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 (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.

```
'{$STAMP BS1}

start:  HIGH  7         'LED 7 einschalten
        PAUSE 500       'halbe Sekunde warten
        LOW   7         'LED 7 ausschalten

        HIGH  6         'LED 6 einschalten
        PAUSE 500       'halbe Sekunde warten
        LOW   6         'LED 6 ausschalten

        HIGH  5
        PAUSE 500
        LOW   5

        HIGH  4
        PAUSE 500
        LOW   4

        HIGH  3
        PAUSE 500
        LOW   3

        HIGH  2
        PAUSE 500
        LOW   2

        HIGH  1
        PAUSE 500
        LOW   1

        HIGH  0
        PAUSE 500
        LOW   0

        GOTO  start     'springe zu "start"
```

1.3 Aufgabe 3

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

```
'{$STAMP BS1}

start:  HIGH  0         'LED 0 einschalten
        PAUSE 500       'halbe Sekunde warten
        LOW   0         'LED 0 ausschalten

        HIGH  1         'LED 1 einschalten
        PAUSE 500       'halbe Sekunde warten
        LOW   1         'LED 1 ausschalten

        HIGH  2
        PAUSE 500
        LOW   2

        HIGH  3
        PAUSE 500
        LOW   3

        HIGH  4
        PAUSE 500
        LOW   4

        HIGH  5
        PAUSE 500
        LOW   5

        HIGH  6
        PAUSE 500
        LOW   6

        HIGH  7
        PAUSE 500
        LOW   7

        HIGH  6         'LED 6 ein- und ausschalten
        PAUSE 500       'sonst wird die LED 7 doppelt
        LOW   6         'blinken!

        HIGH  5
        PAUSE 500
        LOW   5

        HIGH  4
        PAUSE 500
        LOW   4

        HIGH  3
        PAUSE 500
        LOW   3

```

```
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 Sprungsstellen definieren.

```
'{$STAMP BS1}

INPUT 0             'Definiere PIN 0 als Eingabe für die Taste 0
INPUT 1
INPUT 2
INPUT 3

start:  IF PIN0 = 1 THEN show1    'Falls PIN0=1, dann "show1"
        IF PIN1 = 1 THEN show2
        IF PIN2 = 1 THEN show3
        IF PIN3 = 1 THEN show4

        LOW  4                      'LED 4 ausschalten
        LOW  5
        LOW  6
        LOW  7
        GOTO start                  'springe zu "start"

show1:  HIGH 4                      'LED 4 einschalten
        GOTO start                  'springe zu "start"

show2:  HIGH 5
        GOTO start

show3:  HIGH 6
        GOTO start

show4:  HIGH 7
        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.

```
'{$STAMP BS1}

INPUT 0             'Definiere PIN 0 als Eingabe für die Taste 0
INPUT 1
INPUT 2
INPUT 3

start:  IF PIN0 = 1 THEN test1
        IF PIN1 = 1 THEN test0
        IF PIN2 = 1 THEN test3
        IF PIN3 = 1 THEN test2

test0:  IF PIN0 = 1 THEN show5
        LOW  5
        GOTO start

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 25`.

```
'{$STAMP BS1}

INPUT 0

start:  IF PIN0 = 1 THEN blink
        GOTO  start

blink:  FOR B2 = 0 TO 24
            HIGH  7
            PAUSE 300
            LOW   7
            PAUSE 300
        NEXT

        GOTO  start
```

1.8 Aufgabe 8

(a) ...

(b) ...

(c) ...

1.9 Aufgabe 9

(a) ...

(b) ...

(c) Code

```
'{$STAMP BS1}

SYMBOL LED = B2                 'B2 soll "LED" heissen
LED = 0                         'Wert 0 wird in LED gespeichert

        HIGH  LED
        PAUSE 1000

start:  B3 = 0                  'Wert 0 in B3 speichern
        FOR B3 = 0 TO 6         'zähle B3 von 0 bis 6
            LOW led
            LED = LED + 1       'erhöhe Wert in LED um 1
            HIGH  LED
            PAUSE 1000
        NEXT                    'falls B3<6: erhöhe B3 um 1,
                                'sonst weiter im Programm

        B3 = 0                  'Wert 0 in B3 speichern

        FOR B3 = 0 TO 6         'zähle B3 von 0 bis 6
            LOW   LED
            LED = LED - 1       'erniedrige Wert in LED um 1
            HIGH  LED
            PAUSE 1000
        NEXT                    'falls B3<6: erhöhe B3 um 1
                                'sonst weiter im Programm

        GOTO start              'springe zu "start"
```

(d) Code

```
'{$STAMP BS1}

SYMBOL LED1 = B2                'B2 soll "LED1" heissen
SYMBOL LED2 = B4                'B4 soll "LED2" heissen

LED1 = 4                        'Wert 4 in LED1 speichern
LED2 = 0                        'Wert 0 in LED2 speichern

HIGH LED1                       'LED1 einschalten
HIGH LED2                       'LED2 einschalten

PASUE 200                       'warte 200ms

start:  FOR B3 = 0 TO 6         'zähle B3 von 0 bis 6
            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
```