

Unendlichkeits-Pendel

Ein faszinierender Hingucker ist das *Pendel-Interruptus*, das ewig pendelt und nie mit der rotierenden Lochscheibe kollidiert, die es immer wieder durchqueren muss.

von Martin Simeth



Von den Pendelanimationen des 3D-Grafik-Künstlers Andreas Wannerstedts inspiriert (siehe Link in der Kurzinfor), hatte ich überlegt, ob man eine solche Maschine nicht in echt nachzubauen könnte. Meine Wahl fiel auf das Pendel mit der Lochscheibe: Ein Pendel schwingt und passiert am unteren Punkt eine quer zur Pendelrichtung ausgerichtete, rotierende Lochscheibe. Nicht nur, dass das Pendel stetig schwingt und (scheinbar) keinen Antrieb benötigt, es passiert die Lochscheibe immer im genau richtigen Moment, um nicht anzustoßen.

Am Ende des Pendels hängt eine Holzkugel mit einem eingelassenen Magneten, auf den zwei Spulen unter der Bodenplatte wirken. Die Lochscheibe sitzt auf der Achse eines Schrittmotors. Der Pendelmagnet muss so schwach sein, dass er sich vom Metallgehäuse des Motors nicht stören lässt. Die Drehzahl der Lochscheibe wird entsprechend der Pendelfrequenz geregelt.

Im Internet gibt es viele einfache Umsetzungen von ewigen Pendeln: Das Pendel selbst ist ein Magnet, das in der Sensorspule eine Spannung induziert, woraufhin ein Transistor die Aktorspule einschaltet, die dann den Pendelmagneten abstößt. All diese Realisierungen haben jedoch einen großen Nachteil: Sie geben keine Schwingungsebene vor, sodass das Pendel in irgendeine Richtung schwingt. Mein Aufbau sollte aber von außen so reduziert wie möglich sein. Jedoch allein aufgrund der Erdrotation würde sich die Schwingungsebene eines Fadenpendels im Laufe der Zeit drehen (Foucault'sches Pendel).

Elektronik

Um die Erdrotation auszugleichen, könnte man zwei Aktorspulen und zwei Sensorspulen nutzen. Die intelligentere Lösung ist es, nur zwei Spulen einzubauen und diese abwechselnd sowohl als Sensor- als auch als Aktorspule einzusetzen. Die Steuerung hierfür übernimmt ein Arduino Nano, der über ein wenig Elektronik mit den Spulen verbunden wird.

Für den Spulenkörper habe ich gebräuchliche Lautsprecherspulen verwendet, bei denen ich den originalen Draht gegen 0,1mm-Kupferdraht ausgetauscht und davon so viel Draht aufgewickelt habe, bis der Spulenkörper voll war. Die exakte Impedanz der Spulen ist nicht wichtig, da die Software Abweichungen gut ausgleicht. Allerdings ist eine hohe Induktivität der Spulen für das Pendel hilfreich.

Über R4 und D1 beziehungsweise R8 und D3 werden die Kondensatoren C1 und C2 aufgeladen. Dabei prüft ein A/D-Wandler des Arduinos an den Spannungsteilern R2/R3 und R6/R7 permanent, ob eine Spannungs-

