Arduino-Projekt

Schrittmotoren ansteuern

Um neben dem Gleichstrommotorantrieb unserer Lokomotiven begrenzte Drehbewegungen im Funktionsmodellbau auszuführen, bieten sich Servos an. Bewegungen, wie sie eine Schranke ausführen muss oder bewegte Figuren sind nur zwei Beispiele. Fast jeder moderne Decoder bietet heute die Möglichkeit, einen oder mehrere Servos zu steuern. Abseits der digitalen Modellbahn helfen Microprozessoren wie unser Arduino (siehe Ausgabe 106 der „volldampf“).

Im Gegensatz zum Servo, der nur einen begrenzten Drehwinkel abdeckt, kann ein Schrittmotor beziehungsweise Stepper über den 360-Grad-Winkel hinaus und wie ein einfacher Motor unendliche Umdrehungen ausführen. Der Vorteil dieser Stepper liegt jedoch auch in der programmierbaren Bewegung, was Geschwindigkeit und Bewegungsabläufe angeht.

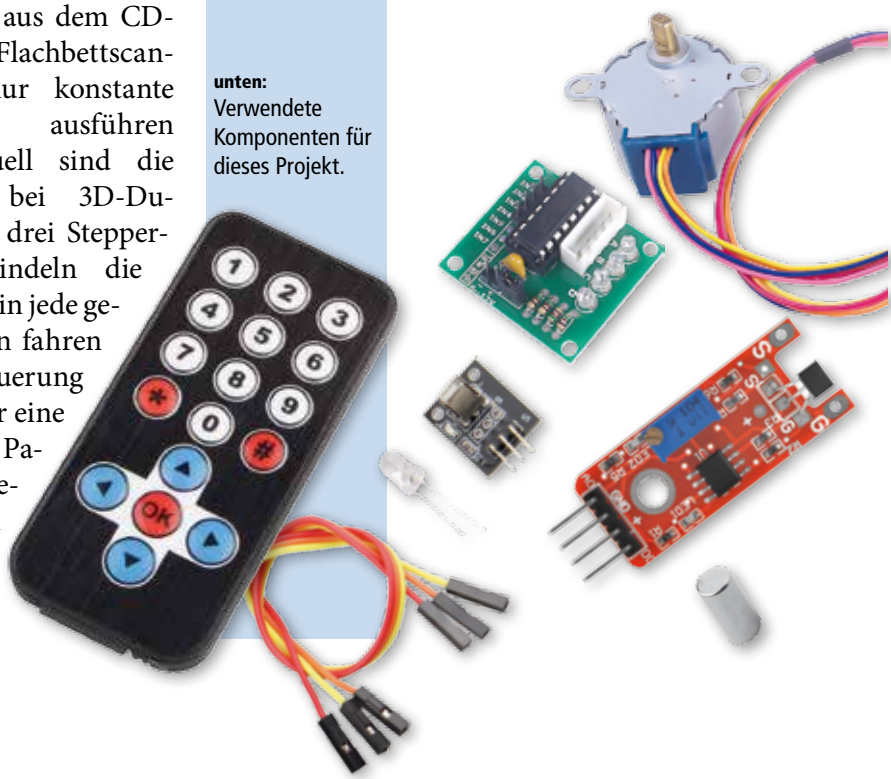
Wir kennen Stepper zum Beispiel als Antrieb aus dem CD-Player oder aus Flachbettscannern, wo er nur konstante Drehbewegungen ausführen muss. Hochaktuell sind die Spindelantriebe bei 3D-Druckern, bei denen drei Stepperangetriebene Spindeln die Werkzeugeinheit in jede gewünschte Position fahren können. Die Steuerung wird zumeist über eine programmierte Parameterliste ausgeführt, die seriell an eine Steuereinheit übermittelt wird.

oben:
Eine Drehscheibe wie diese, auf der Anlage von Joker-Rügen, läst sich mit einem Arduino steuern.

Bild: Dieter Messerschmidt

unten:
Verwendete Komponenten für dieses Projekt.

Wie genau ein Stepper funktioniert, soll hier nicht das Thema sein. Wichtig für jemanden, der es mal ausprobieren möchte, sind jedoch Angaben wie Schrittauflösung – das ist die Anzahl einzelner Positionen bei 360 Grad – sowie Spannungs- und Stromstärke. Diese Werte zeigen, ob ein Stepper für den gewünschten Einsatz ausreicht.



Wenn er zu schwach ist, können Schritte verloren gehen, wenn die Kraft nicht ausreicht, um ein Hindernis zu überwinden. Dadurch würde die genaue Positionierung gestört. Mehr Kraft oder kleinere Schritte bedeuten, dass die Gefahr des Schrittverlustes sinkt und der Motor auch deutlich leiser wird. Für unsere Modellbahnanwendungen reichen jedoch oft die einfachsten Ausführungen aus.

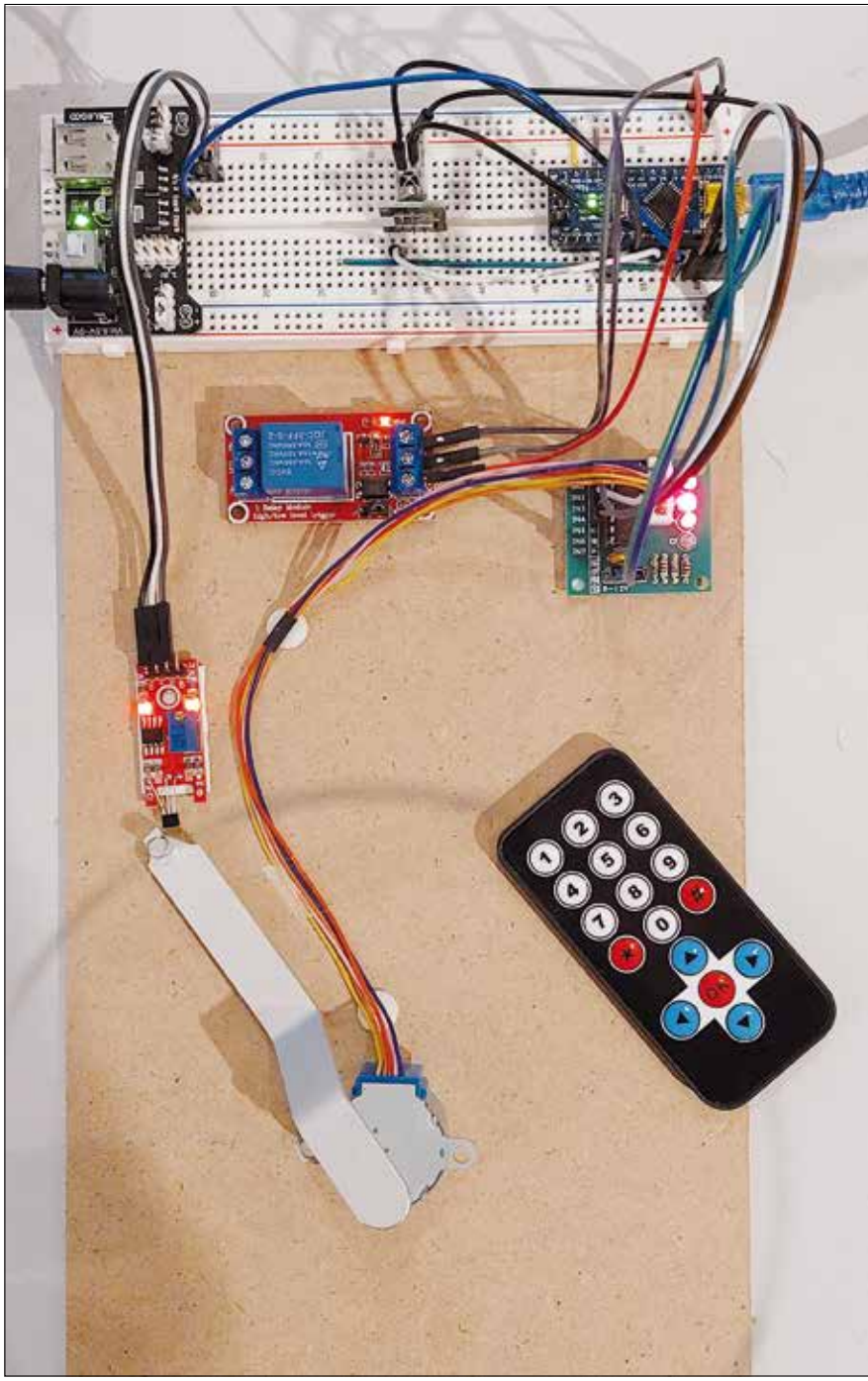
Was kann man nun mit einem Schrittmotor auf der Modellbahn machen? Wenn man die oben beschriebenen Eigenschaften betrachtet, bietet sich diese Antriebsart für eine Drehscheibe geradezu an: Gesteuerte Geschwindigkeit, Positionierung an bestimmten Winkeln und eine endlose Kreisbewegung, falls notwendig. Ebenso kann man sich vorstellen, ein Pärchen im Dreivierteltakt

