

Philip Steffan

# Floppy-Symphonie

Knarz, brumm, klick: Diskettenlaufwerke sorgten früher für ungewollten Krach am Schreibtisch. Heute erlebt die obsoleete PC-Hardware einen zweiten Frühling als Arduino-gesteuerte Musikinstrumente.



## Zutaten

- Arduino UNO (oder kompatibles Board)
- ein bis acht ausgemusterte 3,5-Zoll-Laufwerke aus IBM-kompatiblen PCs
- ein bis acht PC-Steckbrücken (Jumper)
- 5-Volt-Netzteil, z. B. altes PC-Netzteil
- Kabel zum Verdrahten, z. B. Netzwerkkabel oder alte PC-Floppykabel
- optional: 2 bis 16 „Jumper Wires F/M“, Länge 200 mm



## Kurzinfo



**Zeitaufwand:**  
ab 2 Stunden



**Kosten:**  
ab 25 Euro



**Programmieren:**  
Arduino-IDE



**Löten:**  
optional, Grundkenntnisse



**Elektronik:**  
Grundkenntnisse

### Schwierigkeitsgrad



## FLOPPY?

Eigentlich eine unsinnige Verkürzung von „floppy disk drive“ – dem Laufwerk für „schlafte Scheiben“. Gemeint ist die biegsame Magnetscheibe in der Diskettenhülle, im Gegensatz zur Festplatte, der „hard disk“.

Erinnern Sie sich noch an Disketten? Die magnetischen Speicherscheiben waren in verschiedenen Größen über 30 Jahre lang das Mittel der Wahl, um Daten zu sichern und zu transportieren. Seit der Jahrtausendwende haben USB-Sticks, SD-Karten und nicht zuletzt das Internet die Diskette verdrängt. Heute taucht sie nur noch als grafisches Symbol für „Speichern“ in vielen Computerprogrammen auf.

Dennoch haben Sie möglicherweise noch ein Diskettenlaufwerk im Keller herumliegen, aus einem alten PC, falls doch noch mal eine Diskette auftaucht. Da dies vermutlich nie passieren wird: Höchste Zeit, es aus seinem Dornröschenschlaf zu wecken und ein Musikinstrument daraus zu bauen. Das charakteristische Brummen und Klicken, mit dem der Lesekopf über die Spuren der Diskette wandert, kann man mit einem Mikrocontroller-Board wie dem Arduino gezielt steuern.

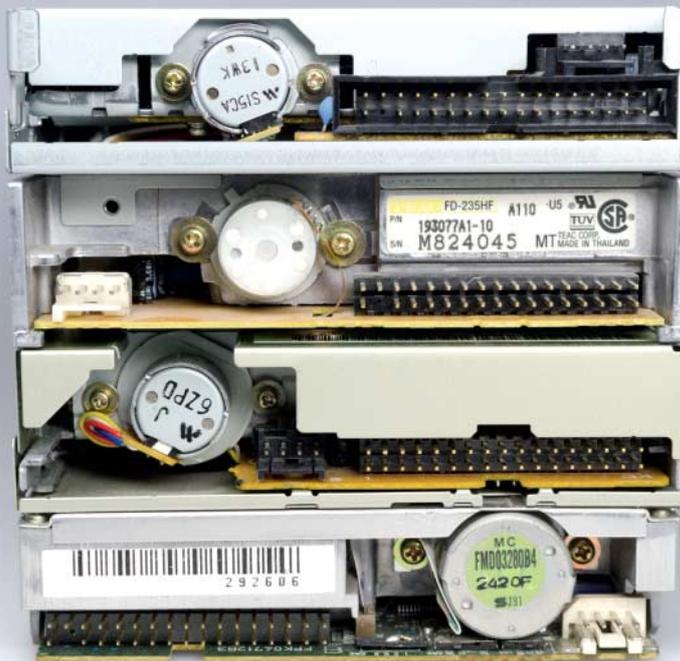
Die Idee stammt aus dem Internet: Bei Youtube gibt es seit einigen Jahren unter dem Stichwort „floppy music“ eine Vielzahl an Videos, in denen bekannte Melodien auf Diskettenlaufwerken und anderen ausgemusterten Peripheriegeräten wie Druckern und Scannern nachgespielt werden.

