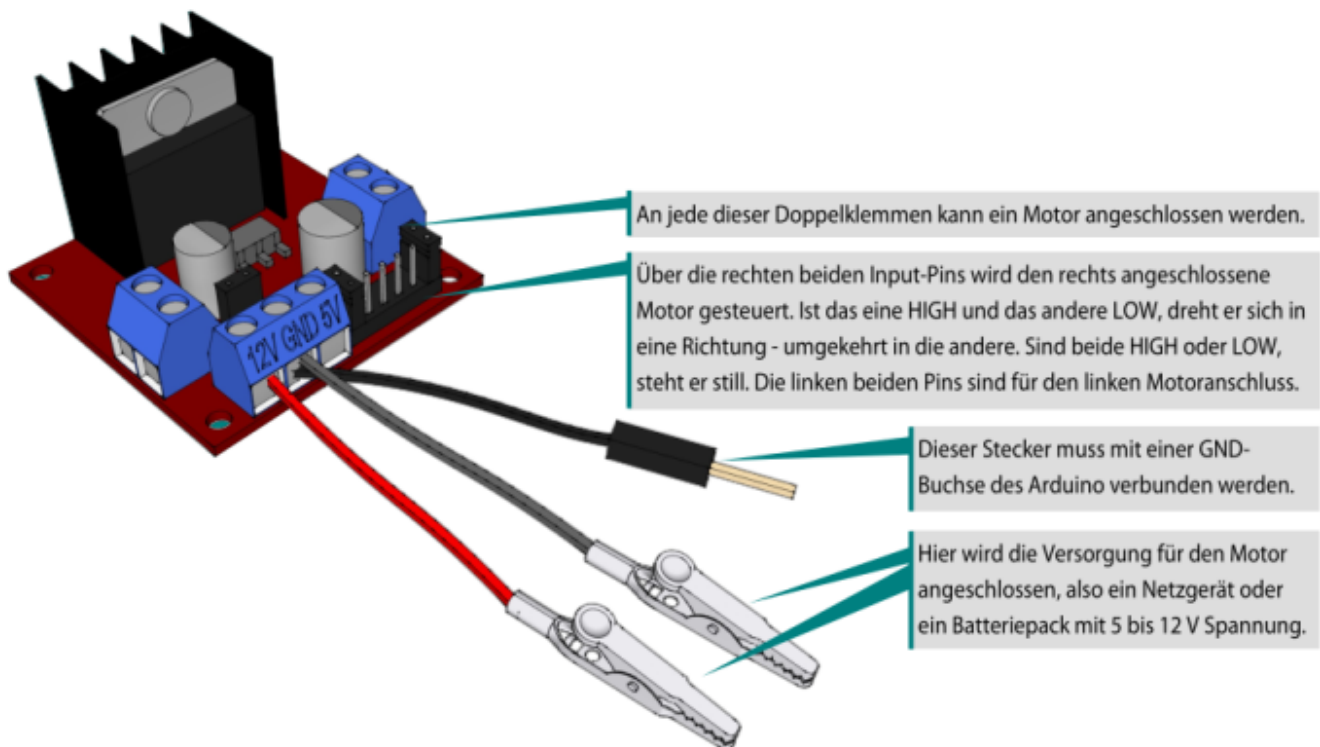


Gleichstrommotoren steuern

Um einen Motor zu betreiben, reicht die elektrische Leistung des Mikrocontrollers nicht aus. Er kann zwar 5V Spannung erzeugen, aber keine hohen Ströme fließen lassen. Mit einer externen Spannungsquelle¹⁾ und einer Transistorschaltung kann man zwar die höhere Leistung der externen Energiequelle nutzbar machen - allerdings könnte man den Motor nur ein- und ausschalten, aber nicht umpolen, um seine Drehrichtung zu ändern.

Deshalb verwenden wir zur Ansteuerung von Gleichstrommotoren eine **Motortreiberplatine**, mit der man zwei Motoren ansteuern kann.



IC L298

Der zentrale Baustein auf der Motortreiberplatine heißt *IC L298*. IC steht für „Integrated Circuit“, also integrierter Schaltkreis. Im Inneren des L298 werkeln 8 geschickt mit einander verschaltete leistungsfähige Transistoren, eingebaut in ein einziges Gehäuse. Die Schaltung nennt man auch **H-Brücke**.

Motorvorbereitung

Je nachdem in welchem Zustand du dein Material vorfindest, musst du noch ein paar Vorbereitungen treffen. Der folgende Abschnitt ist versteckt, so kannst du ihn überspringen, wenn du schon fertig angeschlossene Motoren/Akkus/Motortreiber hast.

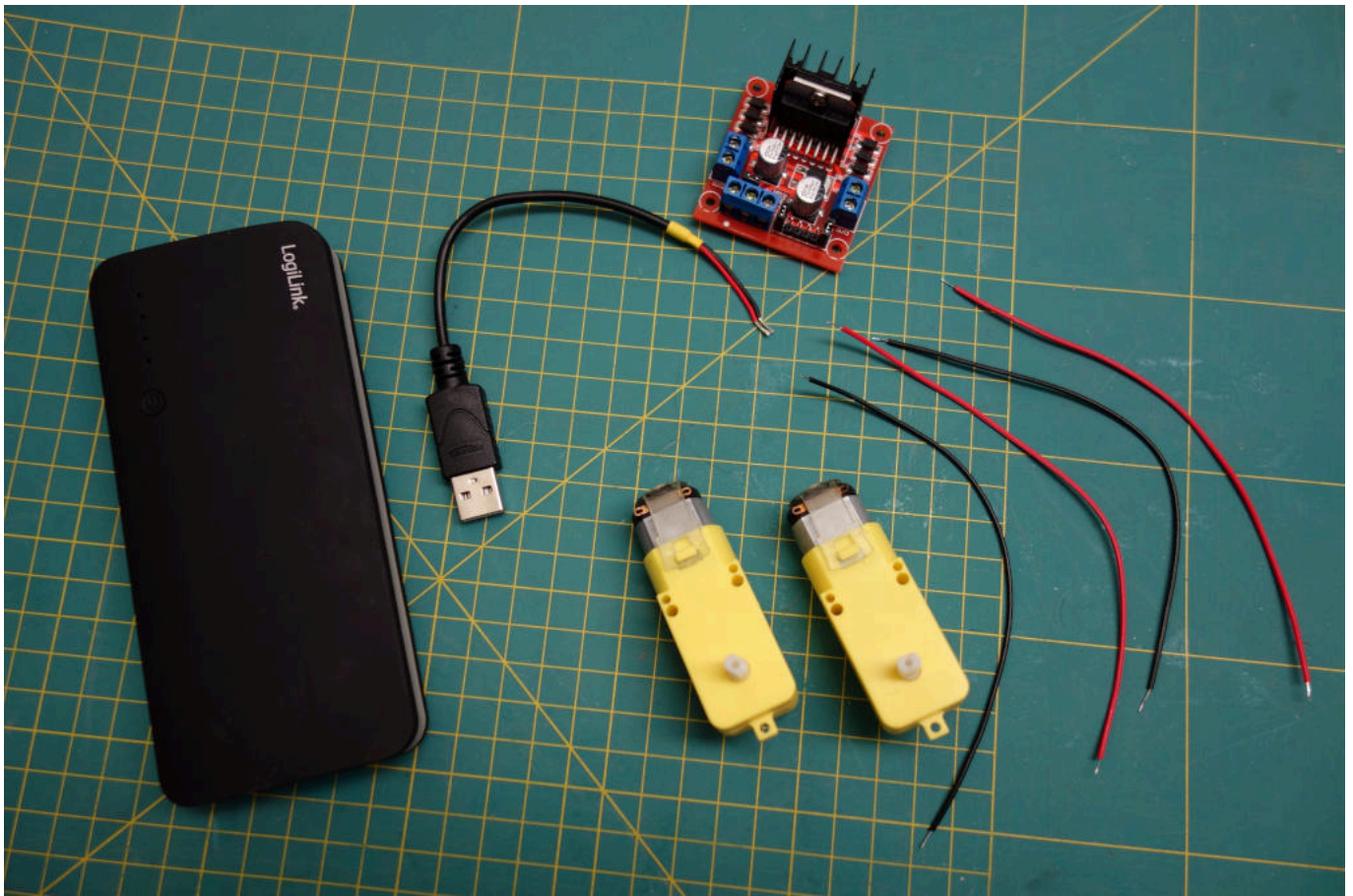
Andernfalls: Klappe die Infos aus!