Kurzinfor

- » Perpetuum mobile als Deko-Objekt
- » Spulen als Sensor und Aktor nutzen

Checkliste



Zeitaufwand:
drei Tage



Kosten:
ca. 100 Euro



Programmieren:
Kenntnisse in Sketch, C++ und Arduino



Löten:
Aufbau und Verlöten auf Lochrasterplatine



Maschinen:
Laser cutter



3D-Druck:
optional

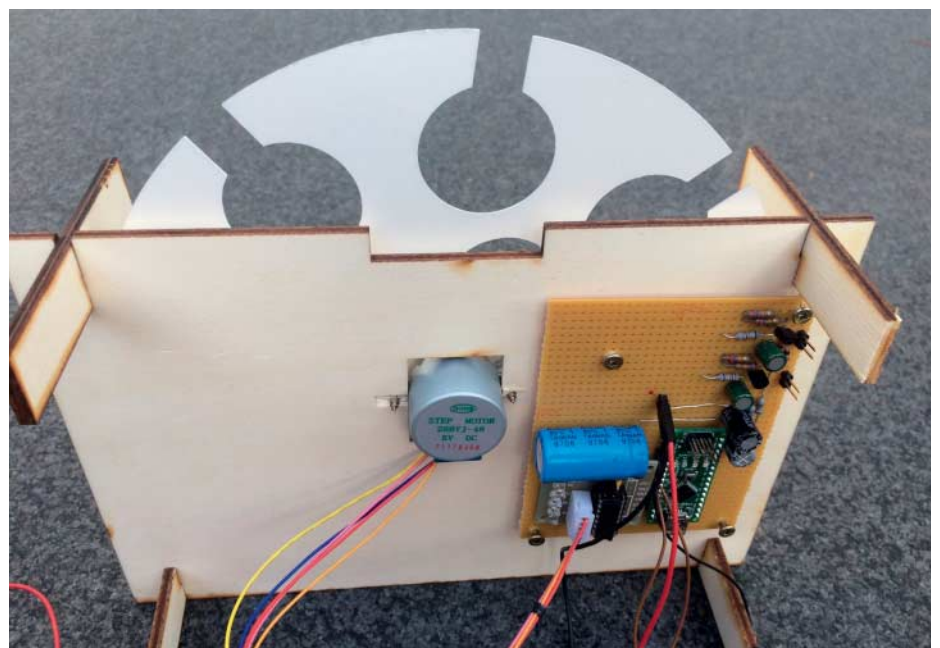
Material

- » 4 × 100Ω Widerstand
- » 4 × 47kΩ Widerstand
- » 1 × 470µF/16V Elektrolytkondensator
- » 1 × 4700µF/16V Elektrolytkondensator
- » 2 × 100µF/6,3V Elektrolytkondensator
- » 4 × 1N4448 Diode oder Vergleichstyp
- » 2 × BC547 Transistor NPN oder Vergleichstyp
- » 2 × Taster
- » 2 × Spulen 0,1mm-Kupferdraht auf einem Spulenkörper Ø25mm, etwa Reichelt VIS SP 5000
- » Lochrasterplatine 10cm × 8cm
- » Arduino Nano
- » 2 × Sperrholzplatten DIN A4, 3–4mm
- » Siebdruckkarton DIN A4, 1mm
- » Steckernetzteil 7–10V, 700mA
- » DC-Hohlbuchse passend zum Netzteil
- » Stepper-Motor mit Motortreiber 5V 28BYJ-48 und ULN2003
- » Neodym-Rundmagnet Ø 5,5mm, Länge 12mm o. Ä.
- » Glashaube etwa 45cm hoch, Ø 23cm
- » Holzkugel Ø 20mm
- » Furnierholz ca. 280cm × 28cm

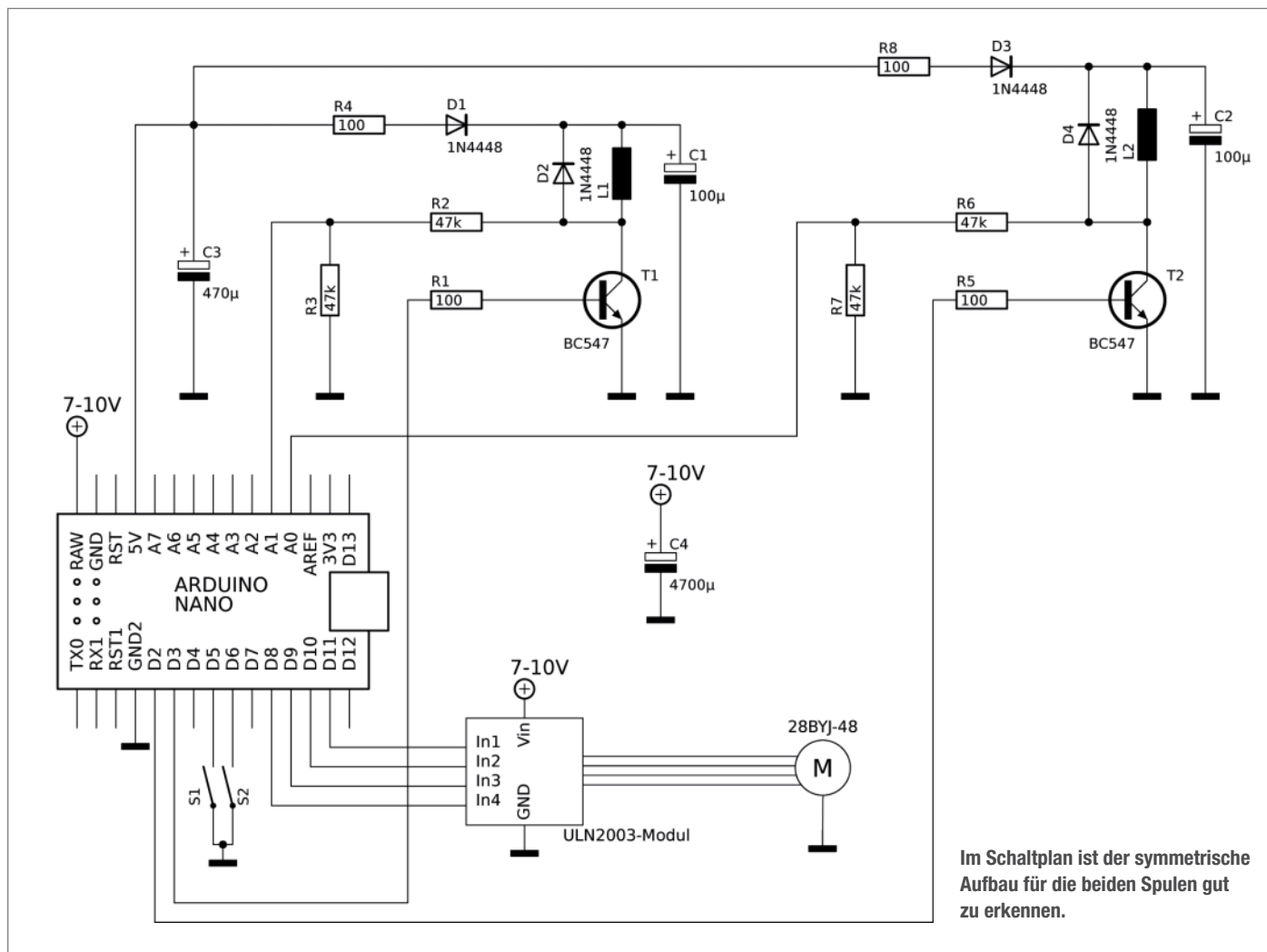
Mehr zum Thema

- » Carsten Meyer, Spulen, Make 3/16, S. 70
- » Clemens Verstappen, Handgestrickte Lochrasterplatten, Make 1/18, S. 106
- » Ulrich Schmerold, Perpetuisten, Make Sonderheft 2018, S. 112
- » Ulrich Schmerold, Der Zeitlupenrahmen, Make 1/18, S. 58

Alles zum Artikel
im Web unter
make-magazin.de/x5ne



Montage der Teile für die Lochscheibe und aufgebaute Elektronik



induktion aufgrund der Überquerung des Magneten an der Spule L1 oder L2 auftritt. Ist das der Fall, schaltet der Arduino den Transistor T1 oder T2, und der Kondensator C1 oder C2 entlädt sich über eine der Spulen, um das magnetische Pendel anzutreiben. D2 und D4 sind Freilaufdioden: Sie verhindern beim Abschalten der Spulen Spannungsspitzen. Die Spannungsversorgung der Pendelelektronik wird vom Arduino gespeist.

Software

Beim Start der Software sollte das Pendel nicht schwingen, da in `setup()` zuerst das Spannungspotenzial der Spulen im Ruhezustand des Pendels gemessen wird. Nach Anstoßen des Pendels wird die Schwingungsdauer `Period` gemessen, indem man die Zeit zwischen dem (n)-ten, und dem (n+3)-ten Durchgang an einer der beiden Spulen misst. In der Hauptroutine `loop()` wird nun per Abfrage permanent die Spulenspannung mit der Ruhespannung `MeanCoilVoltage` verglichen. Wird die Ruhe-

spannung unterschritten, erkennt die Software ein Überstreichen der entsprechenden Spule.

Die Software kann nach jedem erkannten Ereignis die aktuelle Position des Pendels bestimmen. Es werden vier wichtige Positionen unterschieden: zwischen den zwei Spulen (Innendurchgang), außerhalb einer Spule (Außendurchgang A oder B), Transit (tiefster Punkt des Pendels, beim Passieren der Lochscheibe). Um festzustellen, wo sich das Pendel befindet, misst man die Zeit zwischen dem (n)-ten und dem (n+1)-ten Durchgang an einer Spule und vergleicht diese Zeit mit der halben Periodendauer. Ein Außendurchgang ist kürzer als der Innendurchgang. Der Transitzeitpunkt ist nur indirekt messbar. Er findet immer kurz nach der Detektion eines Innendurchgangs statt.

Ein Maß für die Auslenkung des Pendels ist die Zeit, die ein Innendurchgang (`Passage`) dauert. Ändert sich diese Zeit nicht mehr, ist das Pendel eingeschwungen und das System läuft so stabil, dass die Lochscheibe gestartet werden kann. Sobald die

erste der beiden Spulen einen Innendurchgang detektiert, kann anhand der Variablen `Passage` die Zeit bis zum erwarteten Transit an der Lochscheibe berechnet werden. Sie findet im eingeschwungenen Zustand in etwa der Hälfte der Zeit eines kompletten Innendurchgangs statt. Die Spulen werden mit dem Pendeltimer aktiviert: Nach der Detektion eines Außendurchgangs an einer Spule wird die jeweils andere Spule für eine Zeit `COIL_ON_TIME` eingeschaltet, um das Pendel zurückzutreiben. Anschließend wird für die Zeit `COIL_RELAX_TIME` verhindert, dass die gerade bestromte Spule als Sensor verwendet wird.

