

```
ri = RoboInterface(bEnableDist  
thread.start_new_thread(light_  
while True:  
    # normally forward  
    ri.setSpeed(SPEED, SPEED)  
    # forward until we see a  
    while ri.GetD1() > DIST:  
        sleep(0.1)  
    # obstacly ahead, stop  
    ri.setSpeed(0, 0)  
    # turn left  
    ri.setSpeed(-SPEED TURN,
```

Editorial

Gefährdete Spezies

Dirk Fox, Stefan Falk

Dass die demographische Entwicklung in Deutschland eines Tages zu einem Mangel an Fachkräften führen wird, ist nicht gerade eine neue Erkenntnis. Denn die [Geburtenrate](#) macht schon seit den frühen 70er Jahren aus der Bevölkerungspyramide einen Tannenbaum.

Inzwischen ist aus der Befürchtung ein Faktum geworden. Das Institut der deutschen Wirtschaft in Köln ([IW](#)) sieht Ingenieure bereits als „gefährdete Spezies“. Zwar arbeiten in Deutschland 1,6 Mio. davon, und den fünf Branchen mit der höchsten Ingenieursdichte verdanken wir fast die Hälfte der Einnahmen aus dem Außenhandel. Zu denken gibt aber die Altersstruktur.

Auf 100 erwerbstätige Ingenieure im Alter von 55 und mehr Jahren kommen derzeit in Deutschland lediglich 87, die jünger sind als 35 Jahre. Andere Länder sind besser gerüstet: Die Schweiz verfügt über 38 % mehr junge als alte Ingenieure, in Frankreich sind es 248 %, in Schweden 297 % und in Irland sogar 447 % [1].

Der MINT¹-Frühjahrsreport 2012 des IW [2] zeigt die Dimension des Problems: Die Fachkräfte-Lücke (offene Stellen minus Arbeitssuchende) erreichte im April 2012 mit 210.000 einen neuen Höchststand. Obendrein kommen Jahr für Jahr knapp 110.000 neu zu besetzende Stellen hinzu.

Trotz einer Verdoppelung der MINT-Studienanfänger seit dem Jahr 2000 verlassen aber lediglich 98.000 Absolventen jährlich unsere Hochschulen – die Bedarfs-lücke wird also weiter wachsen.

Immerhin steigt offenbar die deutsche MINT-Kompetenz, wie die PISA²-Studien belegen. Lagen deutsche Schüler im Jahr 2000 im naturwissenschaftlichen Bereich noch bei durchschnittlich 487 Punkten, waren es 2009 bereits 520. Aber auch dieser Wert ist deutlich ausbaufähig: in Finnland erreichten die Schüler 554, und in Shanghai sogar 575 Punkte [3].

Vielleicht spielen chinesische und finnische und Kinder einfach häufiger mit fischertechnik?

Beste Grüße,
Euer ft:pedia-Team

- [1] IW: [Ingenieure: Gefährdete Spezies](#). 26.04.2012.
- [2] IW: [MINT-Frühjahrsreport 2012](#). 23.05.2012.
- [3] Wikipedia: [Ergebnis der PISA-Studie 2009](#). (Abruf am 26.06.2012)

P.S.: Am einfachsten erreicht ihr uns unter ftpedia@ftcommunity.de oder über die Rubrik *ft:pedia* im [Forum](#) der ft-Community.

¹ MINT: Mathematik – Informatik – Naturwissenschaften – Technik.

² PISA: *Programme for International Student Assessment* (Programm zur internationalen Schülerbewertung): Seit dem Jahr 2000 werden in den OECD-Staaten alle drei Jahre die Lesekompetenz sowie die mathematischen und naturwissenschaftlichen Kenntnisse 15jähriger Schüler untersucht.

