

**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.

**Info:** Sketch = Skizze (hier: Programmentwurf)  
Setup = Konfiguration  
Loop = Schleife des „Betriebssystems“

## 1. Der erste Sketch

- Schließe an PIN13 zusätzlich die LED des Shield's an. Öffne das Programm „Blink“ und lasse es laufen. Ändere das Programm 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.
- Ändere den Aufbau und den Sketch so, dass zwei verschiedenfarbige LEDs in folgender Reihenfolge jeweils blinken: LED1, Pause, LED1+LED2, Pause, LED2, Pause. Die Delays kannst du selbst wählen.

**Info:** Was bedeutet eigentlich „COM3“? COM steht für „Communication-Port“ (Kommunikations-Schnittstelle)

## 2. Die erste Variable, serieller Monitor

Wir wollen nun einen Zähler aufbauen, der einfach von 0 an hochzählt. Da du das beobachten möchtest, benötigen wir den **Seriellen Monitor (SM)**.

Initialisiere die byte-Variable „n“ und setze im Loop  $n=n+1$ , d.h. bei jedem Lauf durch den Loop wird n um 1 erhöht. Dafür kann man auch schreiben  $n=n++$ .

Der SM wird im Setup durch **Serial.begin(9600);** initialisiert. Durch **Serial.print(„Text“);** wird Text ausgegeben und mit **Serial.print(n);** würde der aktuelle Wert der Variablen n angezeigt. Das wird alles nebeneinander geschrieben. Möchtest du eine neue Zeile schreiben, so lautet die Anweisung z.B.

**Serial.println(„Text“);** es wurde also print durch println ersetzt (print new line).

**Info:** Was ist 1 Baud (gesprochen „boud“)?

Baud ist die Einheit für die Symbolrate in der Nachrichtentechnik. Die Baudrate gibt an, wie viele **Symbole** (z.B. Zahlen oder Buchstaben) pro Sekunde übertragen werden. Die Baudrate wird oft mit der Datenübertragungsrate verwechselt: Menge der übertragenen Daten in Bit je Sekunde = Bitrate.

- Schreibe einen Sketch, mit welchem im Seriellen Monitor im 1-2 Sekundentakt eine „Tabelle“ entsteht, in welcher links die natürlichen Zahlen (beginnend mit  $n=1$ ) stehen und im Abstand daneben die dazugehörigen Quadrat-zahlen.
- Erweitere die Tabelle so, dass n und  $n^2$  zusätzlich binär ausgegeben werden mit **Serial.print(Variable,BIN)**.
- Unser Zähler hört nie auf zu zählen, wir möchten aber nun erreichen, dass er nach z.B.  $n=20$  stehen bleibt.

Das gelingt mit Hilfe einer if-Bedingung (vergl. Aufg. 3b)

## 3. Variables delay

Lade wieder den originalen Blink-Sketch. Du kannst das **delay()** auch variabel gestalten. Wir schreiben also nicht delay(1000), sondern delay(zeit).

Die Variable „zeit“ muss definiert und initialisiert werden.

- „zeit“ ist nun im LOOP hochzuzählen, z.B.

**zeit=zeit+100.** Verwende als OUTPUT den PIN 12 und schließe die LED hier an.

- 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) {mache irgendwas} else {mache etwas anderes}**. Baue diese Abfrage in den Sketch ein und teste ihn.

## 4. Plötzlich Töne

- Öffne wieder den einfachen Blink Sketch und verwende als Ausgang PIN9. Ändere das delay() von 1000ms auf z.B. 2 ms. Schließe nun zwischen PIN9 und GND einen Lautsprecher an. Dann hast du einen Tongenerator. Warum? Erzeuge nacheinander verschiedene Töne.

- Jetzt wird's schon schwieriger: Wir definieren einen Zähler, der nach jedem LOOP eins weiter zählt, 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.

**Info:**

### Deklaration

Unter deklarieren einer Variable versteht man das erste "Erwähnen" der Variable, das dem Compiler sagt "Hallo, ich bin da und kann verwendet werden". Der Wert selbst wird bei der Deklaration nicht festgelegt.

### Definition

Durch die Definition wird einer Variablen ein Speicherbereich zugeteilt. Der Speicherbereich hat eine eindeutige Adresse und dient dazu, Werte abspeichern zu können. Mit dem Variablennamen sprechen wir im Prinzip nur einen Speicherbereich an. Unsere bisherige Art und Weise Variablen zu "erstellen", ist eine Deklaration **und** Definition:

**int zahl;** Durch „int“ wird ein Speicherbereich (hier 2 Byte) zugeordnet und „zahl“ deklariert den Namen der Variable.

### Initialisierung

Haben wir eine Variable deklariert und definiert, so hat sie einen beliebigen Wert, je nach dem was gerade im zugewiesenen Speicherbereich steht. Da wir mit solch einem Zufalls-wert nicht arbeiten wollen, können wir mittels Initialisierung die Variable auf einen Anfangswert setzen. Variablen sollten möglichst immer initialisiert werden.

Beispiel: `int a=5, b=-2, c=a+b;`