Info: Was sind **analoge Eingänge** und wie werden diese ausgewertet?
Viele Sensoren geben nicht `0` und `1` (also 0V oder 5V) aus, sondern auch Zwischenwerte. Dazu verfügt unser Arduino über sechs Stück **10 Bit Analogwandler** (*kurz:* A/D-Wandler) `A0` bis `A5`. Dies bedeutet, dass eine Eingangsspannung zwischen 0V und 5V auf die (binären) Werte 0.....1023 umgesetzt wird. Diesen Wert können wir z.B. mit `analogRead(A0);` abrufen (auslesen). Ist der Wert z.B. 512, so liegt an `A0` eine Spannung von 2,5V an.

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.

b) Aktiviere nun den **Seriellen Plotter (SP)**. Drehe am Poti und beobachte die Kurve.
Der SP wird über „Werkzeuge“ (Tools) aufgerufen.

Info: Was ist ein **Plotter** ? [engl. *to plot* = zeichnen]
Wenn sich z.B. eine Messgröße (Lichtstärke) schnell ändert, gibt eine digitale Darstellung nur ein Zahlen-Gezappel. Der Plotter kann den zeitlichen Verlauf (z.B. der Lichtstärke) als Kurve in einem Koordinatensystem aufzeichnen. Das geschieht hier auf dem Display des Monitors

2. Spannungs-Messgerät programmieren

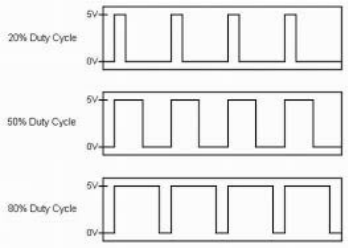
a) Öffne den Sketch **ReadAnalogVoltage** und teste ihn! Vorher den Arduino löschen! Die Schaltung ist mit der von Aufg. 1) identisch. Delay einfügen!
Dieser Sketch bewirkt im Prinzip, dass der Arduino wie ein digitales Spannungs-Messgerät arbeitet. Auch hier lässt sich der **SP** aktivieren.

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, PWM

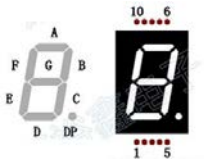
Nun wollen wir die Helligkeit einer LED mit einem Poti einstellen und dabei mehr über analoge Eingänge und **PWM** (**P**uls-**W**eiten-**M**odulation) lernen. Öffne dazu den Sketch **AnalogInOutSerial** und erhöhe das Delay auf 100ms. Stecke die Elektronik auf dem zuvor gelöschten Arduino zusammen (testen 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** (s. später) benötigt man hier nicht. Ersetze diese durch eine simple Division.
Ein Ausgang von 255 entspricht 5V, die LED leuchtet mit voller Helligkeit. 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 des Arduino an. Die gemeinsame Anode wird mit +5V verbunden.

Info: Was ist ein **Array**? [engl. *array* = Feld, Ansammlung]
Ein Array ist ein Datentyp, mit welchem man beliebig viele Werte abspeichern kann. Während eine **Variable** immer ausschließlich einen einzigen Datentyp enthält, kann eine Arrayvariable eine große Anzahl verschiedene Werte enthalten. Ganz grob:
Das Verhältnis zwischen einer Variablen und einem Array entspricht in etwa dem Verhältnis zwischen einem Fahrrad und einem Bus.

```
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!!