

Grundlagen Elektrotechnik für den Maschinenbau

- Praktikum 1 –

- Arduino & Grundlagen –

Vorwort

Im Rahmen der Veranstaltung ‚Grundlagen Elektrotechnik für den Maschinenbau‘ werden die technologischen und methodischen Grundlagen der Elektrotechnik vermittelt. Im Fokus stehen dabei Inhalte, die insbesondere im Kontext des Maschinenbau-Studiums relevant sind. Bestandteil dieser Veranstaltung ist das ‚Praktikum Elektrotechnik‘. Hier werden in Form von experimentellen Aufbauten und Versuchen die während der Vorlesung vermittelten theoretischen Kenntnisse in die Praxis überführt und angewendet.

Um die Praktika zu realisieren, wird Ihnen ein programmierbarer Mikrocontroller, ein sogenannter *Arduino*, zur Verfügung gestellt. Dieser Controller bietet zahlreiche Möglichkeiten und bildet die Grundlage für alle folgenden Versuche. Im ersten Abschnitt des Praktikums soll Ihnen der Arduino als auch die zugehörige Entwicklungsumgebung näher gebracht werden.

Arduino – Einführung

Beim Arduino handelt es sich um eine Open-Source-Elektronikplattform, die aus Hardware und Software besteht und oft in der Erstellung von Elektronikprojekten Verwendung findet. Im Rahmen dieses Praktikums steht als Hardware der *Arduino Uno* zur Verfügung, auf dessen Funktionen später näher eingegangen werden soll. Als Software kann auf die eigens entwickelte Entwicklungsumgebung, die sogenannte *Arduino IDE* (Integrated Development Environment) zurückgegriffen werden.



Abb.: 1 Arduino Uno

Installation der IDE

Die Entwicklungsumgebung von Arduino ist auf der Firmenwebseite (<https://www.arduino.cc>) für alle gängigen Betriebssysteme verfügbar. Nach dem Starten der Anwendung erscheint die Hauptansicht der Entwicklungsumgebung:

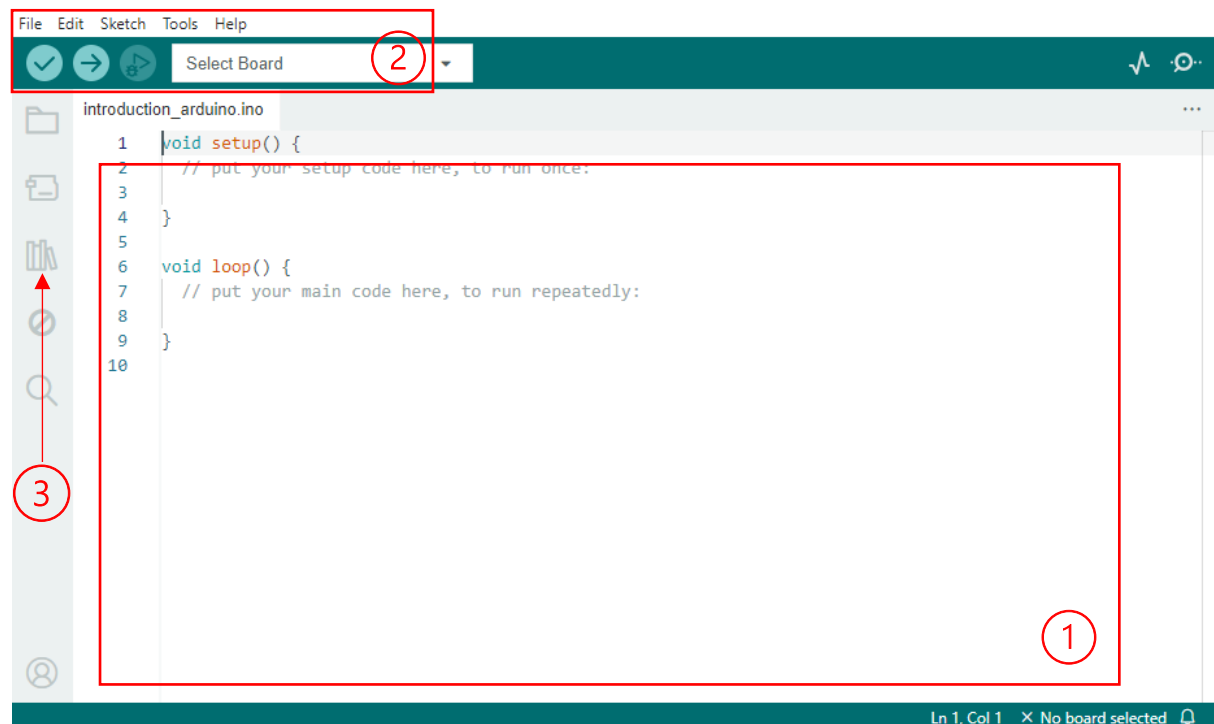


Abb.: 2 Arduino IDE - Startansicht

- 1) **Editorbereich:** hier kann der Quellcode des eigenen Programms implementiert werden
- 2) **Menüleiste:** Zahlreiche Funktion, wie das Speichern und Laden eines Skriptes oder das Hochladen eines Programmes auf den Arduino finden sich hier
- 3) **Bibliotheken-Manager:** Wird für eine Anwendung eine externe Bibliothek benötigt, kann sie hier installiert werden. Auch die Verwaltung bereits installierter Bibliotheken ist hier möglich

Erstes Programm laden und ausführen

Grundsätzlich teilen sich Programme, die auf einen Arduino Mikrocontroller geladen werden, in zwei Teile (*void setup* und *void loop*) auf. Diese Teile sind beim Öffnen eines neuen Skriptes bereits im Editorbereich angelegt.

void setup – dieser Teil des Programmes wird beim Starten des Controllers einmalig durchlaufen. Er eignet sich daher dafür grundlegende Zuweisungen, wie zum Beispiel Pin-Belegungen oder elementare Variablen zu definieren.

void loop – dieser Teil des Programms wird während der Aktivität des Controllers in stetiger Dauerschleife von oben nach unten abgearbeitet. Hier wird die eigentliche Funktionalität des Skriptes implementiert.

Die Inhalte der jeweiligen Bereiche werden in die geschweiften Klammern geschrieben.

Um einen ersten Eindruck von einem funktionierenden Programm zu erhalten, laden wir zunächst das Beispiel-Skript *Blink*, welches uns ermöglicht die werksseitig verbaute LED auf dem Arduino Uno anzusteuern. Über die Menüleiste (*File* → *Examples* → *01.Basics* → *Blink*) kann das Skript direkt in den Editorbereich geladen werden.

Um das Skript nun auf den Mikrocontroller zu laden, muss dieser zunächst mithilfe des Datenübertragungskabels an den Computer angeschlossen werden. Hier ist zu beachten, dass auf Seite des Arduinos die Micro-USB-Schnittstelle verwendet wird. Steht die Verbindung sollte im oberen Bereich der Entwicklungsumgebung der *Arduino Uno* ausgewählt werden können. Danach wird das Programm über die Schaltfunktion *Hochladen (Menüleiste)* auf den Arduino übertragen. Nach einer kurzen Initialisierungsphase sollte das Skript ordnungsgemäß laufen und die LED auf dem Arduino anfangen periodisch zu blinken.

Arduino als Steuereinheit

Eine wesentliche Funktionalität eines Mikrocontrollers besteht darin, über seine verschiedenen Ausgänge/Ports elektronische Bauelemente bzw. Schaltungen anzusteuern. Dazu setzt der Arduino entsprechend dem geladenen Skript definierte Ausgänge auf High (Spannungspotential anliegend) oder Low (spannungsfrei). Die wesentlichen Befehle, die hierfür benötigt werden, sind:

pinMode(port, function) – dieser Befehl wird standardmäßig innerhalb der void setup verwendet und definiert die Funktion (state) eines Ausgangs (port). Der Ausgang wird durch eine Zahl, die dem Controller zu entnehmen ist, repräsentiert, während es für die Funktion die Wahl zwischen output und input gibt.

digitalWrite(port, state) – dieser Befehl wird standardmäßig innerhalb des void setup verwendet und definiert den state eines Ausgangs. Man hat hier die Auswahl zwischen High und Low.

- a) **Basisversuch:** Um sich mit der grundlegenden Funktionalität des Arduino als Steuereinheit vertraut zu machen, wird ein Vorgehen skizziert, um eine externe LED im Intervall blinken zu lassen. Dazu wird folgender Schaltplan und Quellcode gegeben:

```
ard_schalteneinheit_basis.ino
1  // Benennung der LED-Pin
2  const int ledPin = 5;
3
4  void setup() {
5      // Den Pin 5 als Ausgang(output) festlegen
6      pinMode(ledPin, OUTPUT);
7  }
8
9  void loop() {
10     // LED einschalten
11     digitalWrite(ledPin, HIGH);
12     // 1 Sekunde warten
13     delay(1000);
14
15     // LED ausschalten
16     digitalWrite(ledPin, LOW);
17     // 1 Sekunde warten
18     delay(1000);
19 }
20
```

