

VEREIN ZUR FÖRDERUNG DER ICT-BERUFSBILDUNG



ICT-SCHNUPPERTAGE

Elektroniker/in

Autor:
Ervin MAZLAGIĆ

Auftraggeber:
Freddy RINGIER

8. Juli 2013



Inhaltsverzeichnis

1	Lösungen	2
1.1	Aufgabe 1	2
1.2	Aufgabe 2	3
1.3	Aufgabe 3	4
1.4	Aufgabe 4	6
1.5	Aufgabe 5	7
1.6	Aufgabe 6	8
1.7	Aufgabe 7	10
1.8	Aufgabe 8	11
1.9	Aufgabe 9	12

1 Lösungen

1.1 Aufgabe 1

- (a) Die einzelnen Zeilen im Programm sind jeweils Dokumentiert mit einem Kommentar. Kommentare erkennt man daran, dass diese mit einem `'` eingeleitet und in der Entwicklungsumgebung grün dargestellt werden. Alles was rechts von diesem Zeichen steht, ist ein Kommentar und wird vom Computer ignoriert.

Beispiel: In der Zeile 3 steht der Kommentar `'LED 0 einschalten`.

- Als erstes wird mittels `HIGH 0` die LED 0 eingeschaltet.
- Danach wird eine Pause von 500 Millisekunden ($\frac{1}{2}$ Sekunde) mit `PAUSE 500` eingelegt.
- Mit `LOW 0` wird die LED 0 wieder ausgeschaltet.
- Erneut wird wieder mit `PAUSE 500` eine halbe Sekunde lang nichts gemacht.
- Mit `GOTO start` wird als nächstes die Zeile im Programm ausgeführt wo das Label `start` steht (d.h. in Zeile 3). Durch das `GOTO start` wird das Programm unendlich oft wiederholt, da es am Ende immer an den Anfang (`start`) springt.

- (b) Um das Programm langsamer zu machen, muss die Pausenzeit verlängert werden. Man könnte diese z.B. auf 1000ms setzen.

```
1  '{ $STAMP BS1}
2
3  start:  HIGH  0           'LED 0 einschalten
4          PAUSE 1000        'eine Sekunde warten
5          LOW   0           'LED 0 ausschalten
6          PAUSE 1000        'eine Sekunde warten
7          GOTO  start       'springe zu "start"
```

- (c) Um eine andere LED blinken zu lassen muss die Nummer hinter `HIGH` und `LOW` geändert werden, z.B. auf 7.

```
1  '{ $STAMP BS1}
2
3  start:  HIGH  7           'LED 7 einschalten
4          PAUSE 500        'halbe Sekunde warten
5          LOW   7           'LED 7 ausschalten
6          PAUSE 500        'halbe Sekunde warten
7          GOTO  start       'springe zu "start"
```

1.2 Aufgabe 2

- (a) Das Programm schaltet eine LED nach der anderen ein und wieder aus. Es beginnt dabei bei der LED 0 und nach der LED 7 macht es von vorne weiter.
- (b)
- (c) Um eine LED nach der anderen ein- und ausschalten zu lassen von der LED 7 beginnend zur LED 0 müsste wie folgt aussehen.

```
1  '{$STAMP BS1}
2
3  start:  HIGH  7          'LED 7 einschalten
4          PAUSE 500        'halbe Sekunde warten
5          LOW   7          'LED 7 ausschalten
6
7          HIGH  6          'LED 6 einschalten
8          PAUSE 500        'halbe Sekunde warten
9          LOW   6          'LED 6 ausschalten
10
11         HIGH  5
12         PAUSE 500
13         LOW   5
14
15         HIGH  4
16         PAUSE 500
17         LOW   4
18
19         HIGH  3
20         PAUSE 500
21         LOW   3
22
23         HIGH  2
24         PAUSE 500
25         LOW   2
26
27         HIGH  1
28         PAUSE 500
29         LOW   1
30
31         HIGH  0
32         PAUSE 500
33         LOW   0
34
35         GOTO  start      'springe zu "start"
```

