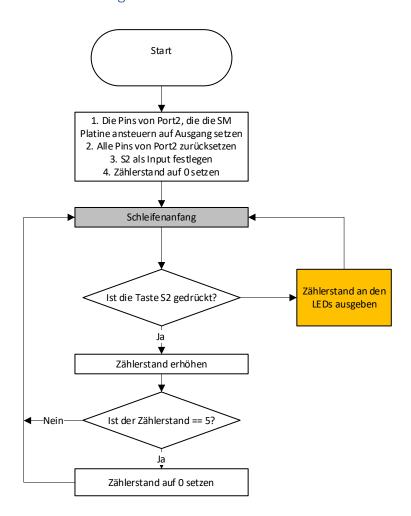
Fügen Sie den Quellcode zum Erzeugen eines Lauflichts aus dem letzten Praktikum ein.

#### 2.3 Informationssammlung

Verwenden sie zum Vergleich von zwei Zahlen das Assemblerkommando CMP.

Suchen Sie das Kommando in slau144k und machen Sie sich mit dem Kommando vertraut.

#### 2.4 Ablaufdiagramm



Erstellen Sie ein ähnliches Ablaufdiagramm für die Aufgabe "Zählerstand an den LEDs ausgeben".

#### 2.5 Programmierung

```
2; Aufgabe: Digitaler Würfel liefert Werte von 0 bis 4 - Ausgabe an die LEDs der SM-Treiberplatinen
 3;
 4; Created on: 28.10.2024
 5; Author: kol
 7; Status:
      241028a: Erste Schritte
 9; 241028b: Schaut gut aus
10;
11;----
       .cdecls C,LIST,"msp430.h" ; Include device header file
12
13
14:----
             .def RESET
                                              ; Export program entry-point to
15
16 ; make it known to linker.
17;------
18 .text
19 .retain
                                             ; Assemble into program memory.
                                               ; Override ELF conditional linking
20
                                              ; and retain current section.
             .retainrefs
21
                                               ; And retain any sections that have
22
                                              ; references to current section.
25; Definitionen
26;-----
25;
27 LED_A .equ BIT0
28 LED_B .equ BIT1
29 LED_C .equ BIT2
30 LED_D .equ BIT3
31 S2 .equ BIT3
32 LED_A_B .equ LED_A + LED_B
33 LED_A_B_C .equ LED_A + LED_B + LED_C
34 ALL_LEDs .equ LED_A + LED_B + LED_C
35
38 RESET mov.w #_STACK_END,SP ; Initialize stackpointer
39 StopWDT mov.w #WDTPW|WDTHOLD,&WDTCTL ; Stop watchdog timer
40
41
```

## 3 Aufgabe 02: Lauflicht mit Drehrichtungsumkehr

#### 3.1 Aufgabenstellung

Wir wollen das Programm zur Erzeugung des Lauflichts aus dem vorherigen Praktikum erweitern.

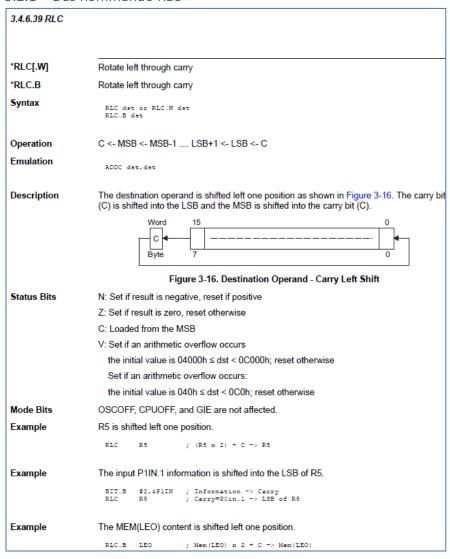
Solange die Taste S2 gedrückt ist, soll das Lauflicht in die andere Richtung laufen.

Verwenden Sie das Register R5 für die Wartezeit, und das Register R6 als Zwischenspeicher für die Ausgabe an Port P2.

Verwenden Sie das Kommando RLC um die Bits im Register R6 nach links zu shiften.

#### 3.2 Informationssammlung

#### 3.2.1 Das Kommando RLC



#### 3.3 Ablaufdiagramm

Erweitern Sie das Ablaufdiagramm aus Versuch 03 - Aufgabe 03: Wenn der Taster S2 gedrückt ist, soll sich das Lauflicht in die entgegengesetzte Richtung bewegen.

#### 3.4 Programmierung

```
\ main_04_01.asm \times \ main_03_02.asm \ mains_04_02.asm \ main_03_03.asm
1;-----
 2; Aufgabe: Lauflicht an P2.0 bis P2.3 mit Richtungsumkehr
 3:
 4; Created on: 24.10.2024
5; Author: kol
 6:
 7; Status:
 8; 241024a: Erste Schritte
    241024b: Die LED Daten werden in R6 gespeichert, Links- und Rechtslauf funktionieren
 9;
10; 241031a: Das Programm kann als Musterlösung verwendet werden
11;
12 ;-----
          .cdecls C,LIST,"msp430.h" ; Include device header file
13
15 ;-----
          .def RESET ; Export program entry-point to
16
17 ; make it known to linker.
18;-----
     .text
.retain
                                  ; Assemble into program memory.
; Override ELF conditional linking
21
                                  ; and retain current section.
         .retainrefs
                                  ; And retain any sections that have
22
                                  ; references to current section.
23
25 ;-----
26; Definitionen
27 ;-----
28 LED_A .equ BIT0
29 LED_B .equ BIT1
30 LED_C .equ BIT2
31 LED_D .equ BIT3
32 S2 .equ BIT3
33
34:-----
```

#### 3.5 Probleme durch Kontaktprellen

Es passiert häufig, daß sich trotz korrektem Programm die Anzahl der leuchtenden LEDs sowohl beim Drücken als auch beim Loslassen des Tasters ändern.

Dies könnte mit dem Kontaktprellen des Tasters zusammenhängen:

