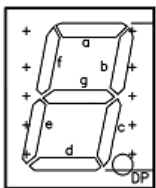


## Arbeiten mit Tabellen im Codespeicher

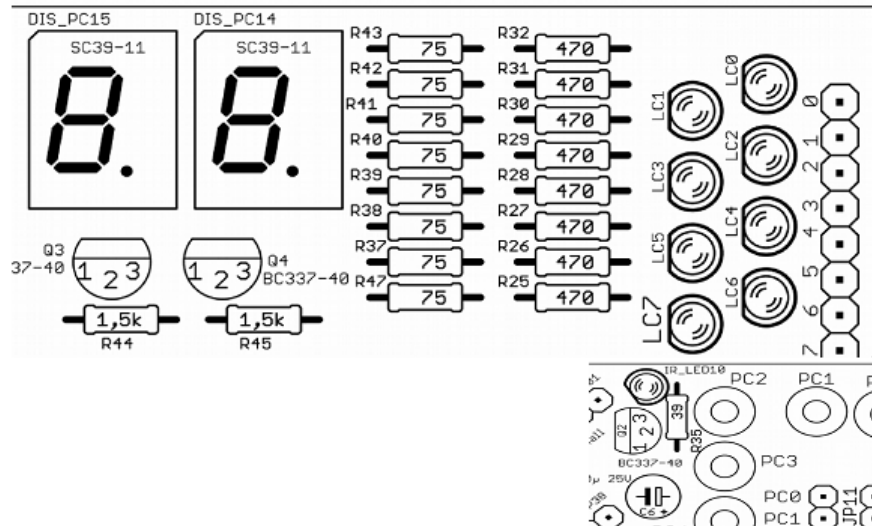
1. Aufgabenstellung: Durch Blinken sollen an Dis\_PC14 nacheinander die Ziffern von Telefonnummern angezeigt werden.
2. Beschreibung der 7-Segment-Anzeige:

## LED und Siebensegmentanzeige

Die LEDs und die Siebensegmentanzeige sind an den Ports PC0 .. PC7 angeschlossen. Die Ports sind als Outputs vorprogrammiert. Hinweis: An den selben Ports sind auch die Hochstromausgänge angeschlossen. Die Ausgänge sind highaktiv. Die beiden Digits der Siebensegmentanzeige werden gemultiplext (d. h. in schnellem Wechsel abwechselnd eingeschaltet). PC15=1 schaltet die Zehnerstelle, PC14=1 die Einerstelle, ein.



Zuordnung:  
a = PC0 g=PC6  
b = PC1 dp = PC7  
c = PC2  
d = PC3  
e = PC4  
f = PC5



Die Ausgabe von 0b00000110b wird beispielsweise eine 1 angezeigt. 0b11111111 bedeutet alles an. 0b00000000 bedeutet alles aus.

- ### 3. Tabellen im Mikrocontroller:

siebensegment:

```
.byte 0b00111111, 0b00000110, 0b01011011, ....
```

Telefonnummer:

```
.byte 1,1,8,3,3
```

Fragen:

Analysieren Sie obige Programmzeile.

Welchen Wert hat der R0 nach folgende Instruktionen:

```
mov R0,#3
ldr R7,=Telefonnummer
ldrb R0,[R7,R0]
```

## Aufgaben:

Aufgabe 1: Erstellen Sie eine Codetabelle für die Telefonnummer: 11833. Als zusätzliche Zahl soll als letztes die Zahl 255 in Ihrer Tabelle erscheinen (kennzeichnet das Tabellenende, Telefonnummern können unterschiedlich lang sein)

Aufgabe 2: Erstellen Sie eine Codetabelle für die Siebensegmentanzeige

Aufgabe 3: Erstellen Sie ein PAP, in dem nacheinander die Ziffern der Telefonnummer aus der Codetabelle geholt werden. Die Ziffern sollen unter Verwendung der Codetabelle aus Aufgabe 2 auf die 7 Segmentanzeige gegeben werden. Die Anzeigedauer soll 1s betragen. Zwischen den Ziffern soll 1 Sekunde Pause sein. Wenn alle Ziffern angezeigt sind (Code 255) soll 3 s Pause sein.

Die Wartezeit muss in einem Unterprogramm `wartel`s mit Zeitschleife realisiert werden

Aufgabe 4:

Schreiben Sie den Programmcode in Assembler

Testen Sie Ihr Programm.

Aufgabe 5: Jetzt sollen mehrere Telefonnummern gespeichert werden. Die Auswahl erfolgt mit den Schaltern (GPIOB).

## Lösung

```
.equ Einerstelle,Bit14
.equ Zehnerstelle,Bit15
```

### main:

```
    bl      startup           //Startupcode
    ldr     R5,=GPIOC        //R5 verweist auf GPIOC
    mov     R4,#0            //Zähler in R4 mit 0 beginnen
    ldr     R6,=seg7         //R6 verweist auf die Tabelle seg7
    ldr     R7,=TelNr
    mov     R0,Bit14         //Anzeige einschalten
    str     R0,[R5,ODR]      //Bit14 von GPIOC.ODR
schleife: //Endlosschleife
    ldrb    R0,[R7,R4]       //Ziffer holen
    cmp     R0,#255
    beq     vonVorne
    ldrb    R0,[R6,R0]       //R0=seg7[R1]
    strb    R0,[R5,ODR]      //R0 auf die Siebensegmentanzeige ausgeben
    mov     R0,#1000
    bl      HAL_Delay

    mov     R0,#0            //kurze Pause
    strb    R0,[R5,ODR]
    mov     R0,#100
    bl      HAL_Delay

    add     R4,#1            //nächste Ziffer

    b       schleife
```

### vonVorne:

```
    mov     R0,0             //Aus
    strb    R0,[R5,ODR]
    mov     R0,#3000         //3s
    bl      HAL_Delay
    mov     R4,#0            //von vorne
    b       schleife
```

### seg7:

```
.byte 0b00111111, 0b00000110, 0b01011011, 0b01001111, 0b01100110, 0b01101101, 0b01111101,
0b00000011, 0b01111111, 0b01101111
```

### TelNr:

```
.byte 1,1,8,3,3,255
.end
```