Inhalt

Gefährdete Spezies.....	2
Flugzeugfahrwerke	4
Vom Zählen und Abzählen (2).....	11
ft-Spezialteile made by TST (Teil 1).....	24
Bootsbau mit fischertechnik.....	26
The fischertechnik Interface for the Rest of us	32
HP-GL-Plotter (Teil 3).....	39

Termine

Was?	Wann?	Wo?
Fan Club Tag	08.07.2012	Tumlingen
Convention 2012	29.09.2012	Erbes-Büdesheim
Modellschau Münster	18.11.2012	Handwerkskammer Bildungszentrum (MS)
Sonderverkauf Salzstetten	01.12.2012	fischertechnik GmbH, Salzstetten
Sonderverkauf Dortmund	08.12.2012	Bildungszentrum Hansemann (DO)

Hinweise

Der fischertechnik-Kasten „[Pneumatik 3](#)“ ist für das „Goldene Schaukelpferd 2012“ (Kategorie 2) nominiert. Die [Abstimmung](#) läuft noch bis **03.07.2012**.

Impressum

<http://www.ftcommunity.de/ftpedia>

Herausgeber: Dirk Fox, Ettlinger Straße 12-14,
76137 Karlsruhe und Stefan Falk, Siemensstraße 20,
76275 Ettlingen

Autoren: Erik Andresen, Stefan Falk (steffalk), Dirk Fox (Dirk Fox), Harald Steinhaus (Harald), Andreas Tacke (TST), Michael Tilli (fish)

Copyright: Jede unentgeltliche Verbreitung der unveränderten und vollständigen Ausgabe sowie einzelner Beiträge (mit vollständiger Quellenangabe: Autor, Ausgabe, Seitenangabe ft:pedia) ist nicht nur zulässig, sondern ausdrücklich erwünscht. Die Verwertungsrechte aller in ft:pedia veröffentlichten Beiträge liegen bei den jeweiligen Autoren.

Flugzeugtechnik

Flugzeugfahrwerke

Harald Steinhaus

Fischertechnik kann nicht fliegen – aber fahren. Was wäre aber auch ein Flugzeug ohne Fahrwerk? Eine besondere Faszination geht von diesen technischen Meisterwerken aus – denn sie sollen einerseits ein Flugzeug tragen und andererseits bei Bedarf elegant im Rumpf verschwinden. Genau die richtige Herausforderung für ein mechanisches ft-Funktionsmodell.

Ein Flugzeugfahrwerk muss das Gewicht des Fliegers nebst den bei Start und Landung auftretenden Kräften ertragen und übertragen können. Bei schnellen Maschinen muss es außerdem bei Nichtgebrauch aus der Luftströmung verschwinden.

Ein Fahrwerk kann natürlich auf allem beruhen, was die Physik in Hydraulik, Pneumatik und Mechanik zu bieten hat. Aus verschiedenen Gründen dominiert die Mechanik, und darin wiederum liegt der Schwerpunkt bei den Hebelgesetzen und den [Koppel- oder Viergelenkgetrieben](#). Die Bandbreite wird ein wenig angedeutet durch die [Fahrwerksmodelle](#) im Bilderpool der ftcommunity.

Fahrwerke können simpel und einfach aufgebaut sein oder recht komplex werden. Unabhängig vom mechanischen Funktionsprinzip möchte ich im Folgenden sieben Ausbaustufen auf dem Weg zum „ultimativen“ Flugzeugfahrwerk unterscheiden.

Ausbaustufe 1

Stufe 1 kennt man aus der Flugschule, der Serengeti oder dem Australischen Outback: Das Fahrwerk trägt das Gewicht des Fliegers, plus die Kräfte bei Start und Landung. Punkt. Da stehen also die Fahrwerksbeine samt Rädern starr aus der Zelle heraus, und fertig.