Mechanischer Aufbau

Die Pendellänge bei der verwendeten Glasglocke beträgt etwa 43cm. Ein längeres Pendel schwingt langsamer, was unserer Regelung entgegenkommt. Bevor jedoch mit der später genutzten Glasglocke gearbeitet wird, sollte der Magnet zuerst mittels Bindfaden an einem Stativ befestigt und

Make_infiniti_pendulum_23.ino

```

if (CoilAValue < (MeanCoilA - 8)) // Magnet ueber Coila
{
  delta1 = CurrentMillis - LastTouchA;
  LastTouchA = CurrentMillis;
  if(delta1 > (Period>>1)) // Aussendurchgang
  {
    digitalWrite(CoilB, HIGH);
    PendelTimerStart = CurrentMillis;
    Passage = (LastTouchA - LastTouchB)>>1;
  }
  else // delta1 < (Period >> 1) // Innendurchgang
  {
    if (MotorControl == true)
    {
      MotorControl = false;
      if (MotorRunning)
      {
        if (MotorPosition > ComparePosition)
        {
          MotorTimerValue = MotorTimerValue + 1;
          Timer1.setPeriod(MotorTimerValue>>2);
        }
        if (MotorPosition < ComparePosition)
        {
          stepper();
          MotorTimerValue = MotorTimerValue - 1;
          Timer1.setPeriod(MotorTimerValue>>2);
        }
      }
    }
  }
}
}
}
}

```

Softwareauszug für den Außendurchgang (Pendel antreiben) bzw. Innendurchgang (Drehzahlregelung)

Anzeige

beide Spulen auf den Boden gelegt werden. So kann man Hard- und Software einfacher testen.

Zuerst wird die Orientierung der Spulen an die Nord-Süd-Ausrichtung des Magneten angepasst. Das Pendel schwingt nur dann stabil, wenn die Spulen den Pendelmagneten bei Aktivierung anziehen. Mittels des Sketches Coil_detection können sowohl die Spulen identifiziert als auch ihre Orientierung auf anziehend gedreht werden.

Jetzt legt man beide Spulen in einem Abstand von ungefähr 4,5cm nebeneinander und positioniert das Stativ so, dass der Pendelmagnet sich exakt zwischen beiden Spulen befindet und die Länge des Pendels ungefähr der Höhe der Glasglocke entspricht. Testen Sie, ob das Pendel funktioniert, indem Sie im Sketch vor Infiniti_Pendulum #define_Plotter die Kommentarzeichen entfernen.

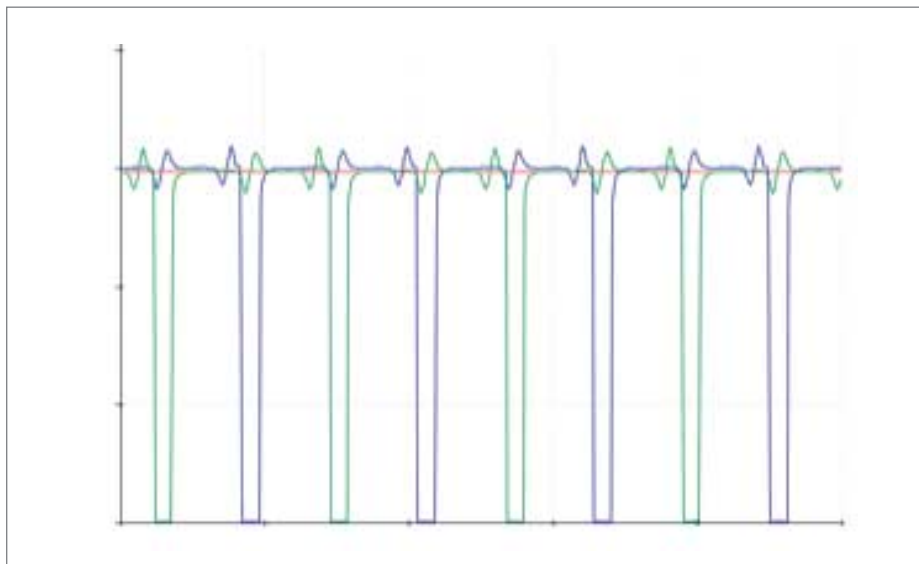
In der IDE wählen Sie *Werkzeuge/Serieller Plotter*, so dass Sie den Spannungsverlauf an den Spulen im Verlauf der Pendelbewegung ähnlich wie bei einem Oszilloskop verfolgen können. Wichtig ist, dass beim Überschreiten des Magneten die induzierte Spannung an jeder Spule zuerst abfällt,

bevor sie beim Verlassen der Spule wieder ansteigt. Ist das nicht der Fall, ist die entsprechende Spule verdreht.

Die Lochscheibe

Jetzt kann mit der Konstruktion der Schlitzplatte begonnen werden. Dazu werden sowohl die Sperrholzgrundplatte sowie zwei Furnier-Halbkreise mit einem Lasercutter ausgeschnitten. Die Sperrholzgrundplatte wird mit Holzleim mit den beiden Furnier-Halbkreisen verklebt. Die Maserung der Furnierhalbkreise sollte parallel zum Schlitz erfolgen, so wird die Verbindungsstelle gut kaschiert. Die Maserung der Sperrholzplatte sollte hingegen senkrecht zum Schlitz verlaufen, damit sich die Schlitzplatte nach dem Verleimen nicht verzieht.

Um einen störungsfreien Durchgang des Pendels durch die Schlitz der Lochscheibe zu gewährleisten, ist es notwendig, jederzeit die aktuelle Position des Lochrades zu kennen. Ebenfalls ist es wichtig, die Drehzahl möglichst genau regeln zu können. Ideal für einen solchen Einsatz sind Schrittmotoren. Ist die Anfangsposition bekannt, kann die Ansteuerungssoftware die Position des Motors



In der Arduino-IDE überprüfen Sie, ob die Spulen richtig ausgerichtet sind. So in etwa soll die Ausgabe aussehen (siehe Text).

mitzählen. Meine Wahl fiel auf einen Schrittmotor mit 4096 Schritten pro Umdrehung – so kann die Software binär und damit zeitsparend rechnen. Allerdings gibt es aufgrund des Getriebes, der in den benutzten Motoren verbaut ist, auch ein erhöhtes Spiel, was die Genauigkeit der ganzen Apparatur begrenzt.

Der Motor wird mit der mitgelieferten Ansterelektronik mit dem Arduino verbunden. Der verwendete Motor erzeugt starke Störungen auf der Versorgungsspannung, weshalb der Kondensator C4 nahe an der Versorgungsspannung der Ansterelektronik platziert wird. Da die Leuchtdioden der

Ansterelektronik später durch die Schlitzplatte leuchten würden, kann man sie (oder ihre Vorwiderstände) auslöten. Die Pfostenstecker habe ich ebenfalls ausgelötet und rückwärts wieder eingelötet – so lässt sich die Ansterelektronik gut mit der Lochrasterplatte verbinden.

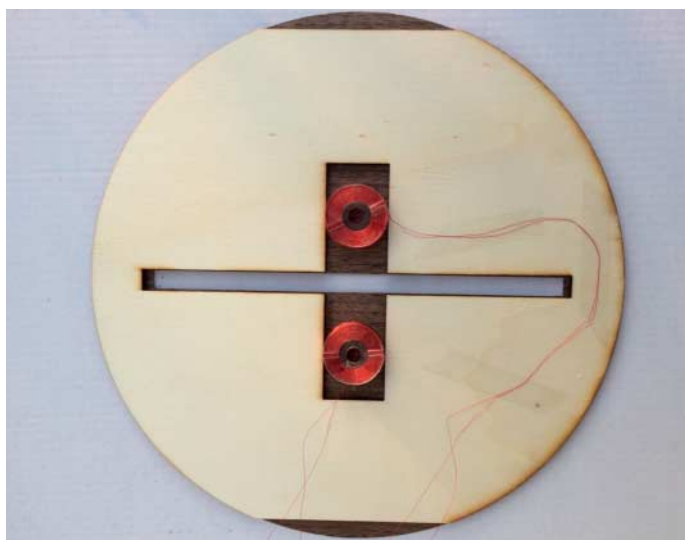
Die Drehzahl des Motors wird mit dem Mikrosekunden-Timer `TimerOne` und der Routine `Stepper()` kontrolliert, der vorgibt, alle wie viel Mikrosekunden der Motor einen Schritt weiter drehen soll. Bei 8 Löchern in der Scheibe benötigt der verwendete Motor von Loch zu Loch 512 Schritte. Aus der Periodendauer des Pendels wird ein passender

Wert für die Drehzahl des Motors berechnet:
 $\text{MotorTimerValue} = \text{Period} \cdot 1000 / 1024 \cdot 1024$
 entspricht 512×2 , da das Pendel während einer Periode zwei mal durch die Lochscheibe schwingt. Diese Rechnung lässt sich in C mit einem Rechtsshift (`>>`) sehr schnell berechnen, 1024 entspricht nämlich 2^{10} :
 $\text{MotorTimerValue} = (\text{Period} \cdot 1000) > 10$.

