

## Aufgabe

### Ansteuerung des 7-Segment-Displays auf dem Dragon12-Entwicklungsboard

(CodeWarrior-Projekt `SevenSegmentDisplay.mcp`)

Auf dem Dragon12-Board befindet sich ein vierstelliges 7-Segment-Display. Die vier Stellen werden im Multiplex-Betrieb angesteuert, wobei Port B die gemeinsamen Datenleitung für die 8 Segmente (Bit 7 ist der Dezimalpunkt, Bit 0 entspricht Segment 0 usw.) aller Stellen bildet.

Die Auswahl, welche Stelle angesteuert wird, erfolgt über Port P.3...0, wobei logisch 0 eine Stelle aktiviert, logisch 1 deaktiviert. Damit der Betrachter den Eindruck hat, dass alle Stellen gleichzeitig sichtbar sind, ohne dass die Anzeige flackert, sollte zwischen den vier Stellen mit einer Multiplex-Frequenz von > 50Hz umgeschaltet werden.

Für die Ansteuerung der 7-Segment-Anzeige wurden die unten dargestellten, leider nicht kommentierten Funktionen `init7Seg` und `displayRaw7Seg` entwickelt, die für das Multiplexen den Modulus-Down-Counter (MDC) verwenden, der bereits aus der Aufgabe „Modulus-Down-Counter“ bekannt ist.

- Kommentieren Sie die Befehle der Funktion `init7Seg`. Mit welcher Frequenz wird die Modulus-Down-Counter-ISR aufgerufen? Was ist am Ende dieser Funktion auf der 7-Segment-Anzeige zu sehen?
- Wieviele Stellen der vierstelligen 7-Segment-Anzeige werden bei jedem Aufruf der Funktion `mdcISR` angesteuert? Wozu dient die Variable `nSeg`?
- Welche Anzeige ist am Ende des folgenden Programmfragmentes im 7-Segment-Display zu sehen?

```
.const:SECTION                                ; ROM: Constant data
myText: DC.B $76, $79, $38, $73

.init: SECTION                                ; ROM: Code section
main:
    . . .
    JSR  init7Seg                             ; Initialize the display
    LDD  #myText                             ; Display data
    JSR  displayRaw7Seg
    . . .
```

---

Ansteuerfunktionen für das 7-Segment-Display:

```
;-----
.data: SECTION
pData: DS.W 1
nSeg:  DS.B 1

;-----
.vect: SECTION
      ORG $FFCA
int26:  DC.W mdcIsr

;-----
init7Seg:
    PSHD

    BSET DDRP, #$0F
    BSET PTP,  #$0F
```

```

        BSET DDRB, #$FF
        MOVB #0, PORTB

        MOVB #0, nSeg
        BSET MCCTL, #$C4

        PULD
        RTS

;-----
displayRaw7Seg:
        STD  pData
        RTS

;-----
mdcIsr: BSET PTP, #$0F
        LDAB nSeg
        LDX  pData
        MOVB B, X, PORTB

        CMPB #0
        BNE  mcd1
        BCLR PTP, #$01
        BRA  mcd

mcd1:   CMPB #1
        BNE  mcd2
        BCLR PTP, #$02
        BRA  mcd

mcd2:   CMPB #2
        BNE  mcd3
        BCLR PTP, #$04
        BRA  mcd

mcd3:   BCLR PTP, #$08

mcd:    LDAB nSeg
        INCB
        ANDB #$03
        STAB nSeg

        BSET MCFLG, #$80
        NOP
        RTI

```

Zuordnung der Segmente bei der  
7-Segment-Anzeige:

