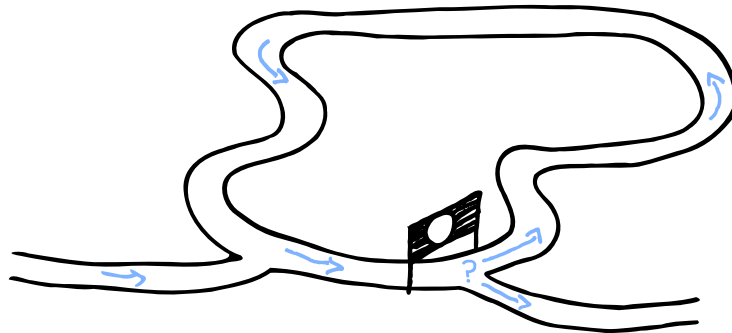


Arbeitsblatt 13 - while-Schleife

Eine while-Schleife dient zur wiederholten Durchführung:



Die Kontrollstruktur **while-Schleife** wird verwendet, um einen bestimmten Programmteil mehrfach zu **wiederholen**. Sie ermöglicht, dass in Abhängigkeit von einer **Bedingung** bestimmte Anweisungen **solange immer wieder** ausgeführt werden, bis die Bedingung **nicht mehr** erfüllt ist.

```
while (Bedingung) {
    Anweisungen;
}
```

Solange die Bedingung **wahr** ist, werden die **Anweisungen** ausgeführt. Die Bedingung wird am Anfang der Schleife, also **vor** dem ersten Durchlauf geprüft. Falls die Bedingung schon bei der ersten Prüfung falsch ist, wird die Schleife gar nicht ausgeführt.

Wir schauen uns mal ein konkretes Beispiel an:

```
5 while (bob3.getArm(1) > 0){
6     bob3.setWhiteLeds(ON, ON);
7 }
8
9 bob3.setWhiteLeds(OFF, OFF);
```

Bedingung

Anweisung

In **Zeile 5** wird die **Bedingung** `bob3.getArm(1)>0` geprüft: Solange die **Bedingung wahr** ist, also wenn die Abfrage von Arm1 eine Zahl größer als 0 liefert, dann werden beide Bauch-Leds weiß eingeschaltet. Anschließend, wenn der Arm nicht mehr berührt wird, werden die beiden Leds ausgeschaltet. **Solange** wir also den Arm1 anfassen, leuchten die beiden Bauch-Leds!

Was genau ist eine Endlosschleife?

Man kann eine while-Schleife auch als **Endlosschleife** programmieren, wir schreiben dann: **while (true)**. Eine Endlosschleife kann z.B. für eine Alarmanlage sehr nützlich sein:

```

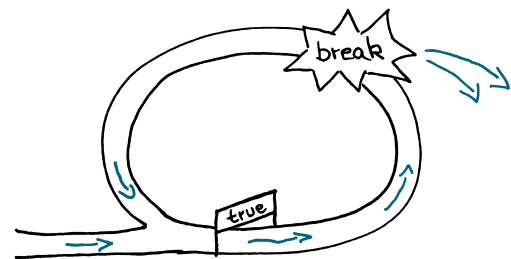
4
5   if (bob3.getIRSensor()>8) {
6     while (true) {
7       bob3.setWhiteLeds(ON, ON);
8       delay(200);
9       bob3.setWhiteLeds(OFF, OFF);
10      delay(200);
11    }
12  }
    
```

Sobald der Alarm **ausgelöst** wurde (Bedingung in Zeile 5 erfüllt), startet ein **Warnblinklicht** mit den weissen Bauch-Leds in einer **Endlosschleife!**

Macht das Sinn? Ja, denn wenn der Alarm einmal ausgelöst wurde, dann soll Bob blinken und nicht mehr aufhören!

Die break-Anweisung:

Für Schleifen gibt es eine wichtige Anweisung, die **break-Anweisung**. Die break-Anweisung steht irgendwo innerhalb der Schleife und wird meistens in Kombination mit einer *if-Abfrage* verwendet. Sobald ein bestimmter Zustand eintritt und der Compiler bei der break-Anweisung ankommt, wird die Schleife **abgebrochen**.



```

5   if (bob3.getIRSensor()>8) {
6     while (true) {
7       bob3.setWhiteLeds(ON, ON);
8       delay(200);
9       bob3.setWhiteLeds(OFF, OFF);
10      delay(200);
11
12      if (bob3.getArm(1) == 3) {
13        break;
14      }
15    }
16  }
    
```

Abbruch!

Wir verwenden in Zeile 13 eine break-Anweisung, um unsere while-Schleife zu verlassen, also **abzubrechen**, **sobald** Arm 1 unten berührt wird!

Aufgabe 1: Wie oft wird die Schleife **while (100 == 100)** ausgeführt?

- keinmal
- einmal
- zweimal
- einhundertmal
- immer und immer wieder

Aufgabe 2: Wie oft wird die Schleife **while (17 < 5)** ausgeführt?

- keinmal
- einmal
- fünfmal
- siebzehnmals
- immer und immer wieder

Aufgabe 3: Wie oft wird die Schleife **while (1 != 1)** ausgeführt?

- keinmal
- einmal
- zweimal
- dreimal
- immer und immer wieder

Aufgabe 4: Wie oft wird die Schleife **while (1 != 2000)** ausgeführt?

- keinmal
- einmal
- zweimal
- zweitausendmal
- immer und immer wieder

Aufgabe 5: Was passiert nach der **break**-Anweisung in Zeile 12?

```

1 #include <BOB3.h>
2
3 void loop() {
4
5     if (bob3.getIRSensor()>8) {
6         while (true) {
7             bob3.setWhiteLeds(ON, ON);
8             delay(200);
9             bob3.setWhiteLeds(OFF, OFF);
10            delay(200);
11            if (bob3.getArm(1) == 3) {
12                break;
13            }
14        }
15        delay(2000);
16    }
17 }
    
```

- Es wird die Zeile 5 ausgeführt
- Es wird die Zeile 6 ausgeführt
- Es wird die Zeile 7 ausgeführt
- Es wird die Zeile 11 ausgeführt
- Es wird die Zeile 15 ausgeführt

Aufgabe 6: Betrachte das folgende Programm. Für welchen Zweck könnte man es verwenden? Hast du mehrere Ideen? Schreibe deine Ideen auf!

```

1 #include <BOB3.h>
2
3 void loop() {
4
5     while (bob3.getArm(1) == 0) {
6         delay(1);
7     }
8     bob3.setEyes(WHITE, WHITE);
9     bob3.setWhiteLeds(ON, ON);
10    delay(60000);
11    bob3.setEyes(OFF, OFF);
12    bob3.setWhiteLeds(OFF, OFF);
13 }
14
    
```