## So funktioniert es

In all diesen Geräten sorgen Schrittmotoren für die Bewegung. Sie rotieren nicht kontinuierlich wie einfache Elektromotoren, sondern drehen die Achse bei jedem elektrischen Impuls um einen definierten Schritt weiter. Dabei entsteht ein kurzes hörbares Geräusch, wenn der Rotor ruckartig stoppt. Folgen diese Klicks schnell genug aufeinander, nehmen wir sie als einen Ton mit einer bestimmten Tonhöhe wahr: Die Untergrenze dafür liegt bei etwa 16 Hertz, also 16 Impulsen pro Sekunde.

Um beispielsweise ein „kleines c“ mit einer Frequenz von 130,8 Hz mit einem Schrittmotor erklingen zu lassen, muss der Motor also 130,8 Steuerimpulse pro Sekunde von einer passenden Motorsteuerung erhalten. Für dieses Vorhaben sind ein übliches Diskettenlaufwerk und ein Arduino Uno das perfekte Team: Beide laufen mit einer Versorgungsspannung von 5 Volt (TTL-Logik) und der Floppy-Motor schaltet immer dann einen Schritt weiter, wenn Pin 20 des Laufwerksanschlusses auf Masse gelegt wird. Dafür reicht beim Arduino der grundlegende Befehl `digitalWrite(Pin,LOW)`.

Man benötigt also keine eigene Motorsteuerung, keine Pegelwandler und muss keine Gehäuse öffnen: Ein paar Kabel sind genug. Als Stromversorgung reicht ein ausrangiertes PC-Netzteil, das bei Ihnen möglicherweise in derselben Kiste wie das Diskettenlaufwerk auf Verwendung harrt.



**1** Jedes Laufwerksmodell sieht rückseitig ein wenig anders aus, die Anschlüsse sind aber immer identisch: Ein vierpoliger Molex-Anschluss für die Stromversorgung und ein 34-poliger Wannenstecker für die Verbindung zum PC über den sogenannten Shugart-Bus. Die Pins zählt man unten links beginnend durch: Die untere Reihe sind Pins 1 bis 33, die obere Reihe Pins 2 bis 34. Alle ungeraden Pins liegen auf Masse. Bei vielen, aber nicht allen Laufwerken fehlt Pin 3. Damit sollte verhindert werden, dass man das Floppy-Kabel falsch herum aufsteckte.

## Software

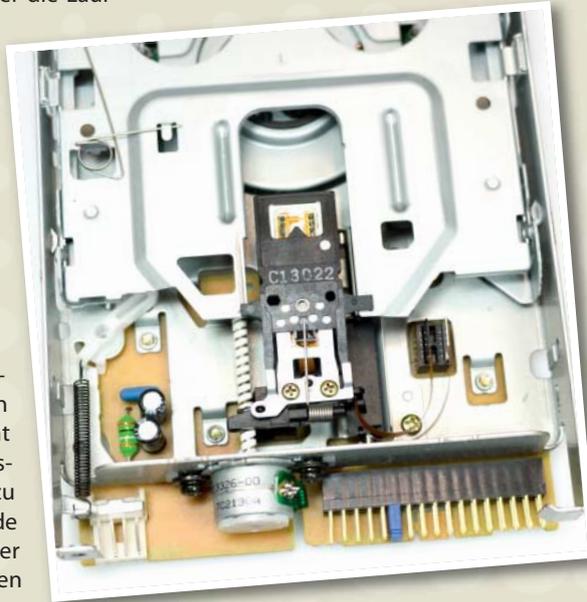
Floppy-Musik ist so populär, dass es bereits mehrere fertige Arduino-Programme gibt, die einem die manuelle Programmierung ersparen. Das Programmpaket Moppy des Programmierers SammyIAM besteht aus zwei Teilen: Einem Arduino-Sketch, der die Laufwerke ansteuert und einem Java-Programm auf dem PC, das den Arduino über USB in Echtzeit mit den zu spielenden Noten versorgt. Dazu kann Moppy entsprechende MIDI-Dateien öffnen, alternativ aber auch die Notendaten eines beliebigen MIDI-Geräts durchreichen. Mit einem MIDI-Keyboard am Rechner kann man die Laufwerke also auch live spielen.

Moppy bindet bis zu 16 Laufwerke musikalisch ein, davon kann der Arduino maximal acht gleichzeitig ansteuern. Mehr Diskettenlaufwerke führen zwar zu einem volleren Sound, ob am Ende eine Symphonie oder doch eher eine Kakophonie an die Ohren dringt, hängt aber vor allem von der Qualität der Steuerdaten ab: Wenn man sich bei einem Pop-Hit auf die charakteristische Bassline und die Melodie beschränkt, grooven die Floppys. Schiebt man eine ganze Orchesterpartitur per MIDI in den Aufbau, entsteht oft nur dissonanter Lärm.

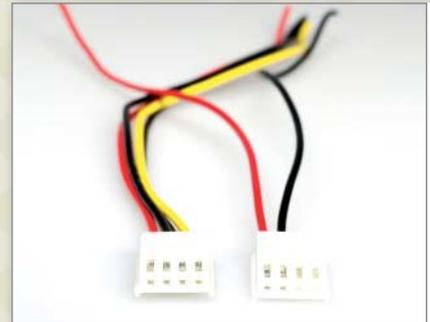
### Mach's nach, mach's besser

Das Schöne an diesem Bauprojekt ist allerdings, wie leicht es sich erweitern lässt: Das grundlegende Prinzip ist mit nur einem Diskettenlaufwerk leicht nachvollziehbar. Für sägende Mehrklänge sollten es schon drei Geräte sein. Perfekt, um damit an einem Wochenende anzufangen und dann bei Gefallen im persönlichen Umfeld mehr ausrangierte Laufwerke aufzutreiben. Unser Eigenbau tönt derzeit mit sechs vergleichsweise lauten Floppys von Teac aus der luxuriösen Restesammlung der c't-Redaktion.

Wenn auch bei Ihnen demnächst die Floppys singen: Machen Sie doch ein Video – das gehört sich heutzutage einfach so – und lassen Sie uns davon per Mail oder über unsere Social-Media-Kanäle (siehe Seite 6) wissen. (phs)



Der Schrittmotor sitzt hinten am Gehäuse und bewegt über ein Schneckengetriebe die Lese-/Schreibeinheit (schwarz) nach hinten und vorne.

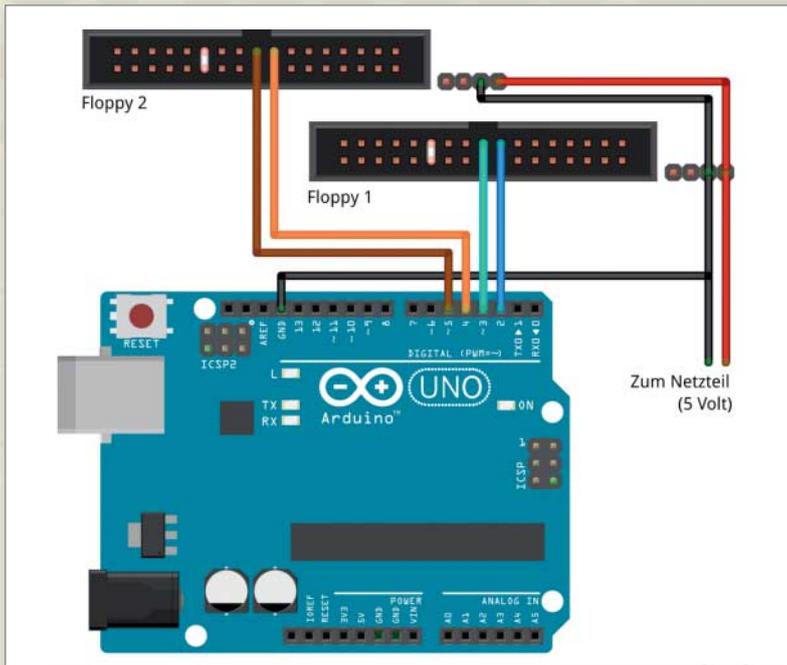


**2** Der passende Molex-Stecker vom PC-Netzteil lässt sich ebenfalls nur in eine Richtung aufstecken. Er führt 5 Volt im roten Kabel und 12 Volt im gelben Kabel, die schwarzen Kabel sind der Masseanschluss (GND). Das Laufwerk benötigt allerdings nur 5 Volt Versorgungsspannung. Um den Kabelsalat zu minimieren, kann man also die 12-Volt-Versorgung entfernen: Übt man mit einem Schraubendreher leichten Druck auf die Kontaktfedern aus, lassen sich die Anschlüsse einfach aus dem Steckergehäuse herausziehen.



**3** Damit aus dem Laufwerk Töne erklingen, müssen drei Signale kontrolliert auf Masse (GND) gesetzt werden: Drive Select B (Pin 12), Direction (Pin 18) und Step (Pin 20). Im ersten Fall reicht dazu ein Jumper, also eine kleine Steckbrücke, mit dem Pin 12 mit dem darunter liegenden Pin 11 verbindet. Sobald der Jumper gesetzt ist, sollte bei aktiver Stromversorgung die Leuchtdiode vorne am Laufwerk aufleuchten. Pins 20 und 18 verbindet man mit den Digitalpins des Arduinos, angefangen am dortigen Pin 2 (siehe Tabelle).

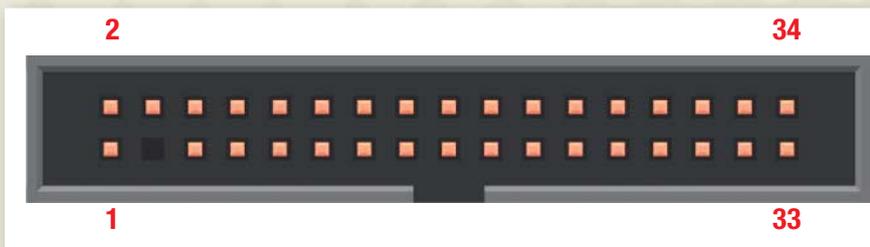
Links und Foren  
[www.ct.de/ch1401148](http://www.ct.de/ch1401148)



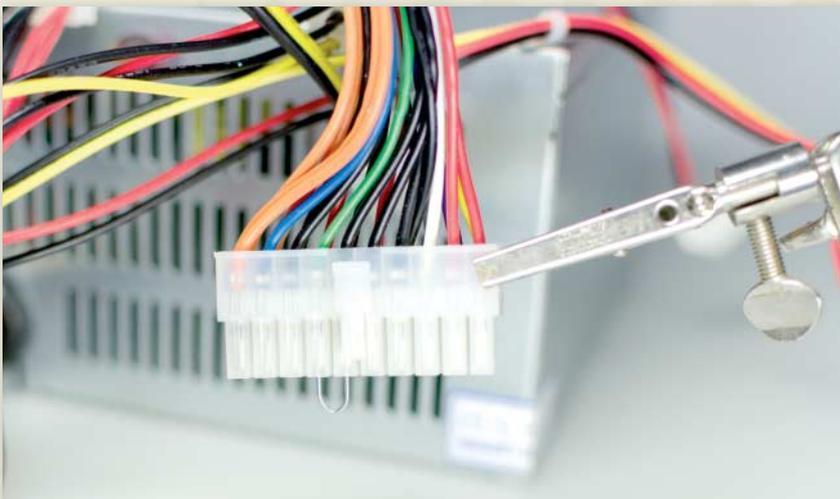
**4** So sieht die Verkabelung für zwei Laufwerke aus: Pins 20 und 18 von Laufwerk 1 kommen an Pins 2 und 3 des Arduinos. Pins 20 und 18 von Laufwerk 2 kommen an Pins 4 und 5. Für maximal acht Laufwerke geht es im selben Muster weiter. Die Arduino-Digital-Pins enden auf der Platine zwar bei Nummer 13, man kann aber auch die Analog-In-Pins unten rechts als Digitalausgänge nutzen. A0 entspricht dabei Ausgang 14, A1 Ausgang 15 und so weiter. Was auf keinen Fall fehlen darf: Eine Verbindung zwischen den Masseanschlüssen der Laufwerke und des Arduinos (schwarze Leitungen an GND). Wenn man den Arduino mit demselben Netzteil wie die Laufwerke versorgt, besteht diese Verbindung bereits über das Netzteil.

## 5,25"-LAUFWERKE

Falls Sie noch ein PC-Laufwerk für 5,25-Zoll-Disketten herumliegen haben: Damit funktioniert es auch. Im Internet heißt es sogar, dass die größeren Laufwerke die hohen Töne besser wiedergeben. Einziger Unterschied ist der Anschluss, der zwar auch 34-polig mit identischer Belegung ist, aber als Platinenstecker ausgeführt ist. Wenn kein passendes Floppy-Kabel mehr auffindbar ist, kann man auch Drähte direkt an die Anschlüsse löten.



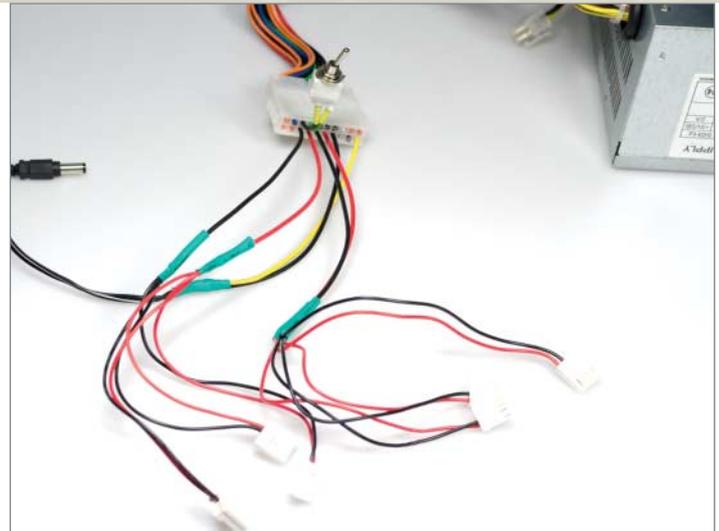
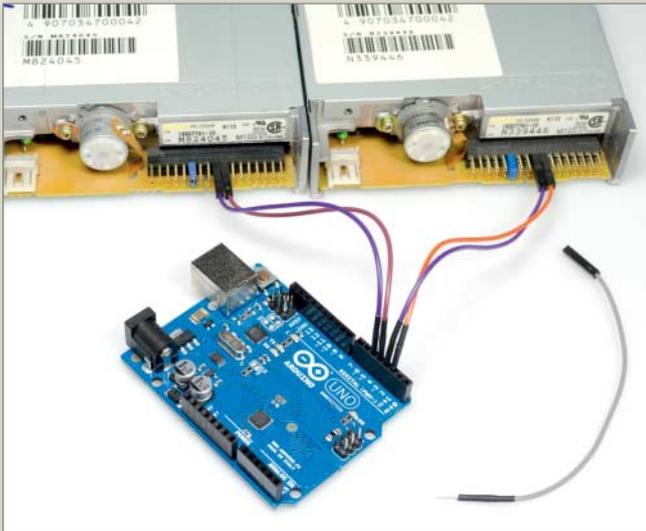
Die 34 Pins am Anschluss zählt man von unten links durch. Pin Nummer 3 fehlt oft, aber nicht immer.



**5** Wer ein PC-Netzteil zur Spannungsversorgung einsetzt, muss es erst einmal einschalten können. Ganz alte AT-Netzteile haben meist noch einen Schalter am Gehäuse, bei ATX-Netzteilen fehlt er in der Regel. Hier fließt der Strom erst, wenn man die grüne Leitung am 20- oder 24-poligen Mainboard-Stecker mit einer schwarzen Masseleitung verbindet. Dazu reicht im einfachsten Fall ein Stück Draht. Komfortabler ist es, einen kleinen Schalter zwischen grün und schwarz zu löten.

## Verbindungen

Arduino-Pin	Laufwerk	Laufwerks-Pin
2	1	20 (STEP)
3		18 (DIR)
4	2	20 (STEP)
5		18 (DIR)
6	3	20 (STEP)
7		18 (DIR)
8	4	20 (STEP)
9		18 (DIR)
10	5	20 (STEP)
11		18 (DIR)
12	6	20 (STEP)
13		18 (DIR)
14 (A0)	7	20 (STEP)
15 (A1)		18 (DIR)
16 (A2)	8	20 (STEP)
17 (A3)		18 (DIR)



**6** Wie man alles verkabelt, hängt vom Budget und dem Füllstand der persönlichen Kramkiste ab. Für alle Verbindungen lassen sich alte PC- und Netzkabel kannibalisieren, man kann auch Drähte direkt an die Stiftleisten der Diskettenlaufwerke anlöten. Unser Redaktionssetup sieht so aus: Die zwei Steuerverbindungen pro Floppy haben wir komfortabel über sogenannte „Jumper Wires“ mit dem Arduino verbunden, die an einem Ende einen Stift und am anderen Ende eine Buchse haben (Typ M/F). Sie passen genau auf die üblichen Leisten im Rastermaß 2,54 mm.

**7** Für die Stromversorgung haben wir sechs originale Molex-Floppyanschlüsse von mehreren Netzteilen stibitzt. Wer die nicht auftreiben kann: Auch sogenannte CD-Audio-Kabel, mit denen man im PC CD-ROM-Laufwerk und Soundkarte verbindet, eignen sich hierfür. Wir haben die Molex-Kabel mit der Buchse eines Netzteil-Verlängerungskabels für 2 Euro (AK MBV ATX bei Reichelt) zu einem kleinen Kabelbaum zusammengelötet. Ein Schalter zwischen der grünen Leitung und GND aktiviert das angeschlossene Netzteil (siehe Schritt 5). Der Vorteil: Sollte das Netzteil mal den Geist aufgeben, kann man einfach ein anderes anschließen.

```

Moppy | Arduino 1.5.5
Moppy
#include <TimerOne.h>
boolean firstRun = true; // Used for one-run-only stuffs;
//First pin being used for floppies, and the last pin. Used for looping over all pins.
const byte FIRST_PIN = 2;
const byte PIN_MAX = 17;
#define RESOLUTION 40 //Microsecond resolution for notes

/*NOTE: Many of the arrays below contain unused indexes. This is to
prevent the Arduino from having to convert a pin input to an alternate
array index and save as many cycles as possible. In other words information
for pin 2 will be stored in index 2, and information for pin 4 will be
stored in index 4.*/

/*An array of maximum track positions for each step-control pin. Even pins
are used for control, so only even numbers need a value here. 3.5" Floppies have
80 tracks, 5.25" have 60. These should be doubled, because each tick is now
half a position (use 150 and 90).
*/
byte MAX_POSITION[] = {
0,0,150,0,150,0,150,0,150,0,150,0,150,0,150,0,150,0};

//Array to track the current position of each floppy head. (Only even indexes (i.e. 2,4,6...) are used)
byte currentPosition[] = {
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

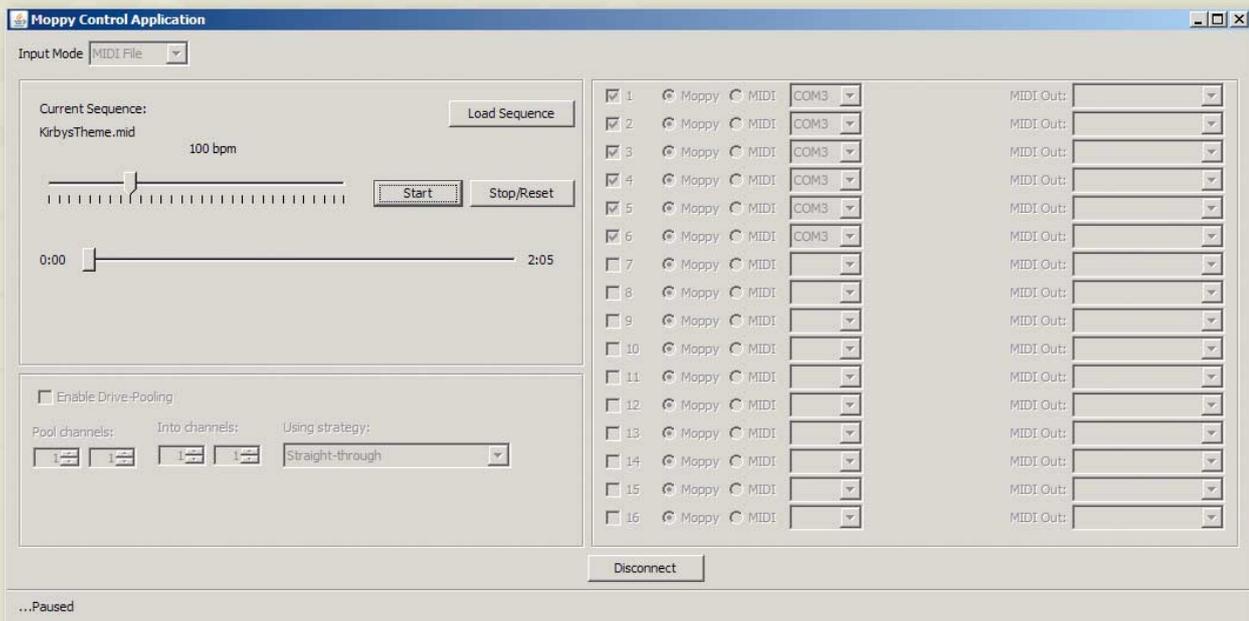
/*Array to keep track of state of each pin. Even indexes track the control-pins for toggle purposes. Odd indexes
track direction-pins. LOW = forward, HIGH=reverse
*/
int currentState[] = {
0,0,LOW,LOW,LOW,LOW,LOW,LOW,LOW,LOW,LOW,LOW,LOW,LOW,LOW,LOW,LOW,LOW};
    
```