Abb.: 3 Quellcode - Basisversuch 1 - blinkende LED

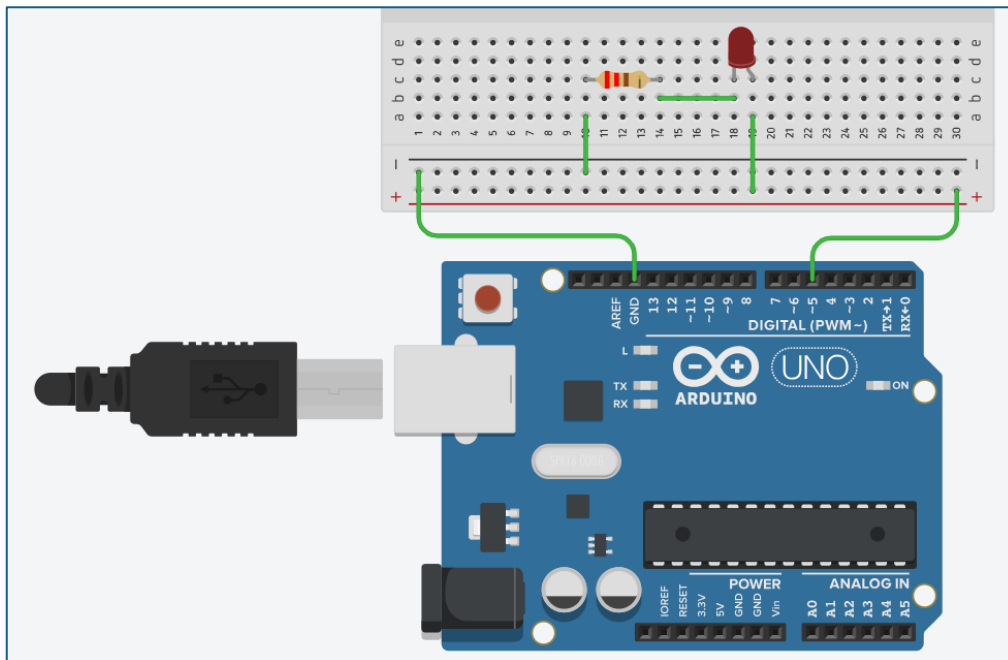


Abb. 4: Schaltplan – Ansteuern einer LED

Beachten Sie hierbei, dass beim Betrieb einer LED ein Vorwiderstand benötigt wird, um den durchfließenden Strom zu begrenzen. Für die Auswahl eines geeigneten Widerstandes wird das Ohmsche Gesetz verwendet. Der ermittelte Widerstand kann durch eine standardisierte Farbcodetabelle gefunden werden.

4 Ringe

1.000 Ω $\pm 5\%$

Farbe	1. Ring	2. Ring	3. Ring	Multiplikator	Toleranz
Schwarz	0	0	0	$\times 1 \Omega$	
Braun	1	1	1	$\times 10 \Omega$	$\pm 1\%$
Rot	2	2	2	$\times 100 \Omega$	$\pm 2\%$
Orange	3	3	3	$\times 1.000 \Omega$ (1 k Ω)	
Gelb	4	4	4	$\times 10.000 \Omega$ (10 k Ω)	
Grün	5	5	5	$\times 100.000 \Omega$ (100 k Ω)	$\pm 0,5\%$
Blau	6	6	6	$\times 1.000.000 \Omega$ (1 M Ω)	$\pm 0,25\%$
Lila	7	7	7	$\times 10.000.000 \Omega$ (10 M Ω)	$\pm 0,1\%$
Grau	8	8	8		$\pm 0,05\%$
Weiß	9	9	9		
Gold				$\times 0,1 \Omega$	$\pm 5\%$
Silber				$\times 0,01 \Omega$	$\pm 10\%$

5 Ringe

2.700 Ω $\pm 1\%$

Abb. 5: Farbcodetabelle (<https://www.elektronik-kompendium.de>)

- b) **Anwendung 1.1:** Bauen Sie mit den aus dem vorherigen Experiment gewonnenen Kenntnissen eine „LED-Lichtleiste“ auf. Diese Leiste soll aus 5 LEDs bestehen und verschiedene Modi abspielen können. Nutzen Sie für die Umsetzung die Schaltung und den Quellcode des Basisversuches und erweitern Sie beides entsprechend. Die Lichtmuster der verschiedenen Modi sind der folgenden Abbildung zu entnehmen:

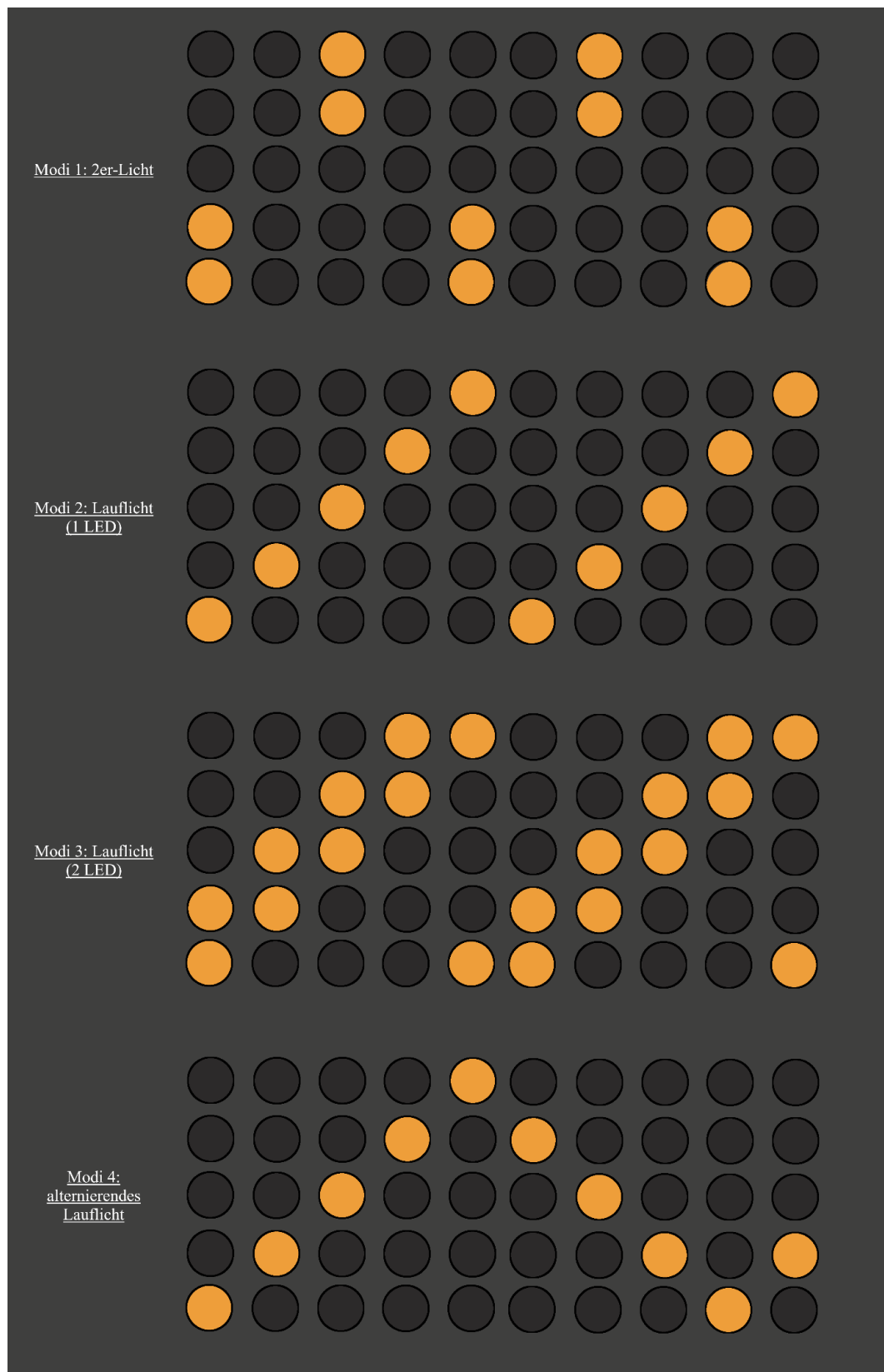


Abb. 6: Modi der Lichtleiste

Arduino als Kommunikationsmittel

Eine weitere Eigenschaft, die der Arduino besitzt, ist seine Kommunikationsfähigkeit über seine seriellen Schnittstellen. Über die Menüleiste in der Entwicklungsumgebung (*Tools* → *serieller Monitor*) lässt sich eine Schnittstelle, der sogenannte serielle Monitor öffnen. Ist der Arduino während der Aktivität mit dem PC verbunden, lassen sich hier alle möglichen Daten anzeigen lassen. Um dies zu realisieren, werden im Skript folgende Befehle verwendet:

serial.begin(9600) – Durch diesen Befehl im Abschnitt des void setup wird der serielle Monitor initialisiert.

serial.print() – Dieser Befehl ermöglicht es einen string-Datentyp zeilenmäßig über den seriellen Monitor anzuzeigen.

- a) **Basisversuch:** Durch Übertragen des folgenden Quellcodes können Sie die Funktionalität des seriellen Monitors überprüfen.

```
ard_sendeeinheit_basis.ino
1 void setup() {
2   // Initialisiere die serielle Kommunikation mit einer Baudrate von 9600
3   Serial.begin(9600);
4 }
5
6 void loop() {
7   // Sende "Hello World" über die serielle Schnittstelle
8   Serial.println("Hello World");
9 }
10
```

Abb. 7: Quellcode - Basisversuch 2 - serieller Monitor

- b) **Anwendungsversuch 1.2:** Erweitern Sie Ihre Lichtleiste aus dem vorherigen Abschnitt derart, dass die jeder Durchlauf im seriellen Monitor durch Plotten des Modus-Namen angezeigt wird.

Arduino als Leseinheit

Die letzte Funktionalität, die in dieser ersten Einleitung gezeigt werden soll, ist die Verwendung des Arduino als Leseinheit. Über seine digitalen Ports ist der Arduino in der Lage zu messen, ob auf diesem Eingang eine Spannung (High) oder keine Spannung (Low) anliegt. Im Zusammenspiel mit elektronischen Bauelementen, wie zum Beispiel einem Schalter ist man somit in der Lage, auf Veränderungen der Schaltung zu reagieren. Nachfolgende Befehle sind für diese Funktionalität essentiell:

pinMode(port, INPUT_PULLUP) – Der Befehl optimiert den angegebenen Eingang für die Belegung mit einem Taster, welcher nun gegen Low verschaltet werden kann

digitalRead(port) – gibt den aktuellen Status (High/Low) eines digitalen Einganges zurück

- a) **Basisversuch:** Mit folgendem Versuchsaufbau lässt sich die Vorgehensweise, um Signale auszulesen, nachvollziehen:

ard_leseeinheit_basis.ino

```
1 void setup() {  
2   // optimiert den Eingang 2 für die Verwendung eines Tasters  
3   pinMode(2, INPUT_PULLUP);  
4 }  
5  
6 void loop() {  
7   // speichern des Tasterzustandes in der Variable buttonState  
8   int buttonState = digitalRead(2);  
9 }  
10
```

Abb. 8: Quellcode - Basisversuch 3 - Taster

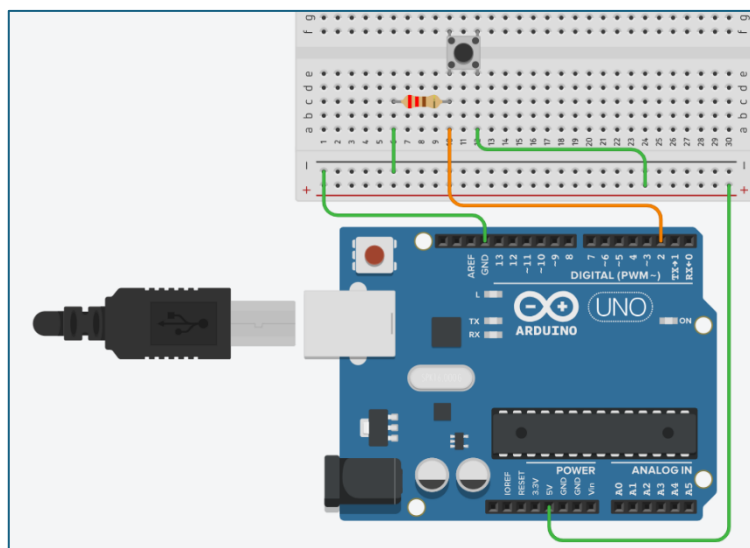


Abb. 9: Schaltplan - Tasteransteuerung

- b) **Anwendung 1.3:** Erweitern Sie Ihre Lichtleiste derart, dass erst ein Betätigen des Schalters einen Licht-Modus auslöst.
(optional: Machen Sie sich Gedanken und recherchieren Sie, wie man mithilfe des Tasters ein Wechseln der verschiedenen Modi im laufenden Betrieb realisieren kann. *Stichworte: switch/case-Funktion, millis()*)