1.3 Aufgabe 3

- (a) Wir können das Programm aus der vorhergehenden Aufgabe nehmen und diese einfach erweitern.

```
1  '{ $STAMP BS1}
2
3  start:  HIGH  0          'LED 0 einschalten
4          PAUSE 500        'halbe Sekunde warten
5          LOW   0          'LED 0 ausschalten
6
7          HIGH  1          'LED 1 einschalten
8          PAUSE 500        'halbe Sekunde warten
9          LOW   1          'LED 1 ausschalten
10
11         HIGH  2
12         PAUSE 500
13         LOW   2
14
15         HIGH  3
16         PAUSE 500
17         LOW   3
18
19         HIGH  4
20         PAUSE 500
21         LOW   4
22
23         HIGH  5
24         PAUSE 500
25         LOW   5
26
27         HIGH  6
28         PAUSE 500
29         LOW   6
30
31         HIGH  7
32         PAUSE 500
33         LOW   7
34
35         HIGH  6          'LED 6 ein- und ausschalten
36         PAUSE 500        'sonst wird die LED 7 doppelt
37         LOW   6          'blinken!
38
39         HIGH  5
40         PAUSE 500
41         LOW   5
42
43         HIGH  4
44         PAUSE 500
45         LOW   4
46
47         HIGH  3
48         PAUSE 500
49         LOW   3
50
```

```
51      HIGH  2
52      PAUSE 500
53      LOW   2
54
55      HIGH  1      'Nur bis zur LED 1 gehen da
56      PAUSE 500    'die LED 0 bei "start" schon
57      LOW   1      'blinkt!
58
59      GOTO  start  'springe zu "start"
```



1.4 Aufgabe 4

- (a) Um das Programm schneller zu machen muss lediglich die Pausenzeit kleiner gewählt werden. Zum Beispiel auf 100ms mit `PAUSE 100`.
- (b) Ihnen sollte aufgefallen sein, dass die Veränderung der Pausenzeit eine einfache Aufgabe ist, jedoch sehr mühsam da man diese an vielen Stellen im Programm von Hand ändern muss.

1.5 Aufgabe 5

- (a) Die Taste 0 wird als Eingabe definiert. Ist diese gedrückt beim Label `start` so springt das Programm zum Label `abc`. Dort angelangt wird die LED 4 eingeschaltet mit `HIGH 4`. Solange die Taste gedrückt ist, bleibt die LED 4 eingeschaltet, denn es springt zu `start` und gleich danach zu `abc`. Lässt man die Taste nun los geht es nicht zu `abc` sondern einfach zur nächsten Zeile im Programm. Dort steht dann `LOW 4` was die LED 4 ausschaltet.

Zusammengefasst:

Drückt man die Taste 0, so leuchtet die LED 4. Lässt man sie los, so leuchtet sie nicht.

- (b) Drückt man eine Taste, so leuchtet immer die zugehörige LED. Das hat nichts mit dem Programm zu tun sondern ist etwas spezielles am Board selbst.

1.6 Aufgabe 6

- (a) Einerseits müssen wir mehr Taster definieren und andererseits müssen wir mehr Sprungstellen definieren.

```
1  '{ $STAMP BS1}
2
3  INPUT 0           'Definiere PIN 0 als Eingabe für die Taste 0
4  INPUT 1
5  INPUT 2
6  INPUT 3
7
8  start:  IF PIN0 = 1 THEN show1    'Falls PIN0=1, dann "show1"
9          IF PIN1 = 1 THEN show2
10         IF PIN2 = 1 THEN show3
11         IF PIN3 = 1 THEN show4
12
13         LOW 4           'LED 4 ausschalten
14         LOW 5
15         LOW 6
16         LOW 7
17         GOTO start      'springe zu "start"
18
19 show1:  HIGH 4          'LED 4 einschalten
20         GOTO start      'springe zu "start"
21
22 show2:  HIGH 5
23         GOTO start
24
25 show3:  HIGH 6
26         GOTO start
27
28 show4:  HIGH 7
29         GOTO start
```

- (b) Es muss zwei mal nach einander angefragt werden ob die Tasten gedrückt sind. Zudem können die Tasten in verschiedenen Reihenfolgen gedrückt werden.

```
1  '{ $STAMP BS1}
2
3  INPUT 0           'Definiere PIN 0 als Eingabe für die Taste 0
4  INPUT 1
5  INPUT 2
6  INPUT 3
7
8  start:  IF PIN0 = 1 THEN test1
9          IF PIN1 = 1 THEN test0
10         IF PIN2 = 1 THEN test3
11         IF PIN3 = 1 THEN test2
12
13 test0:  IF PIN0 = 1 THEN show5
14         LOW 5
15         GOTO start
16
17 test1:  IF PIN1 = 1 THEN show5
```

```
18          LOW  5
19          GOTO start
20
21 test2:    IF PIN2 = 1 THEN show7
22          LOW  7
23          GOTO start
24
25 test3:    IF PIN3 = 1 THEN show7
26          LOW  7
27          GOTO start
28
29 show5:    HIGH 5
30          GOTO start
31
32 show7:    HIGH 7
33          GOTO start
```

- (c) Es gibt insgesamt 16 Kombinationen, denn man hat vier Taster mit jeweils zwei Zuständen (Ein/Aus) und somit $2^4 = 2 \cdot 2 \cdot 2 \cdot 2 = 16$ Kombinationen.

