

## Ein Arduino-Voltmeter, das Spannungen bis 30V messen kann

In diesem Beispiel benutzen wir eine einfache Schaltung wie in der Abbildung bei der Erklärung. Die Widerstände ( $R_1 = 100$  und  $R_2 = 10$  Kilohm) reduzieren die zu messende Spannung auf ein Elftel des ursprünglichen Werts.

Wenn an den Eingangspins des Arduino nicht mehr als 5V ankommen dürfen, ergibt sich so eine theoretisch mögliche Eingangsspannung von ( $11 * 5 = 55$ ) 55V

### Erklärung

Zur Berechnung der Teilspannung  $U_2$  über  $R_2$  muss zunächst der Gesamtwiderstand berechnet werden

$$R_{ges} = R_1 + R_2$$

Wenn die Gesamtspannung und die Widerstände bekannt sind, lässt sich nach dem Ohmschen Gesetz der Strom ( $I$ ) bestimmen:

$$I = U/R_{ges} = U/R_1 + R_2$$

Nach den Regeln der Reihenschaltung ist der Strom durch alle Bauteile gleich, somit lässt sich das gesuchte  $U_2$  bestimmen

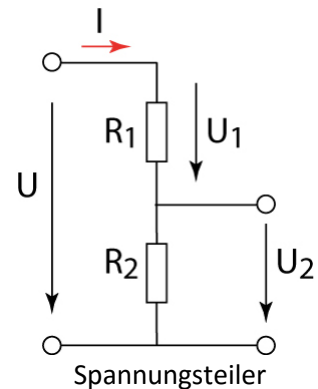
$$U_2 = I * R_2$$

Setzt man die Formel für den gemeinsamen Strom ( $I = U/R_{ges}$ ) ein, ergibt sich die Ausgangsspannung (in Abhängigkeit von den Widerständen und der Eingangsspannung)

$$U_2 = U/R_{ges} * R_2$$

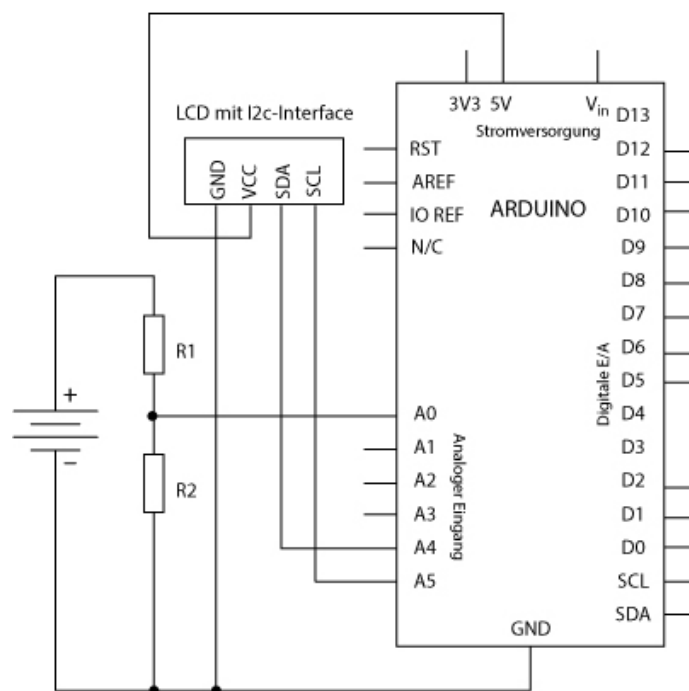
Setzt man die oben angegebenen Werte ein, ergibt sich

