

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

**Info:** **Setup** = Konfiguration (hier kommen Anweisungen rein, die nur beim ersten Mal (ersten Aufruf) ausgeführt werden. **Loop** = Schleife des „Betriebssystems“ (hier stehen Anweisungen, die endlos bei jedem Aufruf abgearbeitet werden)

## 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“)?

Die **Baudrate** gibt an, wie viele **Symbole** (z.B. Zahlen oder Buchstaben) pro Sek. übertragen werden.

Die **Datenübertragungsrate** ist die Menge der übertragenen Daten in Bit je Sekunde = Bitrate. Im SM ist die Bitrate z.B. 9600 Bit/Sekunde. Die Bezeichnung „Baud“ ist hier falsch!

- 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-Abfrage (vergl. Aufg. 3b)

- Für Profis:** Nun soll zusätzlich der Kehrwert der Zahl, also  $1/n$  ausgegeben werden dafür aber keine Binärzahlen.

- Und zum Schluss: Entwerfe einen Sketch, der auf dem Monitor in je einer Zeile (n), (n, BIN), ( $n^2$ ), ( $n^2$ , BIN) von  $n=125$  bis  $n=100$  in absteigender Reihenfolge innerhalb von 2,5 Sekunden entstehen lässt. Denke auch an die Kopfzeile!

## 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 deklariert und initialisiert werden.

- „zeit“ ist nun im LOOP hochzuzählen, z.B. **zeit=zeit+1;** 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+1;** 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. Je nach Problem ist der „else“-Teil nicht unbedingt notwendig.

### Info:

Eine **Variable** wird **deklariert** indem man ihr einen Namen zuweist und den Datentyp festlegt. Beispiel:

**int Zeit;** „int“ gibt an, wieviel Speicherplatz „reserviert“ wird und „Zeit“ ist der Name der Variable. Hier die wichtigsten Datentypen und deren Mindestspeicherbedarf:

bool	1 Bit	long	4 Byte
byte	1 Byte	float	4 Byte
int	2 Byte		

bool, byte, int und long sind ganzzahlig (integer) und float steht für (Fließ)Kommazahlen.

Eine **Variable** sollte zusätzlich **initialisiert** werden, d.h. man gibt ihr einen Startwert. Beispiel:

**int Pause;** // Deklaration  
**Pause=1000;** // Zuweisung oder Initialisierung  
 Zusammengefasst: **int Pause=1000;**

Es ist guter Programmstil,

\* jeder Variable einen aussagekräftigen Namen zu geben,

\* jede Variable bei der Deklaration zu initialisieren und

\* mit einem Kommentar zu versehen, Beispiel:

**int Pause=1000;** // Zeit in ms für die Dunkelpause zwischen zwei Lichtimpulsen