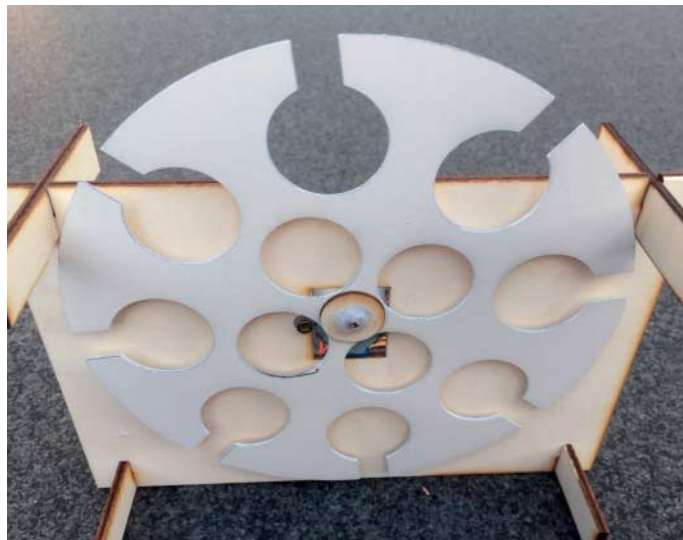
Die Motoraufhängung wird aus Sperrholz ausgeschnitten und der Motor mit kleinen Holzschrauben fixiert. Die Lochscheibe wird ebenfalls mittels Lasercutter erstellt und muss plan und formstabil und leicht sein, weshalb Siebdruckkarton zum Einsatz kommt. Beim Montieren der Lochscheibe auf die Achse des Motors ist darauf zu achten, dass die Scheibe nirgends schleift und nicht eiert. Dazu startet man den Sketch Motortest, der den Motor langsam drehen lässt. Unter Umständen muss der Motor durch Zurechtbiegen der Befestigungsfahnen noch ausgerichtet werden. Gegen zu viel Spiel helfen die Verstärkungsscheiben, die in der Vorlagendatei enthalten sind (Downloads siehe Link in der Kurzinfor). Ich habe die Scheibe bei laufendem Motor ausgerichtet und sie anschließend, ebenfalls bei laufendem Motor, mit Heißkleber vorsichtig auf der Motorachse fixiert.

Synchronisierung von Pendel und Motor

Aufgrund unterschiedlicher mechanischer und elektrischer Gegebenheiten, müssen in der Software einige Parameter angepasst werden, die nur empirisch ermittelt werden können. Das Pendel wird zunächst nicht durch die Lochscheibe geschickt, sondern



Die Spulen werden mittig mit gleichem Abstand zum Schlitz auf die Unterseite des Furniers in die entsprechende Aussparung geklebt. Durch die Aussparung sitzen die Spulen später möglichst dicht unter dem Pendel.



Die inneren Löcher der Lochscheibe dienen der Gewichtseinsparung, und man kommt zudem nach der Montage noch an die Befestigungsschrauben des Motors heran.

Lochscheibe und Pendel agieren erst einmal unabhängig voneinander. Dennoch lässt sich die Motorregelung auf diese Weise überprüfen und feintunen.

Der Motor der Lochscheibe ist inaktiv, bis das Pendel stabil schwingt. Als Maß hierzu dient die Zeit, die ein Innendurchgang (startend bei Spule A) benötigt (Passage). Dazu wird die Innendurchgangszeit gleitend gemittelt. Da der Zielwert der Innendurchgangszeit nicht bekannt ist, wird anschließend die Varianz der Innendurchgangszeiten gleitend gemittelt (VarianzPassage). Fällt die Varianz unter einen vorgegebenen Wert, ist das Pendel eingeschwungen.