[Ausklappen!](#)

Benötigtes Material

Im Bild sind zu sehen:

- Motortreiberplatine
- 4 Kabel (Motor ↔ Motortreiber)
- USB → 5V offene Enden. Das rote Kabelende ist +5V, das schwarze GND.
- USB Akkupack
- Nicht im Bild: Ein (schwarzes) Breadboardkabel, eine Seite Male, die andere verzinnt, um die Masse des Arduino mit der des Motortreibers zu verbinden.

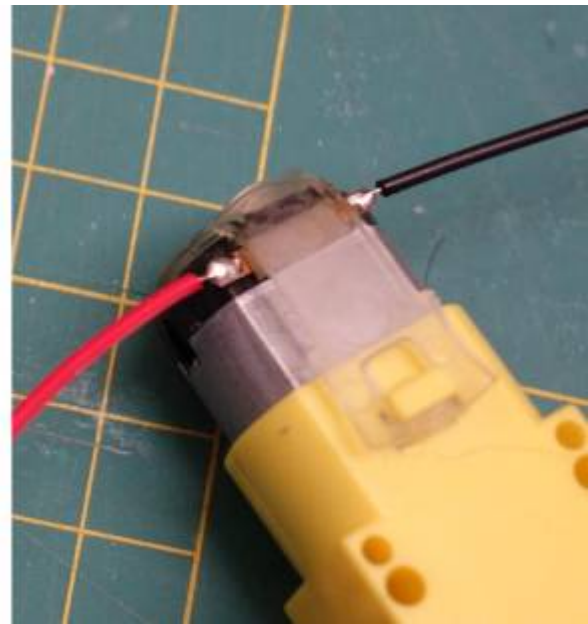


Kabel an die Motoren anlöten

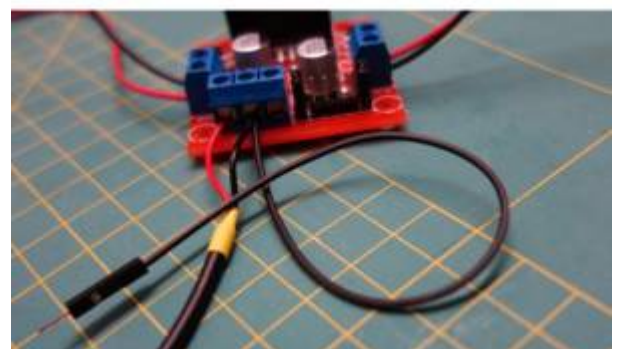
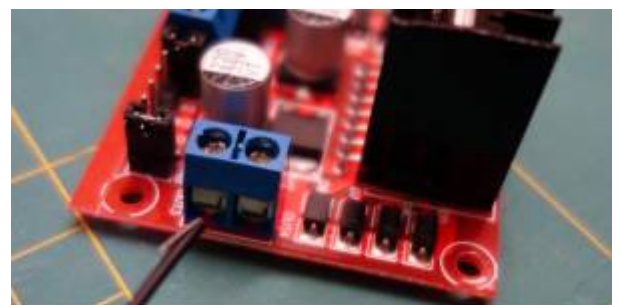
Wenn die 4 Kabel noch nicht mit den Motoren verlötet sind, muss das als erstes erledigt werden.

- Die Kabel müssen an die kleinen Metalllaschen angelötet werden.
- Häufig sind die Kabel bereits verzinnt, dann musst du nur noch die Laschen verzinnen und die Kabel anlöten.
- Achte darauf, dass du die transparenten Kunststoffbänder nicht mit den LötKolben berührst, sonst verschmort der Kunststoff.
- Jeder Motor bekommt ein schwarzes und ein rotes Kabel. Dabei ist es egal, welches Kabel an welche Lasche gelötet wird²⁾.

- Verzinne bei dieser Gelegenheit außerdem das schwarze Massekabel an seinem offenen Ende großzügig.



