

Schrittmotoren & das FabScan-Shield

Präzises Bewegen mit Schrittmotoren!

Das **FabScan-Shield** macht es euch einfach, mit einem Arduino mehrere **Schrittmotoren** anzusteuern!

Schrittmotoren, das sind keine einfachen Elektromotoren sondern höchst präzise Motoren, die ihr sehr genau steuern könnt. Ähnlich komplex wie die Motoren ist normalerweise auch ihre Verkabelung, doch das FabScan-Shield nimmt euch jede Menge Arbeit ab.

Einige Dinge gibt es, die ihr wissen müsst, bevor ihr loslegen könnt:

Die Schrittmotor**controller** (die kleinen grünen Platinen auf eurem Shield, siehe Abb. 1) haben drei Steuereingänge, die ihr benutzen müsst:

En(able): Dieser Pin muss auf LOW geschaltet werden, damit der Controller aktiviert wird.

Dir(ection): Ist dieser Pin auf HIGH geschaltet, dreht euer Motor im Uhrzeigersinn, ist er LOW dreht er genau anders herum.

Dreht sich der Motor bei euch verkehrt herum ist das gar nicht schlimm. Merkt euch: Wechselt ihr die Spannung am Pin Dir, ändert ihr die Drehrichtung!

Step: Dieser Pin veranlasst die Bewegung des Motors. Für jedes Einschalten dieses Pins dreht sich der Motor um genau eine Position weiter. Also: *An-Aus-An-Aus* dreht den Motor genau zwei Schritte weiter. Für eine Umdrehung braucht der Motor **200** Schritte.

Denkt daran, dass euer Arduino wahnsinnig schnell schaltet, gebt dem Motor also ein wenig Zeit die Bewegung auszuführen. *Stichwort: delay()*

Tipp: Ist delay() nicht schnell genug, guckt euch mal die Funktion delayMicroseconds() an.

Jede der grünen Platinen kann genau einen Schrittmotor steuern, wir haben für euch jedes FabScan-Shield mit zwei davon bestückt. Angeschlossen werden die Motoren jeweils an den vier Kontakten daneben (siehe Abb. 1).

Die Anschlüsse der Motoren werden an folgende Pins des Arduino angeschlossen (diese könnt ihr wie gewohnt ansteuern):

Funktion	Motor 1	Motor 2
Enable	2	5
Step	3	6
Direction	4	7

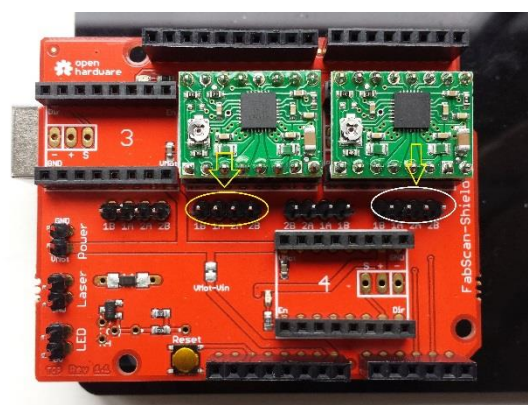


Abb. 1:
Motoranschlüsse

Schrittmotoren & das FabScan-Shield

Die digitalen Pins **8-13** des Arduino und die analogen Pins könnt ihr also noch frei verwenden. Diese werden vom FabScan-Shield auch nach außen geführt, sind dort aber leider nicht beschriftet. Um es euch ein bisschen einfacher zu machen, haben wir das in Abbildung 2 für euch übernommen:

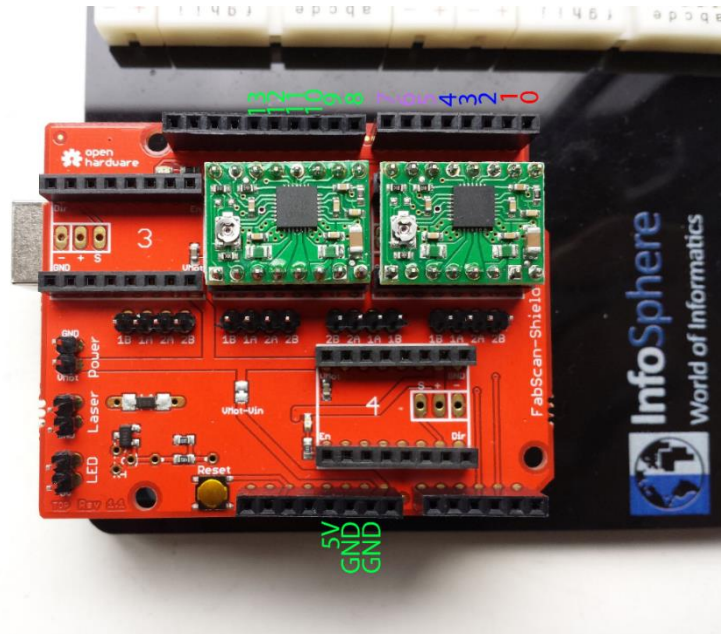


Abb. 2: Pins am Shield

Die Controller nehmen es euch übel, **wenn kein Motor angeschlossen ist**. Merkt euch: Ist der Arduino mit Strom versorgt, müssen Motoren angeschlossen sein!

Außerdem reicht der Strom vom USB-Anschluss für die starken Motoren nicht aus. Lasst euch daher vom Betreuer ein **Netzteil** geben und erklären!



```
int enPin = 2;
int stepPin = 3;
int dirPin = 4;

void setup()
{
  pinMode(enPin,OUTPUT);
  pinMode(stepPin,OUTPUT);
  pinMode(dirPin,OUTPUT);
  digitalWrite(enPin,LOW); // Controller einschalten
}

void loop() {
  digitalWrite(dirPin,HIGH); // Drehrichtung festlegen
  for(int i = 0; i < 200; i++) // 200 Schritte = 1 Umdrehung
  {
    digitalWrite(stepPin,HIGH);
    delay(1);
    digitalWrite(stepPin,LOW);
    delay(1);
  }
  delay(1000);
}
```

Dieses einfache Beispielprogramm dreht Motor 1 um eine volle Drehung und wartet dann eine Sekunde.

Quellen: Sämtliche Abbildungen stammen vom Infosphere-Team.