1. Analoge Werte lesen und anzeigen

a) Öffne den Sketch **AnalogReadSerial** in Beispiele (01Basics). Schließe an den zuvor gelöschten Arduino das Potentiometer (kurz „Poti“) an (prüfen lassen!). Ändere das Delay auf ca. 100ms. Lasse den Sketch laufen und beobachte den Seriellen Monitor während du am Poti drehst. Der analoge Eingang wird mit einer Auflösung von 10 Bit (0 bis 1023) ausgelesen. Erkläre, was das bedeutet.

b) Aktiviere nun den **Seriellen Plotter (SP)**. Drehe am Poti und beobachte die Kurve.

2. Spannungs-Messgerät programmieren

a) Öffne den Sketch **ReadAnalogVoltage** und teste ihn! Vorher den Arduino löschen! Die Schaltung ist mit der von Aufg. 1) identisch. Delay einfügen!
Dieser Sketch bewirkt im Prinzip, dass der Arduino wie ein digitales Spannungs-Messgerät arbeitet. Auch hier lässt sich der **SP** aktivieren.

b) Miss die Spannung eines Li-Ionen-Akkus. Dazu muss zwischen A0 (analoger Eingang) und GND noch ein (pull-down) Widerstand von 10 KΩ (*braun-schwarz-orange*) angeschlossen werden.

3. Dimmer für LED´s

Nun wollen wir die Helligkeit einer LED mit einem Poti einstellen und dabei mehr über analoge Eingänge und **PWM** (Puls-Weiten-Modulation) lernen. Öffne dazu den Sketch **AnalogInOutSerial** und erhöhe das Delay auf 100ms. Stecke die Elektronik auf dem zuvor gelöschten Arduino zusammen (testen lassen!).

Der Ausgang wird nur auf 8 Bit aufgelöst, kann also Werte zwischen 0 und 255 annehmen. Deshalb müssen die Eingangswerte durch 4 geteilt werden (siehe Besprechung). Die **map-Funktion** benötigt man hier eigentlich nicht. Ersetze diese durch eine Division. Ein Ausgang von 255 entspricht 5V, die LED leuchtet mit voller Helligkeit. Bei einem Wert von 124 leuchtet die LED …… überlege selbst! Schließe nun das Digital-Oszilloskop an: Der schwarze Pin an GND und der rote Pin an die LED. Drehe am Poti und beobachte die Änderung auf dem Oszi.
Für Profis: Versuche anhand der Einstellung des Oszilloskops herauszufinden, wie oft die LED pro Sekunde ein- und ausgeschaltet wird.

4. Arrays verwenden, Lauflicht

Anwendung von **Arrays**: Mit folgendem Programm kannst du auf der 7-Segment-Anzeige ein Lauflicht erzeugen. Schließe dazu die Segmente a bis f der Reihe nach an die Pins 7 bis 12 an. Die gemeinsame Anode wird mit +5V verbunden.

```
int ledPin[6] = {7, 8, 9, 10, 11, 12};
                        // LED-Array mit Pin-Werten
void setup() {
  for(int i = 0; i < 6; i++) {
    pinMode(ledPin[i], OUTPUT);    // Alle Pins des
                       //Arrays als Ausgang deklarieren
  }
}
void loop() {
  for(int i = 0; i < 6; i++) {
    digitalWrite(ledPin[i], LOW);      // Array-
                     //Element (=Pin) auf LOW-Pegel
    delay(50);

    digitalWrite(ledPin[i], HIGH);    // Array-
                                //Element auf HIGH
   }
}
```

Zusatz: Vertausche LOW und HIGH und erkläre….

Auf dem USB-Stick befindet sich das Programm „Lauflicht-mit-Fehlern“. Lade das Programm und finde die Fehler **ohne** auf die oben stehende Lösung zu schauen!!!

5. Elektronisches Keyboard

Zunächst müssen wir die Library **pitches.h** etwas verstehen. Damit lässt sich eine Klaviatur simulieren. Diese Library (Bibliothek) wird hier inkludiert.
Wir wollen mit z.B. drei Tastern verschiedene Töne abspielen. Folgende Probleme sind dabei zu lösen: Es müssen dauernd die Taster abgefragt werden, ob ihr Zustand sich ändert. Ist das bei einem Taster der Fall, so muss ein bestimmter Ton aus einem Array abgespielt werden.
Öffne dazu den Sketch aus den Beispielen (02digital) **toneKeyboard**. Dieser ist für Touch-Sensoren aufgebaut, die wir (noch) nicht haben. Wir verwenden Buttons, die an digitale Eingänge 0,1,2 (besser wäre 2,3,4) angeschlossen und mit **INPUT_PULLUP** auf einen HIGH-Level gezogen werden, falls der Button nicht gedrückt ist. Ansonsten hätte der Digital-Eingang einen nicht definierten Zustand.
Verändere also den Arduino-Sketch und teste ihn.

Ausblick: Später verwenden wir das Modul LED&KEY und können mit 8 Tastern zumindest eine Oktave abdecken. Wir kommen also auf das Keyboard zurück. Alternativ kannst du auch jetzt schon mit 8 Tastern arbeiten.