Motoren und USB-Kabel an den Motortreiber anschließen



- Löse die schrauben an den blauen Anschluss terminals
- Schließe links und rechts jeweils einen Motor an. Knicke dabei die dünne Litze des Kabels nach hinten, so dass du das ganze Kabel bei Festziehen der Schraube einklemmen kannst. Auch hier ist es egal, welches Kabel in welchen Anschluss geklemmt wird.



Achtung: Beim **Anschluss des USB Kabels muss die korrekte Polung beachtet werden**, andernfalls können Bauteile beschädigt werden!

Betrachte den Motortreiber von vorne (siehe Bild rechts).

- SchlieÙe das schwarze Kabelende des USB-Kabels und das verzinnte Ende des Masse-Kabels am mittleren Anschluss des Dreierterminals an.
- SchlieÙe das rote Kabelende des USB-Kabels am linken Anschluss des Dreierterminals an.

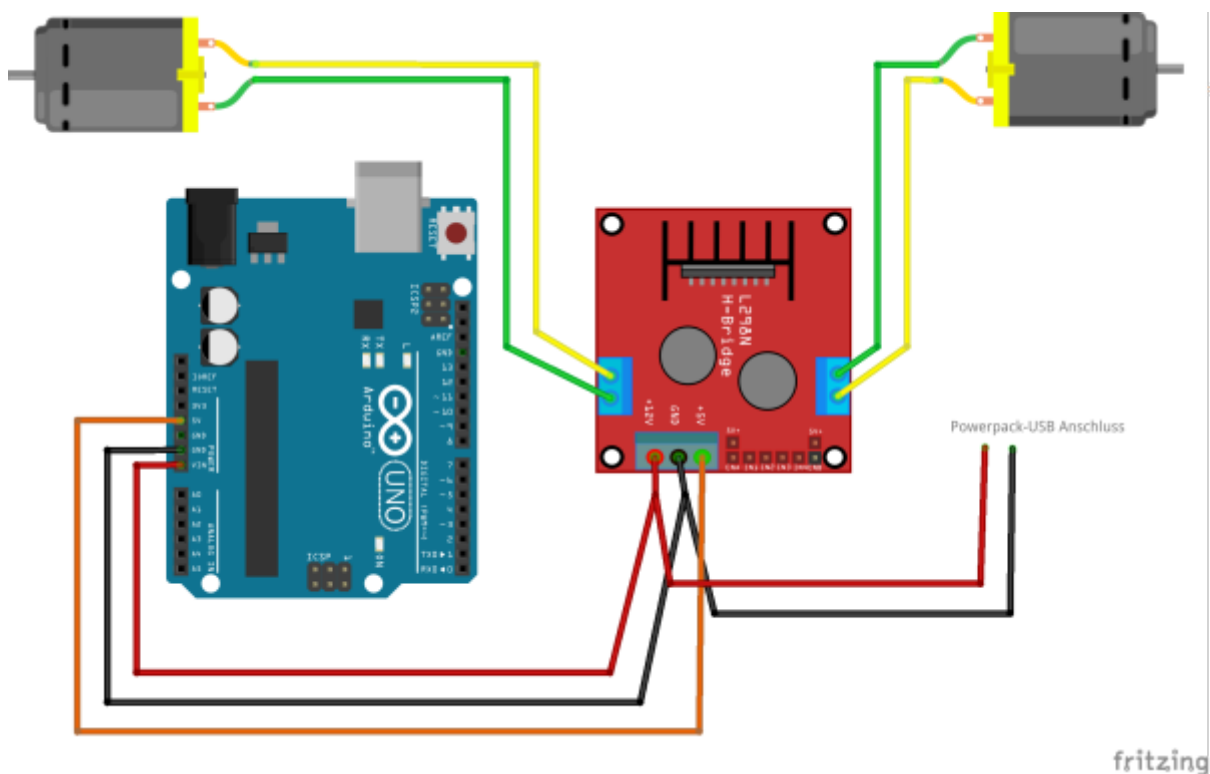


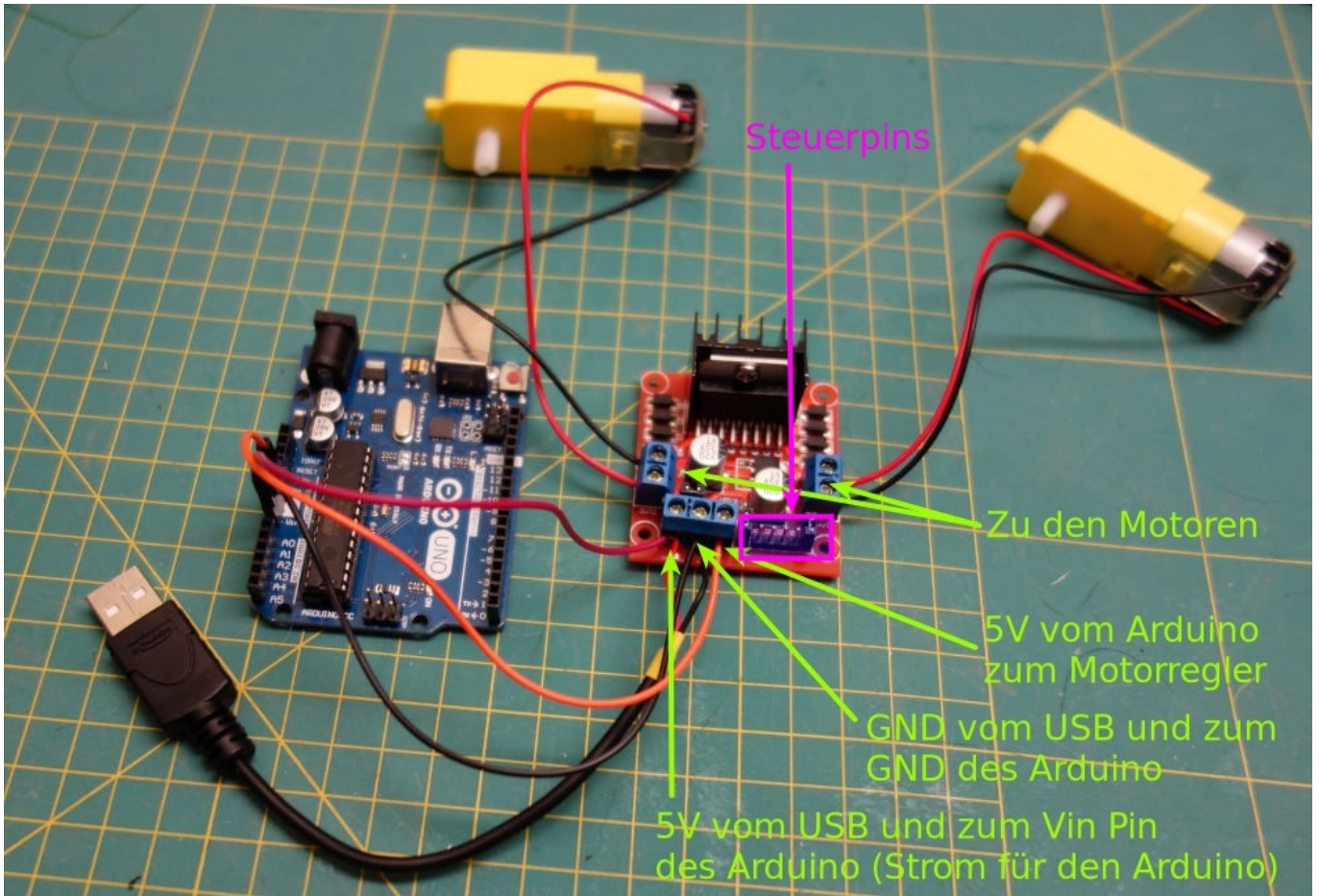
Wenn du dir sicher bist, dass du alles korrekt angeschlossen hast, kannst du den USB Stecker in die Powerbank einstecken - wenn du unsicher bist, lasse deinen Aufbau kontrollieren.

Wenn alles klappt, sollte die rote LED auf dem Motorregler leuchten - deine Motoren sind jetzt bereit, vom Arduino über den Motorregler gesteuert zu werden.