**8** Kommen wir zur Software: Laden Sie Moppy Advanced und die Library TimerOne herunter (siehe „Links und Foren“). Starten Sie die aktuelle Arduino-IDE. Wählen Sie im Menü Sketch → Library importieren → Library hinzufügen und wählen Sie die Datei TimerOne-r11.zip aus. Entpacken Sie moppy-advanced.zip und öffnen Sie in der Arduino-IDE die Datei Moppy.ino im Verzeichnis Arduino/Moppy. Schließen Sie den Arduino an den PC an. Achten Sie darauf, dass im Menü Werkzeuge → Port der richtige serielle Anschluss gewählt ist. Unter Windows ist das meist der letzte „COMx“-Port, bei Linux und OS X etwa „/dev/tty.usbmodem123456“. Mit einem Klick auf „Hochladen“ wird das Programm auf das Board kopiert.

## RumbleRail

Der Hacker „kiu“ hat eine discotaugliche Profi-Version namens RumbleRail gebaut: Acht Laufwerke, jedes angesteuert von einem eigenen Controller. In der Mitte sitzt ein Steuermodul mit Display, das MIDI-Dateien von einer SD-Karte abspielt. Unter jeder Floppy steckt eine RGB-Leuchtdioden-Platine, die den Sound optisch begleitet. Alle zum Nachbau benötigten Dateien gibt es unter [www.schoar.de/tinkering/rumblerrail](http://www.schoar.de/tinkering/rumblerrail).





9 Da die nur als Quelltext herunterladbare Moppy-Steuer- software auf dem PC etwas knifflig zu starten ist, wenn Sie sich nicht mit Java auskennen, haben wir für Windows, OS X und Linux fertige Binaries (ausführbare Programme) kompiliert. Laden Sie die für Ihr Betriebssystem passende Fassung herunter und starten Sie das Moppy-Programm. Es gibt zwei Betriebsmodi, die Sie oben links unter „Input Mode“ auswählen können. Über „MIDI In Port“ kann man die Floppys über einen auf dem Computer vorhandenen MIDI-Port ansteuern, also eine Musiksoftware oder ein über ein MIDI-Interface angeschlossenes Keyboard. Damit lassen sich die Laufwerke auch live spielen. Im Modus „MIDI File“ kann man MIDI-Dateien, sozusagen digitale Notenblätter, laden

und über die Laufwerke abspielen. Im rechten Feld sehen Sie die Zahlen von 1 bis 16. Wählen Sie in allen Drop-Down-Feldern den seriellen Anschluss aus, an dem der Arduino hängt (siehe Schritt 8). Ein Klick auf „Connect“ öffnet die Verbindung. Jetzt können Sie, je nach Modus, MIDI-Dateien oder -Datenströme über die Laufwerke wiedergeben. Einige MIDI-Dateien zum Testen finden Sie im Moppy-Download-Archiv. Spielen Sie außerdem mit den Einstellungen unter „Enable Drive-Pooling“: Hier können Sie die bis zu 16 Spuren (Instrumente) einer MIDI-Datei auf die Anzahl der angeschlossenen Laufwerke umleiten. Ohne diese Einstellung würden Sie bei einem Eigenbau mit zum Beispiel vier Laufwerken sonst nie hören, was auf den Kanälen 5 bis 16 läuft.



So sieht unser Eigenbau mit sechs Laufwerken aus. Wichtig für den Sound: Eine harte Unterlage wie ein Tisch oder ein Brett, das die Vibrationen verstärkt.