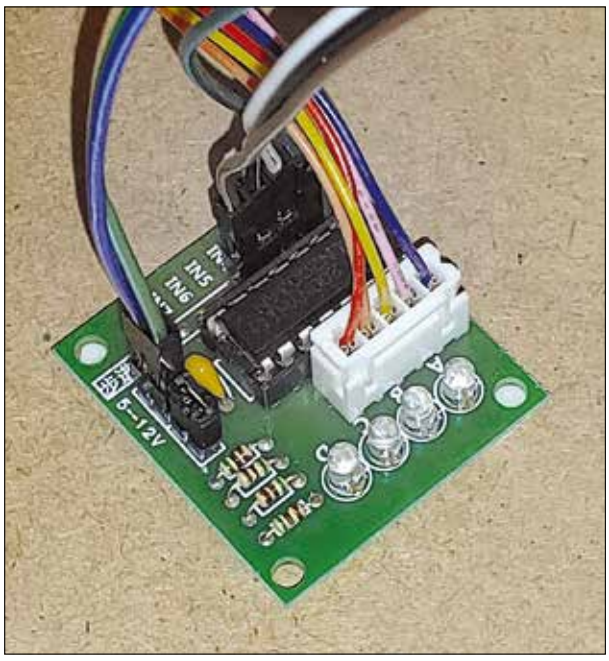
rechts: Der komplette Versuchsaufbau: Oben, das Breadboard mit Spannungsversorgung, Infrarotempfänger und Arduino.

Darunter links erkennt man den Hall-Sensor., in der Mitte den Relaisbaustein und rechts die Steuerplatine.

Der Aluminium-Bügel auf dem Stepper simuliert die Drehscheibe.

Ohne Spannungsversorgung schlugen die verwendeten Komponenten gerade mal mit 15 Euro zu Buche.





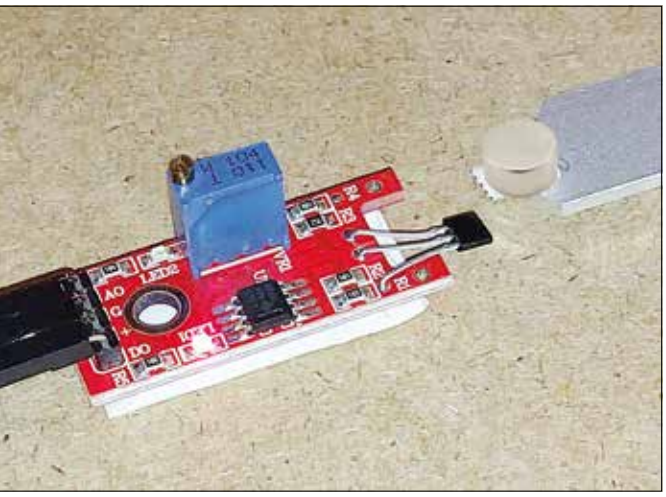
oben: Die Steuerplatine, die die Befehle des Arduinos für den Stepper umwandelt.

durch das Tanzkaffee kreisen zu lassen oder über eine Welle (Gewindestange) eine Figur oder ein Fahrzeug entlang einer Strecke frei zu bewegen.

Unser Beispiel einer Stepper-Steuerung beruht wieder auf einer Arduino-Nano-Variante von Atmega und dem Schrittmotor 28BYJ-48 mit Antrieb-Modulplatinen, die bestens für kleine Modellbahnprojekte geeignet sind und für unter zwei Euro im Set zu haben sind.

Unabhängig von der Antriebsart einer etwaigen Drehscheibe wollen wir hier nur eine programmtechnische Vorgehensweise aufzeigen. Mit der Anpassung einiger Werte wäre der Sketch jedoch an fast jede Drehscheibe anzupassen und mit vielen Funktionen erweiterbar.

Zuerst muss der Stepper über die Steuerplatine an den Arduino angeschlossen werden. Das erfolgt über vier Leitungen. Der Sketch, von unserer Webseite downloadbar, stellt sich folgendermaßen dar: Wir nutzen die Bibliothek „AccelStepper.h“, die zunächst in der Programmierumgebung installiert werden muss. Nach der Einrichtung der entsprechenden Pins am Ardu-



ino wird der Stepper innerhalb des „setup()“ in einer langsamen Geschwindigkeit in Gang gesetzt.

Nun kommt ein zweiter Baustein zum Einsatz. Da der Arduino nach dem Einschalten „dumm“ ist, weiß er natürlich nicht, wo die Drehscheibe gerade steht. Wir benötigen also einen definierten Nullpunkt. Hierzu verwenden wir einen Hall-Sensor wie er zum Beispiel auch in Lokgetrieben für die Magnetische Glocken- oder Pfeifen-Auslösung Verwendung findet.

Ein Arduino-konformer Baustein ist zum Beispiel der KY-024, der zum Preis von rund zwei Euro zu haben ist. Zusätzlich benötigen wir einen kleinen Neodym-Magneten. Wenn der Hall-Sensor den Magneten passiert, registriert dies unser Programm und setzt den erreichten Startpunkt für alle kommenden Aufgaben.

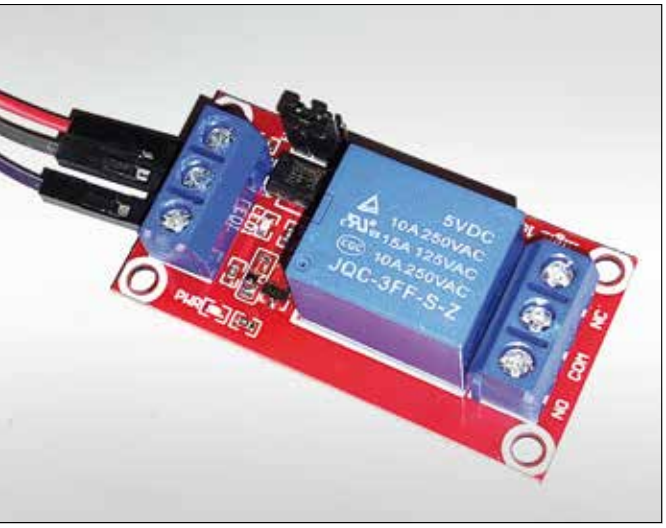
Die Drehscheibe wäre also nun einsatzbereit. Wir müssen unserem Stepper nur noch mitteilen, wohin er sich nun bewegen soll. Hierzu benötigen wir zusätzliche Eingaben. Da wären zum Beispiel je Gleisabgang eine Taste an je einem Pin am

oben:
Der Hall-Sensor reagiert auf den Neodym-Magneten.

Mitte:
Der IR-Sensor.

unten:
Der Relais-Baustein wird für eine eventuelle Umpolung benötigt.

Arduino oder eine Tastatureingabe per Tastenfeld möglich. Wir haben uns für eine weitaus schickere Möglichkeit entschieden. Da es recht einfach ist, per Infrarotsender Befehle an den Microcontroller zu senden, nutzen wir ein Set aus Empfängerbaustein und Handsender, welches für rund vier Euro erhältlich ist.



Im „loop()“ unseres Programms wird also unaufhörlich der Eingang des IR-Empfängers abgefragt. Je nach empfangenem Signal wird dem Stepper ein entsprechender Anfahrwinkel mitgeteilt, den er dann sofort anfährt. Zur besseren Übersicht haben wir in unserem Beispiel die möglichen Winkel beziehungsweise Tastenbelegungen unserer Infrarot-Fernbedienung in einer Liste zusammengefasst.

Die Fahrbewegung in unserem Sketch kann immer nur zwischen null und 360 Grad stattfinden. Die hierzu verwendeten Funktionen aus der Bibliothek sind recht einfach zu handhaben, lassen aber keine Überquerung des Nullpunktes zu. Das ist jedoch bei den meisten Drehscheiben, die eine Anfahrt auf der einen Seite und drei bis vier Abgänge auf der gegenüberliegenden Seite haben, nicht notwendig. Mit etwas Grips und programmiertechnischem Aufwand ist aber auch eine „relative“ Fahrt zu machen. Sie werden sehen, die Bibliothek bietet viele Möglichkeiten.

Ein weiterer Vorteil der Null-360-Grad-Fahrt ist, dass man sich kaum Gedanken um die Verdrahtung der Bühnengleise machen muss. Ein angeschlossenes Kabel dreht sich problemlos eine Rund mit. Sollte die Drehbewegung jedoch 180 Grad überschreiten, so ist eine Umpolung der Gleisspannung notwendig. Dies erreichen wir über ein Relais, welches winkelabhängig umgeschaltet werden kann.

Unser Beispiel der Drehscheibensteuerung ist bestimmt noch nicht ausgereift, sondern soll viel mehr die Möglichkeiten der Schrittmotorsteuerung aufzeigen. Die freien Tasten auf der Fernbedienung ließen sich für langsames Vor- und Zurückfahren, händische Nullpunktjustierung oder endlose Fotofahrt mit langsamer Bewegung nutzen.

Sketch, Fritzing-Plan und weitere Informationen finden Sie wieder auf unserer Webseite unter www.volldampf.org im Downloadbereich zum Arduino-Projekt. (pb)