Der Start des Motors muss exakt im Transit des Pendels erfolgen. Da hierfür kein Sensor vorgesehen ist, muss jeweils der Startzeitpunkt eines Innendurchgangs (an Spule A) als Maß herangezogen werden. Zu diesem Zeitpunkt muss die halbe Innendurchgangszeit (plus ein später zu ermittelnder Korrekturfaktor `COIL_MIDDLE_COMPENSATION`) addiert werden, dann startet der Motor. Wenn der Motor startet, kennen wir dessen Motorposition (`MOTOR_START_POS/Motorposition`). Für die Regelung interessiert uns allerdings die Motorposition,

Make_infiniti_pendulum_23.ino

```
if (MotorTimerValue == 0)
    MotorTimerValue = ((long)Period * 2000)/(512 + PauseAtTransit - 1);
else
    MotorTimerValue = MotorTimerValue << 2;

Timer1.initialize(MotorTimerValue >> 2);
#ifdef Monitor
    Serial.println("Periode MotorTimerValue");
    Serial.print(Period);
    Serial.print(",");
    Serial.println(MotorTimerValue>>2);
    Serial.println("");
    Serial.println("PassageTime Varianz");
#endif
```

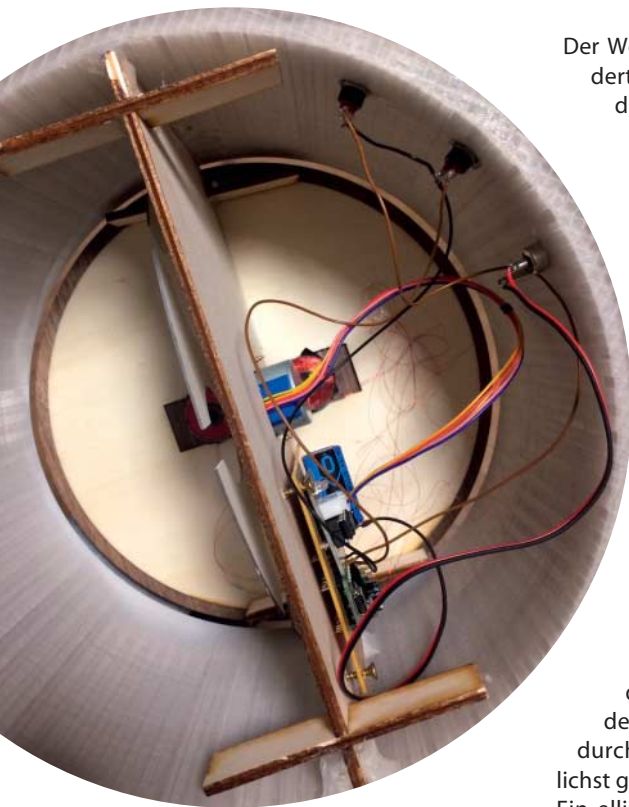
Ausgabe der Einschwingphase

die der Motor zum Startzeitpunkt des Innendurchgangs (`ComparePosition`) gehabt hätte (wäre er da schon gelaufen). Diese fiktive Position liegt in der Vergangenheit, wir müssen von Motorposition also etwas abziehen:

```
ComparePosition = Motorposition -
(1024/Period) * Passage
```

Es ist bekannt, dass 1024 Motorschritte für eine Pendelperiode `Period` benötigt werden. Dieser Quotient wird daher mit der gemessenen Zeit multipliziert, die zwischen Innendurchgang und Transit liegt und man erhält den Zielwert `ComparePosition` für die Motorregelung für jeden weiteren Innendurchgang an Spule A. Da die Software die Motorposition mitzählt, erkennt das Programm, ob

Anzeige



In den Sockel wird von oben die Motor-aufhängung mit verklebter Grundplatte geschoben.

das Loch früher oder später am Innendurchgang ankommt und erhöht – oder verringert die Motordrehzahl durch Variation des `MotorTimeValue`. Zur Überprüfung der Regelung kann man in der Software `#define Monitor` aktivieren (alles andere deaktivieren) und das Zusammenspiel zwischen Lochscheibe und Pendel im Monitor der Arduino-Entwicklungsumgebung anschauen.

Das Programm wird bei ruhendem Pendel gestartet, damit die Software die Spulenruhespannung messen kann. Anschließend wird das Pendel angestoßen. Im Programmteil `setup()` wird zuerst die Pendelperiode und die daraus resultierende initiale Motordrehzahl ausgegeben. Jetzt kann man verfolgen, wie sich das Pendel einschwingt. Wird der Wert von `Varianz-Passagen` unterschritten, startet der Motor. Die angezeigten Größen sind nun die jeweils gemessene Motorposition und die aktuelle Motordrehzahl `MotorTimeValue`. Je exakter der Wert zum Start des Motors ist, desto schneller wird der eingeschwungene Zustand erreicht. Daher gibt es in der Software die Möglichkeit, alternativ zur Bestimmung in der Routine `setup()` einen langfristig gemittelten Wert zur Initialisierung von `MOTOR_TIMER_VALUE` zu verwenden.

