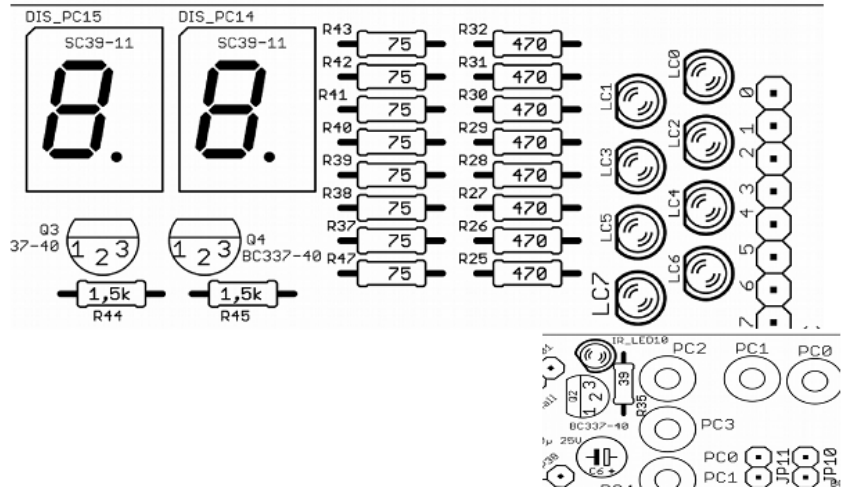
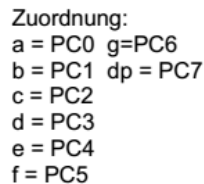


Boardbeschreibung GSOE Base-Shield V1.0

2. 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.



Siebensegmentanzeigen heißen Siebensegmentanzeige weil sie 7 Segmente hat. Beide Digits sind parallel an Port PC angeschlossen. Die Auswahl, welches Digit leuchtet erfolgt mit PC14 (Einerstelle) und PC15 (Zehnerstelle).

Die Ausgabe von 0b00000110 wird beispielsweise als eine 1 angezeigt.

Aufgabe 1: Test der Siebensegmentanzeige und Ermittlung der Kodierung.

Mittels folgendem Testprogramm können Sie die Siebensegmentanzeige mit den Schalterchen an Port GPIOB durchprobieren und die Tabelle ausfüllen:

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

```
main:
ldr R2,=GPIOC
ldr R3,=GPIOB
mov R0,Einerstelle //Einerstelle wird eingeschaltet
str R0,[R2,BSR]
```

```

schleife:
ldrb R0,[R3,IDR]    //In der Schleife werden die Schalterchen an GBIOB
strb R0,[R2,ODR]    //auf die Anzeige (GBIOC) übertragen.
b schleife
.end

```

<i>Ziffer</i>	<i>Code</i>
0	0b00111111
1	
2	
3	
4	
5	
6	
7	
8	
9	

Aufgabe 2: Verwendung einer ROM-Tabelle (Array)

a) Tabelle anlegen nach **b schleife** und vor **.end**

```

...
b schleife
seg7:
.byte 0b00111111, ...
.end

```

Code für 0

Code für 1 , 2 usw. immer mit Komma getrennt

b) Ausgabe mit Tabelle (Array)

Ausgabe: //R0 beinhaltet den Tabellenindex: Ausgabe=seg7[R1];

```

ldr    R3,=seg7
ldrb   R0,[R3,R1]      // Schreibweise in C#: R0=R3[R1]
strb   R0,[R2,ODR]

```

c) Schreiben Sie ein Programm, das die Anzahl der Tastenbetätigungen von PA10 in R1 zählt und an der Einerstelle ausgibt. Beachten Sie, nach 9 soll es wieder mit 0 weitergehen. Beachten Sie weiter, dass immer auf Taste losgelassen, gewartet wird.

d) PAP von c)

Hilfe: Siehe Eisenbahnvideo

Anhang Codeschnipsel

```
schleife:      //Endlosschleife
```

```
main:  
    bl      startup      //Startupcode
```

```
ldr      R2,=GPIOC      //R2 verweist auf GPIOC
```

```
ldrb     R0,[R3,R1]      //R0=seg7[R1]
```

```
mov      R1,#0           //Zähler in R1 mit 0 beginnen
```

```
gedrueckt:      //wenn PA10-Taste gedrückt  
    add     R1,#1         //um 1 hochzählen
```

```
ldr      R3,=seg7        //R3 verweist auf die Tabelle seg7
```

```
ldr      R4,=GPIOA       //R4 verweist auf GPIOA
```

```
b        schleife       //von vorne
```

```
mov      R0,Bit14        //Anzeige einschalten  
str      R0,[R2,ODR]     //Bit14 von GPIOC.ODR
```

```
b        schleife       //von vorne
```

```
ldr      R0,[R4,IDR]     //GPIOA.IDR einlesen (Bit10=Taster PA10)  
tst      R0,Bit10        //Taster prüfen  
bne      gedrueckt       //nicht 0 dann gedrückt
```

```
nochgedrueckt:      //warten solange Taste-PA10 noch gedrückt  
    ldr     R0,[R4,IDR]   //GPIOA.IDR einlesen  
    tst     R0,Bit10      //Bit10 prüfen  
    bne     nochgedrueckt //bei noch gedrückt erneut Taste abfragen
```

```
seg7: //Codetabelle der Siebensegmentanzeige  
.byte 0b00111111, ...
```

```
strb     R0,[R2,ODR]     //R0 auf die Siebensegmentanzeige ausgeben
```

```
cmp      R1,#10          //R1==10  
bne      schleife        //nein, dann von vorne  
mov      R1,#0           //ja, dann R1 wieder bei 0 beginnen
```