Grundlagen der Elektrotechnik

Nachdem Sie sich nun mit den grundlegenden Funktionen und Möglichkeiten einer Arduino-Entwicklungsumgebung vertraut gemacht haben, soll im anschließenden Praktikumteil der Fokus auf essentiellen Grundlagen der Elektrotechnik liegen. Dazu werden Themen und Zusammenhänge, die sie in der Vorlesung gelernt haben an praktischen Versuchsaufbauten veranschaulicht und das zugehörige Wissen vertieft.

Ohm'sches Gesetz

Eines der elementarsten Gesetze der Elektrotechnik ist das „Ohm'sche Gesetz“. Es verknüpft die Größen Stromstärke, Spannung und Widerstand und kann so über die wirkenden Kräfte in einer Schaltung Auskunft geben. Die Größen stehen in folgender Verbindung zueinander:

$$R = \frac{U}{I}$$

Anhand dieser Gleichung sieht man, dass man unter Kenntnis zweier der Größen die dritte berechnen kann. Wir möchten diese Verhältnismäßigkeit experimentell nachweisen.

- a) **Aufgabe 2.1:** Bauen Sie entsprechend des untenstehenden Schaltplans das Experiment auf und messen Sie für unterschiedliche Widerstände (220Ω, 2000Ω, 5000Ω, 10000Ω) jeweils die anliegende *Spannung* über dem Widerstand als auch die *Stromstärke*. Überprüfen Sie die gemessenen Werte mit der theoretischen Gleichung des Ohm'schen Gesetzes. Beachten Sie bei der Messung mit dem Multimeter die Bedienungshinweise (Seite X), um Schäden zu vermeiden!

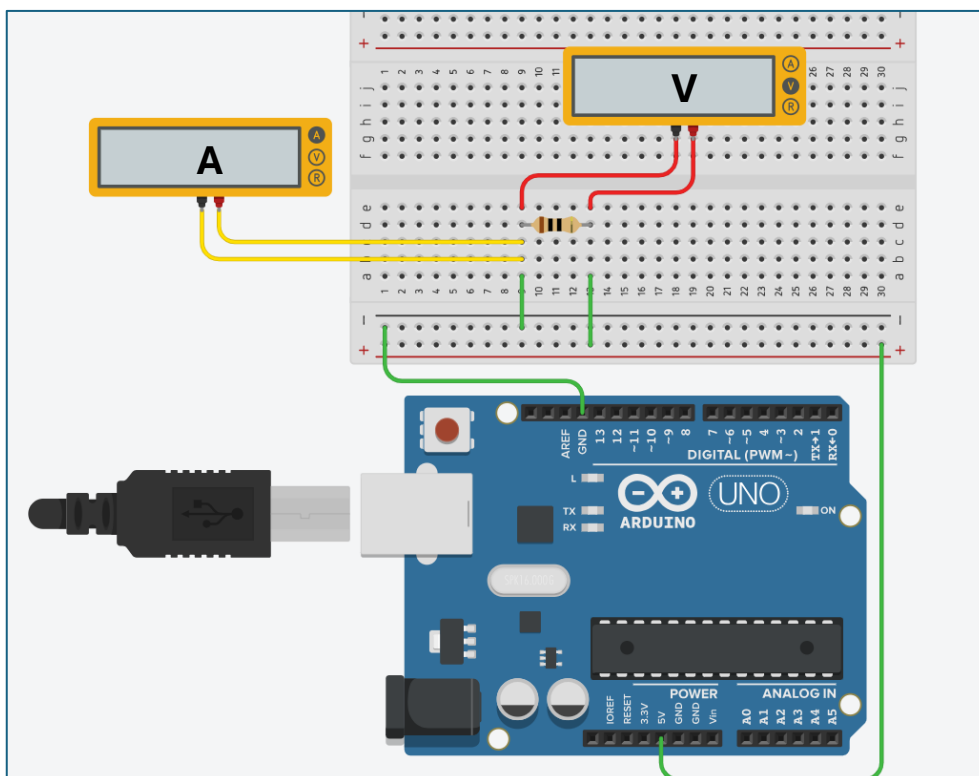


Abb. 10: Schaltplan – Ohmsches Gesetz

Wir möchten nun die gewonnenen Erkenntnisse nutzen, um ein Bauteil für die kommenden Experimente zu präparieren. Es handelt sich hierbei um ein sogenanntes Potentiometer.



Aufgabe dieses Bauteils ist es einen händisch variablen Widerstand zu realisieren. In den meisten Fällen wird hierfür ein langer Widerstandsdraht als Leiter verwendet. Die Drahtlänge korreliert mit einem bestimmten Widerstand. Greift man die Leitung an einem früheren Punkt ab, resultiert ein geringerer Widerstand. Über das Drehen des Knopfes kann der Abgriffspunkt und somit der Widerstand des Bauteils eingestellt werden.

