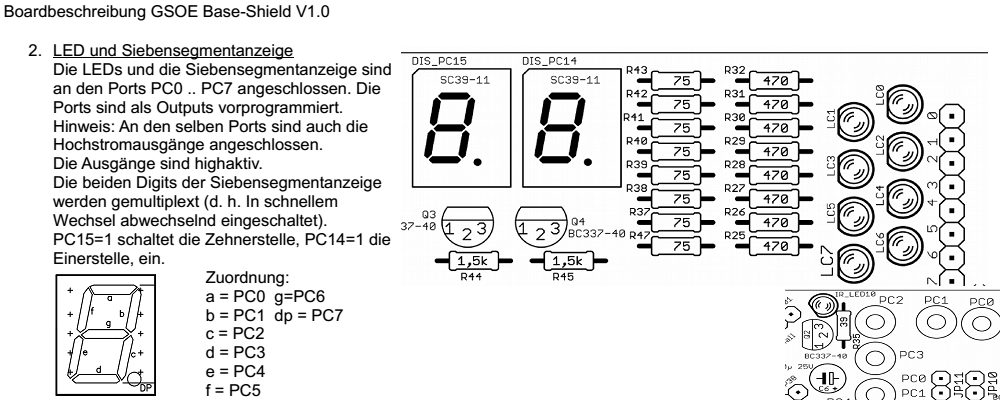
Auf der Mikrocontrollerplatine befinden sich 2 Siebensegmentanzeigen: DIS\_PC15 und DIS\_PC14.

**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 R1,=GPIOC**

**ldr R2,=GPIOB**

**mov R0,Bit14 ;Einerstelle wird eingeschaltet**

**str R0,[R1,BSR]**

schleife:

**ldrb R0,[R2,IDR] //In der Schleife werden die Schalterchen an GBIOB**

**strb R0,[R1,ODR] //auf die Anzeige (GBIOC) übertragen.**

b schleife

.end

|  |  |
| --- | --- |
| ***Ziffer*** | ***Code*** |
| 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

**Anzeige:**

**.byte 0b00111111, ...**

.end

b) Ausgabe mit Tabelle (Array)

Ausgabe: //R0 beinhaltet den Tabellenindex: Ausgabe=Anzeige[R0];

**ldr R3,=Anzeige**

**ldrb R4,[R3,R0] // Schreibweise in C#: R4=R3[R0]**

**strb R4,[R1,ODR]**

c) Schreiben Sie ein Programm, das die Anzahl der Tastenbetätigungen von PA10 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