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

1. Der erste Sketch

a) 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.
b) Ermittle, wie schnell (also wieviel mal pro Sekunde) du die LED blinken lassen kannst, bis es so wirkt als sei sie ständig an.
c) Ändere den Aufbau und den Sketch so, dass zwei verschiedenfarbige LEDs in folgender Reihenfolge jeweils 300ms blinken: LED1, LED1+LED2, LED2 .

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).
Schreibe den Sketch, füge ein sinnvolles delay() ein und teste ihn. Was fällt dir auf?

3. Variables delay

Lade wieder den original Blink-Sketch. Du kannst 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.

4. 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, 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.

5. 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.

a) Beobachte das Verhalten der LED. Die letzte Anweisung im Loop muss **n=n+1;** lauten (entspricht **n++**).

b) Setze nun 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.....sie ist nicht ganz einfach ☺

Vielleicht hilft dir folgende Änderung: Ersetze **int n=0;** durch **byte n=0;** später durch **long n=0;** beobachte den Ablauf genau und vergleiche.

Info zwischendurch:

Deklaration
Mit der Deklaration benennen wir eine Variable und machen diese dem Compiler bekannt. D.h. der Compiler weiß nun, ob wir uns im Laufe unseres Programms beim Eintippen des Variablennamens vertippt haben.

Definition
Durch die Definition wird einer Variable 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 Zufallswert nicht arbeiten wollen, können wir mittels Initialisierung die Variable auf einen Anfangswert setzen. Variablen sollten möglichst immer initialisiert werden, um zu vermeiden, dass mit einem Zufallswert gearbeitet wird. Beispiel: int a=5, b=-2, c=a+b;