- b) **Aufgabe 2.2:** Messen Sie die Stromstärke über verschiedene Einstellungen des Potentiometers. Berechnen Sie die jeweiligen Widerstandswerte des Potentiometers und erstellen Sie eine sinnvoll abgetastete Skala für den Widerstandsbereich des Potentiometers.

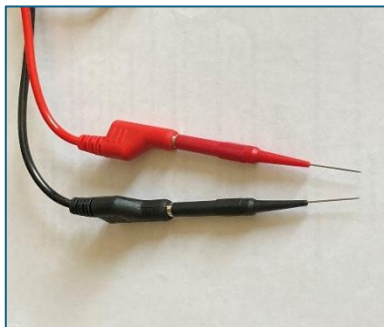
Bedienungshinweise Multimeter

Für die Bearbeitung des gesamten Kofferpraktikums Elektrotechnik steht Ihnen ein Multimeter (XX) zur Messung elektrischer Größen zur Verfügung. Um fehlerhafte Messungen und technische Beschädigungen zu vermeiden, müssen ein paar grundlegende Hinweise beachtet werden:

- 1) **Auswahl der Messgröße:** Das Multimeter ist in der Lage Stromstärke, Spannung und Widerstände zu messen. Bei der Spannungsmessung kann man sich zudem zwischen der Messung im Wechsel- und Gleichstrom entscheiden. Um eine der Größen zu messen, muss der große Drehknopf in den entsprechenden Sektor zeigen. Die Bezeichnung richtet sich hier nach der Einheit der zu messenden Größe:



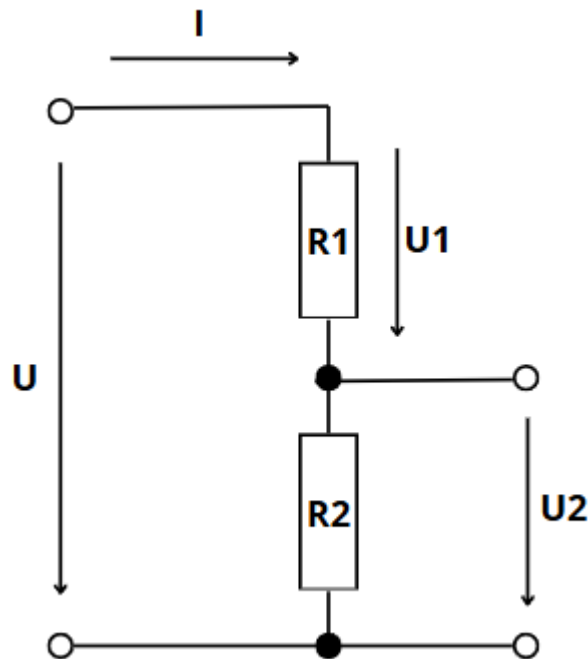
- 2) **Auswahl des Messwertbereiches:** Um eine möglichst exakte Messung zu gewährleisten, verfügt das Multimeter über unterschiedliche Messwertbereiche. Bei einer Messung ist darauf zu achten anfangs mit dem größtmöglichen Messwertbereich zu starten, um ein Durchbrennen der Sicherungen zu verhindern. Schrittweise kann nun bei ausreichender Reserve des Messwertes zum nächst kleineren Messwertbereich, dieser herabgesetzt werden.
- 3) **Auswahl der Messspitzen:** Die originalen Messspitzen des Multimeters sind aufgrund ihres hohen Durchmessers nicht in der Lage in die Struktur des Breadboards einzudringen. Um trotzdem Messungen an unseren Experimenten durchzuführen, liegen kleinere Messspitzen bereit, die Sie verwenden können. Seien Sie Umgang mit diesen feinen Spitzen vorsichtig, da Verletzungsgefahr besteht!



Spannungsteiler

Mehrere Widerstände können verwendet werden, um die Spannung in einem Stromkreis zu teilen. Dies kann verwendet werden, wenn ein bestimmter Teil der Schaltung eine niedrigere Spannung als die der Stromversorgung benötigt. Die Formel zur Berechnung der Spannung am Ausgang des Spannungsteilers ergibt sich aus dem Ohmschen Gesetz:

$$U_2 = \frac{R_2}{R_1 + R_2} \cdot U$$



Aufgabe 3.1: Entwerfen Sie unter Verwendung der verfügbaren Elemente einen Spannungsteiler, der die Eingangsspannung durch 2 teilt. Bauen Sie die Schaltung zusammen und prüfen Sie mit einem Multimeter experimentell die Funktion des Spannungsteilers.