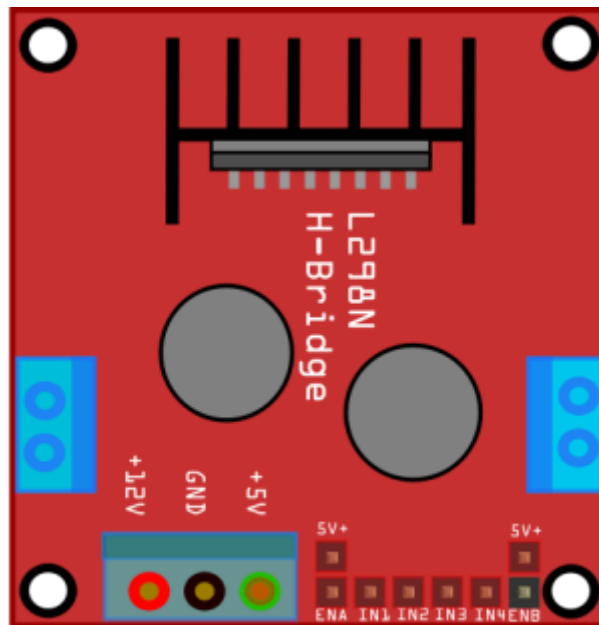
Motorsteuerung

Zunächst musst du den Motortreiber mit dem Arduino verkabeln:





Steuerung des Motortreibers



fritzing

Die angeschlossenen Motoren werden über die untere Pin Leiste gesteuert (Beschriftung ENA, IN1, IN2, IN3, IN4, ENB). Die drei Pins links steuern den Motor, der an den linken Anschlüssen angeschlossen ist, die Pins rechts den auf der rechten Seite.

Die mit ENX beschrifteten Pins dienen dazu, die Energiezufuhr zu regulieren, was dazu führt, dass sich die Motoren schneller oder langsamer drehen. Je höher die Spannung an der ENX Pins ist, desto mehr Energie erhalten die zugehörigen Motoren.

Wenn die ENX Pins nicht angeschlossen sind, ist die Spannung 0 und der betreffende Motor dreht sich nicht.

Die mit INX bezeichneten Pins legen die Drehrichtung fest: Ist einer HIGH und der andere LOW, dreht sich der Motor in eine Richtung - vertauscht man HIGH und LOW In die andere. Sind beide Pins HIGH oder beide LOW, steht der Motor still.



(A1)

Verbinde die rechten beiden Pins mit den Pins 3 und 4 deines Arduino. Schreibe nun ein Programm, das immer wieder Pin 3 HIGH und Pin 4 LOW schaltet, zwei Sekunden wartet, dann beide Pins HIGH (oder LOW) schaltet und wieder zwei Sekunden wartet. Nun sollte der Motor laufen, stehen, laufen, stehen...

Nimm auf diese Weise auch den anderen Motor in Betrieb. Achte darauf, dass die ENX Pins mit 5V versorgt werden, beispielsweise mit einem Jumper oder einem kleinen Kabel.



(A2)

Wenn man die ENX Pins mit Analogausgängen des Arduinos verbindet, kann man die Geschwindigkeit verändern: Je höher die Spannung an den Pins ist, desto schneller dreht sich der Motor.

Schreibe ein Programm, das dein Fahrzeug sanft anfahren lässt und nach einigen Sekunden wieder sanft abbremst.



(A3)

Wenn du beide Motoren in Betrieb genommen und getestet hast, schreibe Unterprogramme für die folgenden Situationen:

- „Beide vorwärts“: Beide Motoren sollen sich in eine Richtung drehen
- „Beide rückwärts“: Beide Motoren sollen sich in eine Richtung drehen, aber andersherum als bei „Beide vorwärts“ |
- „Links vorwärts“: Der linke Motor dreht sich, der rechte steht.
- „Rechts vorwärts“: Der rechte Motor dreht sich, der linke steht.
- „Stopp“: Beide Motoren stehen still

Führe Motorstart und Motorstopp dabei möglichst immer „sanft aus.

Mit diesen Unterprogrammen bist du der Steuerung eines Fahrzeugs schon recht nahe, das zwei Motoren als Hinterräder benutzt.

1)

Batterien, Akkupack

2)

Warum?

From:

<https://wiki.qg-moessingen.de/> - **QG Wiki**

Permanent link:

<https://wiki.qg-moessingen.de/faecher:nwt:arduino:lernbaustein1:motorsteuerung:start?rev=1663610930>

Last update: **19.09.2022 20:08**