1.7 Aufgabe 7

- (a) ...
- (b) ...
- (c) Ein Zyklus soll 15 Sekunden dauern. In solch einen Zyklus soll eine gerade Anzahl Pausen mit 300 Millisekunden stattfinden. Cooles Problem denn wir können einfach Algebra verwenden:

$$15 \text{ Sekunden} = x \cdot 2 \cdot \text{Pausenzeit}$$

$$15 = x \cdot 2 \cdot 0.3 \quad | \div 2$$

$$\frac{15}{2} = x \cdot 0.3 \quad | \div 0.3$$

$$\frac{15}{2 \cdot 0.3} = x \quad | \Leftrightarrow x$$

$$x = \frac{15}{2 \cdot 0.3}$$

$$x = \frac{15}{0.6}$$

$$x = 25$$

Nun wissen wir, dass wir 25 Zyklen brauchen. Da wir bei 0 und nicht bei 1 zu Zählen beginnen, müssen wir 1 abziehen d.h. wie schreiben `FOR B2 = 0 TO 24`.

```
1  '{ $STAMP BS1}
2
3  INPUT 0
4
5  start:  IF PIN0 = 1 THEN blink
6          GOTO start
7
8  blink:  FOR B2 = 0 TO 24
9          HIGH 7
10         PAUSE 300
11         LOW 7
12         PAUSE 300
13     NEXT
14
15     GOTO start
```



1.8 Aufgabe 8

- (a) ...
- (b) ...
- (c) ...

1.9 Aufgabe 9

- (a) ...
- (b) ...
- (c) Code

```

1  '{ $STAMP BS1}
2
3  SYMBOL LED = B2           'B2 soll "LED" heissen
4  LED = 0                   'Wert 0 wird in LED gespeichert
5
6      HIGH LED
7      PAUSE 1000
8
9  start: B3 = 0              'Wert 0 in B3 speichern
10     FOR B3 = 0 TO 6        'zähle B3 von 0 bis 6
11         LOW led
12         LED = LED + 1      'erhöhe Wert in LED um 1
13         HIGH LED
14         PAUSE 1000
15     NEXT
16                             'falls B3<6: erhöhe B3 um 1,
17                             'sonst weiter im Programm
18
19     B3 = 0                  'Wert 0 in B3 speichern
20
21     FOR B3 = 0 TO 6        'zähle B3 von 0 bis 6
22         LOW LED
23         LED = LED - 1      'erniedrige Wert in LED um 1
24         HIGH LED
25         PAUSE 1000
26     NEXT
27                             'falls B3<6: erhöhe B3 um 1
28                             'sonst weiter im Programm
29
30     GOTO start              'springe zu "start"

```

- (d) Code

```

1  '{ $STAMP BS1}
2
3  SYMBOL LED1 = B2          'B2 soll "LED1" heissen
4  SYMBOL LED2 = B4          'B4 soll "LED2" heissen
5
6  LED1 = 4                  'Wert 4 in LED1 speichern
7  LED2 = 0                  'Wert 0 in LED2 speichern
8
9  HIGH LED1                 'LED1 einschalten
10 HIGH LED2                 'LED2 einschalten
11
12 PASUE 200                 'warte 200ms
13
14 start: FOR B3 = 0 TO 6     'zähle B3 von 0 bis 6
15         LOW LED1          'LED1 ausschalten

```

```
16          LOW    LED2          'LED2 ausschalten
17          LED1 = LED1 + 1      'Wert in LED1 um 1 erhöhen
18          LED2 = LED2 + 1      'Wert in LED2 um 1 erhöhen
19          HIGH    LED1          'LED1 einschalten
20          HIGH    LED2          'LED2 einschalten
21          PAUSE 200            'warte 200ms
22          NEXT              'B3<6: erhöhe B3 um 1,
23                          'sonst weiter im Programm
24
25          GOTO start
```