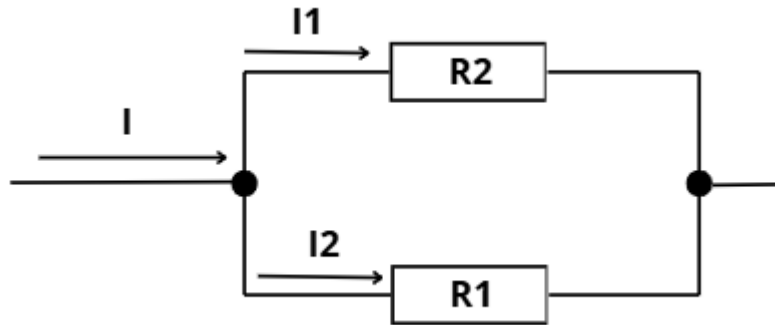
Aufgabe 3.2: In einem Stromkreis mit einer Quellenspannung $U = 48 \text{ V}$, sind fünf Widerstände angeschlossen: $R_1 = 2 \text{ k}\Omega$, $R_2 = 3 \text{ k}\Omega$, $R_3 = 4 \text{ k}\Omega$, $R_4 = 6 \text{ k}\Omega$, $R_5 = 12 \text{ k}\Omega$. Widerstände R_1 und R_2 sind in Reihe geschaltet. Die Widerstände R_3 und R_4 sind ebenfalls in Reihe geschaltet. Die beiden resultierenden Zweige sind parallel zueinander geschaltet. Der Widerstand R_5 wird in Reihe zu dieser gesamten Parallelschaltung geschaltet. Bestimmen Sie den Gesamtwiderstand der gesamten Schaltung. Ermitteln Sie die Spannung an den einzelnen Widerständen. Wie wird die Spannung an R_5 , wenn R_3 auf $10 \text{ k}\Omega$ erhöht werden sollte?

Stromteiler

Stromteiler funktionieren auf die gleiche Weise wie ein Spannungsteiler, nur dass die Widerstände jetzt parallel statt in Reihe geschaltet sind.

$$\frac{I_1}{I_2} = \frac{R_2}{R_1}$$

$$I = I_1 + I_2$$



Aufgabe 4.1: Entwerfen Sie unter Verwendung der verfügbaren Elemente einen Stromteiler, der den Eingangsstrom durch 2 teilt. Bauen Sie die Schaltung zusammen und prüfen Sie mit einem Multimeter experimentell die Funktion des Stromteilers.

Aufgabe 4.2: In einem Stromkreis mit einem Quellenstrom $I = 10 \text{ mA}$, sind drei Widerstände angeschlossen: $R_1 = 2 \text{ k}\Omega$, $R_2 = 3 \text{ k}\Omega$, $R_3 = 6 \text{ k}\Omega$. Widerstände R_1 , R_2 und R_3 sind parallel geschaltet. Der Widerstand $R_4 = 1 \text{ k}\Omega$ wird in Reihe zu dieser Schaltung geschaltet. Bestimmen Sie den Gesamtwiderstand der gesamten Schaltung. Bestimmen Sie den Strom, der durch jeden der Widerstände in Parallelschaltung fließt.

Motorversuch

In diesem Versuch lernen Sie, wie die Motordrehzahl durch ein Potentiometer gesteuert werden kann. Dazu wird ein einfaches Schaltungskonzept genutzt, bei dem das Potentiometer als einstellbarer Widerstand fungiert und somit die Spannung und den Strom des Motors beeinflusst.

Ein Potentiometer ist ein einstellbarer Widerstand, der als Spannungsteiler verwendet wird. Es hat drei Anschlüsse: zwei äußere Anschlüsse, die den Gesamtwiderstand des Potentiometers darstellen, und einen mittleren Anschluss (Schleifer), der eine Spannung proportional zur eingestellten Position des Potentiometers ausgibt.

Die Ausgangsspannung U_{out} am Schleifer kann mit der folgenden Formel berechnet werden:

$$U_{out} = \frac{R_2}{R_1 + R_2} \cdot U$$

wobei:

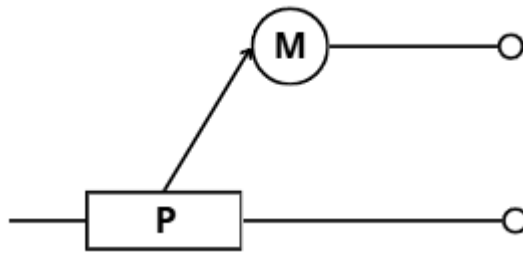
- die Eingangsspannung U (hier 12V) ist,
- R_1 und R_2 die beiden Widerstandswerte des Potentiometers sind.

Durch Verändern der Position des Schleifers wird R_1 bzw. R_2 verändert, wodurch sich die Ausgangsspannung ändert. Diese veränderte Spannung steuert die Drehzahl des Motors.

Benötigte Bauteile:

- Netzteil: 12V, 1A
- Gleichstrommotor: 12V
- Potentiometer: 1 kOhm
- Verbindungskabel
- Multimeter zur Spannungsmessung
- Skala zur Messung der Drehzahl

Aufgabe 5.1: Bauen Sie aus den vorhandenen Elementen eine Schaltung auf, mit der Sie den Strom für den Motor über ein Potentiometer steuern können.



Aufgabe 5.2: Messen Sie die Motordrehzahl (RPM) bei 0, 250, 500, 750 und 1000 Ohm mit Hilfe der Skala für das Potentiometer, welche Sie im vorherigen Experiment hergestellt haben, der weißen Scheibe am Motor und einer Stoppuhr. Wiederholen Sie jede Messung dreimal und ermitteln Sie den Durchschnittswert, um Messunsicherheiten zu minimisieren.

Aufgabe 5.3: Zeichnen Sie das Diagramm der Motordrehzahl in Abhängigkeit von der Spannung auf.

