

Bitte beachten: Bevor man einen neuen Sketch „verdrahtet“ muss der Arduino „gelöscht“ werden, d.h. man muss das „BareMinimum“ laden. Schaltung erst prüfen lassen! Dann USB-Kabel reinstecken.

1. Der erste Sketch

Schließe an PIN13 zusätzlich die LED des Shield's an. Öffne das Programm „Blink“ und lasse es laufen. Nehme Veränderungen am Programm vor und beobachte, wie sich die LED verhält. Ermittle, wie schnell (also wieviel mal pro Sekunde) du die LED blinken lassen kannst, bis es so wirkt als sei sie ständig an.

2. Die erste Variable

Wir können das **delay()** auch variabel gestalten. Wir schreiben also nicht **delay(1000)**, sondern **delay(zeit)**. Die Variable „zeit“ muss deklariert und der Anfangswert festgelegt werden.

- a) „zeit“ ist nun im LOOP hochzuzählen, z.B. **zeit=zeit+100;**. Verwende als OUTPUT den PIN 12 und schließe die LED hier an.
- b) Durch eine **if-else** Abfrage können wir erreichen, dass z.B. zu Beginn das Hochzählen langsam (**zeit=zeit+5**); und ab einem bestimmten Zeitwert schneller läuft (**zeit=zeit+50**); Syntax für if-else: **if (Bedingung) {irgendwas} else {irgendwas anderes}**. Baue diese Abfrage in den Sketch ein und teste ihn.

3. Plötzlich Töne

- a) Öffne wieder den einfachen „normalen“ Blink Sketch und verwende als Ausgang PIN9. Ändere das **delay()** von 1000 auf z.B. 2 ms und. Schließe nun zwischen PIN9 und GND einen Lautsprecher an. Dann hast du einen Tongenerator. Warum? Erzeuge nacheinander verschiedene Töne.
- b) Jetzt wird's schon schwieriger: Wir definieren einen Zähler, der nach jedem LOOP eins weiter zählt, also **int n=0;** (Deklariere die Variable), dann **n=n+1;** dann fragen wir in einer **if**-Abfrage ob z.B. **n>2000** ist, falls ja erhöhen wir das **delay(zeit)** z.B. von **zeit=2** auf **3(ms)**. Wenn alles stimmt, wird sich die Tonfrequenz nach einer bestimmten Zeit ändern.

4. Merkwürdiges Blinken

Wir verwenden wieder den originalen „Blink“-Sketch. Nun soll die LED aber nur genau 5-mal blinken. Dazu verwenden wir zunächst eine if-Abfrage und definieren einen Zähler mit der Zählvariablen **n** durch **int n=0;** Hier wird der Startwert auf 0 gesetzt.

I) Führe die Deklaration der Variablen probeweise durch

- a) im globalen Bereich (global sector)
b) im Setup,
c) im Loop

und beobachte das Verhalten der LED. Die letzte Anweisung im Loop muss **n=n+1;** lauten (entspricht **n++**). Vergleiche jeweils den Programmablauf.

II) Definiere **n** wieder wie in a) und setze die Anweisung **n=n+1;** im Loop **vor** die if-Abfrage. Füge hinter die if-Abfrage (nicht **im** if-) ein Delay von 1ms ein.

Lasse diesen Sketch mindestens 40 Sekunden lang laufen und beobachte die LED. Diese Merkwürdigkeit werden wir noch klären.....ist nicht ganz einfach ☺ Vielleicht hilft dir folgende Änderung: Wiederhole Aufgabe 4 II) und ersetze **int n=0;** durch **byte n=0;** Dann durch **long n=0;** Beobachte den Ablauf genau und vergleiche.

5. Die for-Schleife

Die LED soll wieder 5-mal blinken. Diesmal verwenden wir jedoch eine **for-Schleife**. Die Zählvariable „n“ könnten wir auch in der for-Schleife definieren. Syntax: **for(int n=0; n<6; n++) {mach irgendwas}**. Möchte man den Wert jeweils um 2 erhöhen, müsste man **n=n+2** schreiben oder **n+=2**, ist aber unanschaulicher.

- a) setze die Schleife in das Setup
b) in den Loop

Beobachte jeweils die LED und versuche den Unterschied zu erklären. Nun definieren wir **n** mit **int n=0;** vor dem Setup im globalen Bereich und schreiben in der for-Schleife anstelle **int n=0;** nur **n;** Teste, was passiert und begründe es.

6. Die while-Schleife

Syntax: **while(expression) { statement(s) },** Beispiel: **int n = 0;**

while(n < 200) { do something; n++; }

Realisiere den Blink-Sketch (LED soll 5-mal blinken) mit Hilfe einer „While-Schleife“. Tipp: Es ist empfehlenswert, den Zähler mit **n--** runter zählen zu lassen.

Erweitere den Sketch, so dass nach 5-mal langsam Blinken ein schnelles Dauerblinken entsteht.