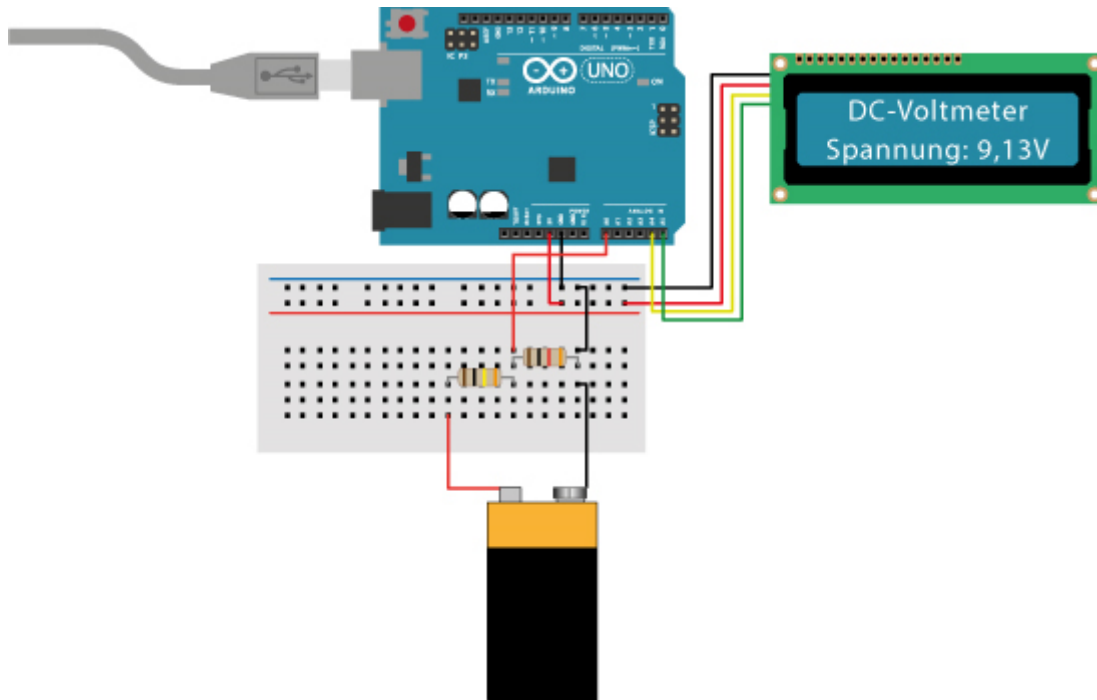
$$U_2 = 55/110 * 10 = 5$$



Weil bei realen Widerständen die angegebenen Werte nur näherungsweise zutreffen, empfiehlt es sich, einen Sicherheitsbereich zu den errechneten Grenzwerten einzuhalten und die Eingangsspannung auf 30V zu begrenzen.

### Schaltskizze





### Programmcode

(Beim Kopieren bitte so verfahren, wie im Kapitel [LCD-Display](#) erklärt)

// Stromteiler-Gleichstrom-Voltmeter 0 - 30V

```
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
int analogInput = 0;
```

```
float vout = 0.0;
```

```
float vin = 0.0;
```

```
float R1 = 100000.0; // Widerstandswert R1 (100K) - siehe Schaltskizze!
```

```
float R2 = 10000.0; // Widerstandswert R2 (10K) - siehe Schaltskizze!
```

```
int values = 0;
```

```
void setup() {
```

```
  Serial.begin(9600); //Seriellen Monitor einbinden
```

```
  pinMode(analogInput, INPUT);
```

```
  lcd.begin();
```

```
  lcd.backlight();
```

```
  lcd.setCursor(2, 0);
```

```
  lcd.print("DC-Voltmeter");
```

```
}
```

```
void loop() {
```

```
  // Werte am analogen Pin lesen
```

```
  values = analogRead(analogInput); //Messwerte am analogen Pin als "values" definieren
```

```
  vout = (values * 5.0) / 1024.0; // Messwerte in Volt umrechnen = Spannung am Ausgang des
```

```
    // Spannungsteilers mit Zuleitung zu Pin A0
```

```
  vin = vout / (R2 / (R1 + R2)); // Berechnen, welche Spannung am Eingang des Spannungsteilers
```

```
// anliegt. Das entspricht der Spannung der zu untersuchenden
```

```
Batterie
```

```
if (vin < 0.09) {  
  vin = 0.0; // Unterdrücken unerwünschter Anzeigen  
}  
lcd.setCursor(0, 1);  
lcd.print("Spannung: ");  
lcd.print(vin);  
lcd.print("V");  
delay(500);  
  
Serial.print("V= ");  
Serial.println(values); // Messwerte am analogen Pin im seriellen Monitor anzeigen  
}
```