Wheatstone-Brücke

In der Elektrotechnik gibt es bestimmte Schaltgruppen, die in Ihrer Grundform immer wiederkehren und daher von besonderem Interesse sind. Durch kleine Variationen lassen sich diese Grundsteine auf zahlreiche Anwendungen anpassen. Es lohnt sich also sich die Grundformen genauer anzuschauen und deren Funktion zu verstehen. Eine dieser Baugruppen ist die sogenannte Wheatstone-Brücke, benannt nach dem britischen Physiker Charles Wheatstone (*1802, †1875). Der grundlegende Aufbau einer solchen Brücke ist in der nachfolgenden Abbildungen zu erkennen:

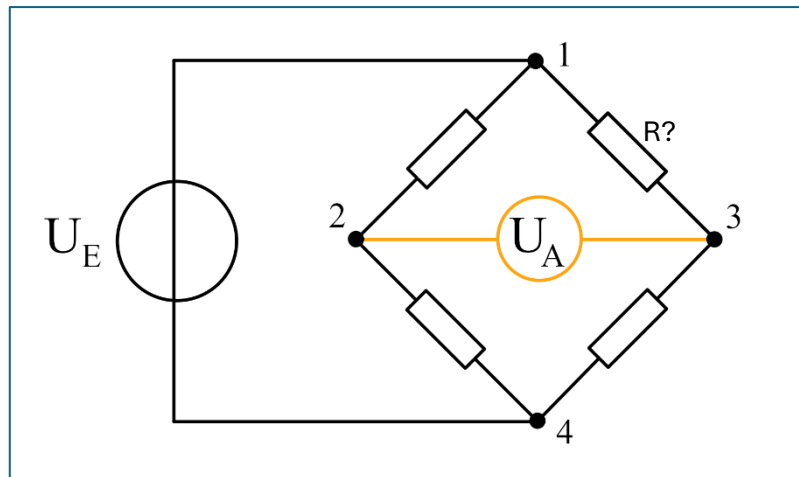


Abb. X: Schaltplan einer Wheatstone-Brücke

In der Regel werden 4 Widerstände auf die oben gezeigte Art und Weise miteinander verschaltet. Über die Punkte 1 und 4 wird die Schaltung mit der Eingangsspannung versorgt. Zwischen den Punkten 2 und 3 kann eine Ausgangsspannung gemessen werden. Durch diese ‚Brückenverbindung‘ erlangt die Wheatstone’sche Schaltung auch ihren Namen.

- a) **Aufgabe X.1:** Bauen Sie eine Wheatstone-Brücke (mithilfe des gegebenen Schaltplans). Für den Widerstand $R?$ setzen Sie nacheinander Widerstände der Größe 1000Ω , 2000Ω , 5000Ω und 10.000Ω ein. Messen Sie die Ausgangsspannung über den Punkten 2 und 3 und dokumentieren Sie Ihre Ergebnisse.

Versuchen Sie nun mit Ihrem Wissen zu Strom- und Spannungsteilern einen Zusammenhang zwischen der Ausgangsspannung und dem variierenden Widerstand $R?$ herzuleiten und dokumentieren Sie nachvollziehbar Ihre Rechenschritte. Die drei übrigen Widerstände können Sie als gegeben betrachten.

Sie haben festgestellt, dass die Ausgangsspannung einer Wheatstone-Brücke abhängig von den Widerstandsverhältnissen seiner Bestandteile ist. Die Frage bleibt jedoch, wie man sich diesen Umstand zu Nutzen machen kann. Eine Möglichkeit ist die Nutzung einer sogenannten ‚abgeglichenen‘ Wheatstone-Brücke. Hierbei ist das Ziel einen unbekannten Widerstand zu quantifizieren. Um dies zu erreichen wird zusätzlich zu dem unbekannten Widerstand ein variabler Widerstand der Wheatstone-Brücke hinzugefügt. Durch Variation dieses Widerstandes kann man idealerweise eine Ausgangsspannung von 0V erreichen. Ist dies der Fall, spricht man von einer abgeglichenen Brücke und es lässt sich nun unter Kenntnis von drei der vier verschalteten Widerstände der unbekannte Widerstand berechnen.

- b) **Aufgabe X.2:** Bauen Sie einen geeigneten Versuch auf. Nutzen Sie das ‚skalierte‘ Potentiometer, um die Widerstandswerte der drei unbekannten Bauteile (grün, rot und blau) zu bestimmen. Dokumentieren Sie ebenfalls Ihre durchgeführten Berechnungen in einer nachvollziehbaren Art und Weise.

Im nächsten Schritt wollen wir die Funktionalität einer Wheatstone-Brücke verwenden, um einen praktischen Nutzen daraus zu ziehen und eine Temperaturmessung durchführen. Dazu verwenden wir einen sogenannten NTC-Temperatursensor. NTC steht im Englischen für ‚negative temperature coefficient‘. Das bedeutet, dass der Sensor aus einem Material gefertigt ist, welches bei steigender Temperatur einen sinkenden Widerstandswert aufweist. Der Zusammenhang zwischen Umgebungstemperatur und Widerstandswert kann dem Datenblatt des Sensors entnommen werden. Für die meisten Anwendungsbereiche zeichnet sich ein standardmäßiger PT100-Sensor (Material: Platin, Widerstand bei 20Grad: 100 Ohm) durch nahezu lineares Verhalten aus.

- c) **Aufgabe X.3:** Integrieren Sie einen PT100-Sensor in Ihre Wheatstone-Brücke und bestimmen Sie unter Kenntnis der Abgleichbedingung die aktuelle Raumtemperatur. Dokumentieren Sie alle dazu notwendigen Messwerte und Berechnungen sorgfältig. *Hinweis: Achten Sie auf konstante Umgebungsbedingungen, um Messabweichungen zu vermeiden.*