Der Wert `MOTOR_START_POS` wird so geändert, dass `ComparePosition` ungefähr den Wert 250 annimmt. Motor und Pendel können gestoppt werden, indem beide Taster zugleich gedrückt werden, die Lochscheibe wird dann wieder in der Anfangsposition stehenbleiben.

Wenn das System so weit funktionstüchtig ist, dass die Regelung nach Start des Motors schnell und stabil einsetzt, können alle Komponenten zusammengefügt werden, um zu überprüfen, ob das Pendel auch tatsächlich passgenau das jeweilige Loch trifft. Dazu verbindet man die Schlitzeplatte mit der Motorhalterung. Nach Start des Programms wird mit den beiden Tasten die Lochscheibe so gedreht, dass sich ein Loch genau oben befindet. Zum Schluss wird das Pendel durch die Lücke hindurch in einer möglichst geradlinigen Bewegung angeschubst. Ein elliptisch schwingendes Pendel benötigt mehr Platz als ein geradliniges Pendel, aber die Regelung des Pendels korrigiert dies im Laufe der Zeit.

Im Prinzip sollte jetzt schon alles funktionieren, leider triggern die Spulen aber nicht immer exakt in der Spulenmitte, so dass der Korrekturfaktor `COIL_MIDDLE_COMPENSATION` zum synchronen Motorstart variiert werden muss. Ist der Korrekturfaktor zu klein, bleibt das Pendel schon am ersten Loch hängen. Ist der Korrekturfaktor zu groß, eckt das Pendel beim Zurückschwingen an der Hinterseite des nächsten Lochs an. Es kann etwas dauern, bis der ideale Wert für `COIL_MIDDLE_COMPENSATION` gefunden ist, da wir bei jedem Versuch warten müssen, bis das Pendel eingeschwungen ist. Ein Notaus in der Software stoppt die Lochscheibe, wenn das Pendel nicht wie erwartet die Spulen abwechselnd überstreicht. An D13 (LED) können Sie ein Stroboskop anschlie-

ßen, um die genaue Position von Pendel und Lochscheibe im Transit zu erkennen.

Veredelung

Eine passende Glasglocke findet sich im Internet. Mithilfe von transluzentem 2K-Kleber wird möglichst mittig an der inneren Oberseite der Glasglocke ein Faden angeklebt. Für das Pendel habe ich eine massive Holzkugel mit Durchmesser 20mm verwendet. Wichtig ist, dass in der Ruhestellung der Faden mittig in einem Loch der Lochscheibe hängt und die Kugel möglichst flach über die Schlitzeplatte pendelt. Der Magnet und der Faden werden ebenfalls mit Kleber an der Kugel befestigt.

Für den Sockel gibt es eine 3D-Drucker-Vorlage mit Führungsnut für die Motoraufhängung. Die andere Seite der Aufhängung wird so eingeklebt, dass die Lochscheibe und Grundplatte mittig im Sockel liegt. Anschließend noch die Taster und die DC-Buchse verkabeln. Ein schönes Äußeres erhält der Sockel, indem man ihn mit Furnier beklebt. Um bei aufgesetzter Glasglocke das Pendel anzustoßen, versetzt man das ganze Objekt zum Start einfach ruckartig.

Variationen

Einen Autostart des Pendels hatte ich versucht zu implementieren, scheiterte aber an der Chaostheorie: Schon minimale Änderungen der Startbedingungen können zu sehr unterschiedlichen Ergebnissen führen. Vielleicht hat einer der Leser hierzu eine zielführende Idee. Statt die Lochscheibe einfach nur konstant in eine Richtung drehen zu lassen, könnte man die Scheibe vor- und zurückdrehen lassen, Zufallszahlengenerator-gesteuert mal pausierend oder gar mit der doppelten Geschwindigkeit drehen. Wählt man einen Schrittmotor ohne Getriebe, ist die Positioniergenauigkeit deutlich erhöht. Dadurch sollten sich die Löcher der Lochscheibe noch deutlich verkleinern lassen. —fls

Make_infiniti_pendulum_23.ino

```
// Die empirisch zu ermittelnden Parameter:
#define MOTOR_START_POS 360 // so wählen, dass ComparePosition ~250 ist
#define COIL_MIDDLE_COMPENSATION 85 // Für das reibungsfreie Zusammenspiel anpassen
#define MOTOR_TIMER_VALUE 0 // Wenn im "Monitor" stabil: Wert hier eintragen

// Debug Möglichkeiten für Pendel (Plotter) und die Regelung (Monitor)
// jeweils nur einen auswählen!
// #define Monitor // Alle Parameter der Pendel-Motor Regelung anzeigen
// #define Plotter // Plot-Ausgabe für das Pendel
```