Ausbaustufe 2

In Stufe 2 soll das Fahrwerk im Flug eingezogen und zum Landen ausgefahren werden. Das kommt der Aerodynamik zugute; außerdem ist das Fahrwerk besser gegen Vereisung geschützt, wenn es im Rumpfinnenen transportiert wird. In ausgefahrenem Zustand muss es gegen unbeabsichtigtes Einklappen unter Last verriegelt werden. Da kann man z. B. Riegel und Bolzen einsetzen, die aber wieder eigenständige Antriebe brauchen und selber auch verriegelt werden müssen. Der einfachste Weg ist hier das *überstreckte Kniegelenk* (Abb. 1).



Abb. 1: Ein überstrecktes Kniegelenk.

Vorbild ist das menschliche Knie, das im gestreckten Zustand ein längeres Stehen ermöglicht, ohne dass der Streckmuskel im Oberschenkel dazu gespannt sein muss. Im

vermeintlich einfachsten Fall baut man so ein Knie direkt nach. Nur ist dieser Aufbau durchaus nicht einfach, denn es gibt auch Nachteile: zum Einfahren müssen beide Teile, Ober- und Unterschenkel, angetrieben werden, und beide Teile müssen die volle Last tragen können.

Ausbaustufe 3

In Stufe 3 besteht das Fahrwerk aus einem in sich starren Stab (dem *Fahrwerksbein*), der an einem Ende drehbar gelagert ist und am anderen Ende die Räder hat. Nur dieses Bein muss auf volle Last ausgelegt sein. Ein zweiteiliges Kniegelenk verriegelt es im ausgefahrenen Zustand. Die Geometrie besteht aus vier Teilen: Oberschenkel, Unterschenkel, Fahrwerksbein (der Teil zwischen der Verbindung zum Rumpf und dem Anschluss des Unterschenkels) und schließlich dem starren Rumpf zwischen Fahrwerksbein und Oberschenkel (Abb. 2).

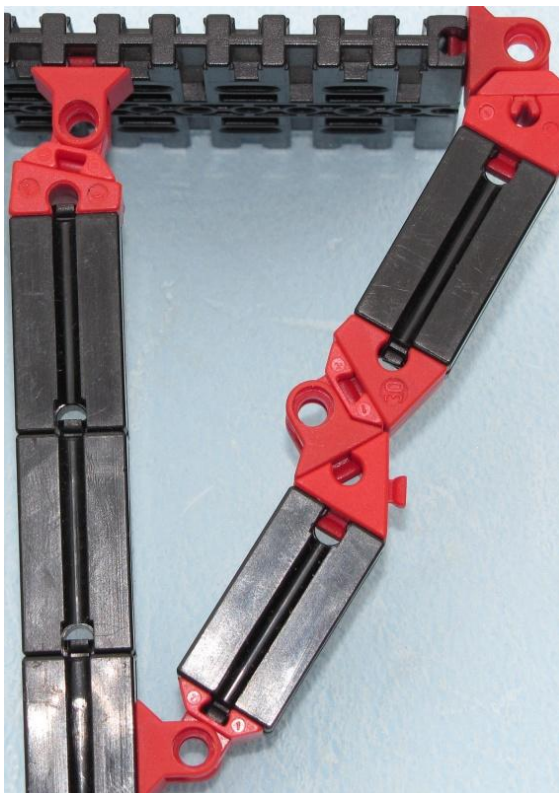


Abb. 2: ein überstrecktes Kniegelenk als Verriegelung des eigentlichen Fahrwerksbeins

Diese Anordnung ist als Koppel- oder Viergelenkgetriebe bekannt. Im Maschinenbau muss ein „Getriebe“ keine Zahnräder enthalten: im technischen Sinne besteht ein Getriebe aus irgendwelchen antreibenden und irgendwelchen angetriebenen Bauteilen von beliebigem Aussehen. Das Viergelenkgetriebe kennt eine Vielzahl von Unterscheidungen, denn je nach Zusammensetzung der Längen der Glieder können einige davon komplette Drehungen um einen Endpunkt ausführen oder das ganze Gebilde bewegt sich überhaupt nicht. Hier sei auf die vertiefte Betrachtung in der fischertechnik-hobby-Reihe verwiesen [1].

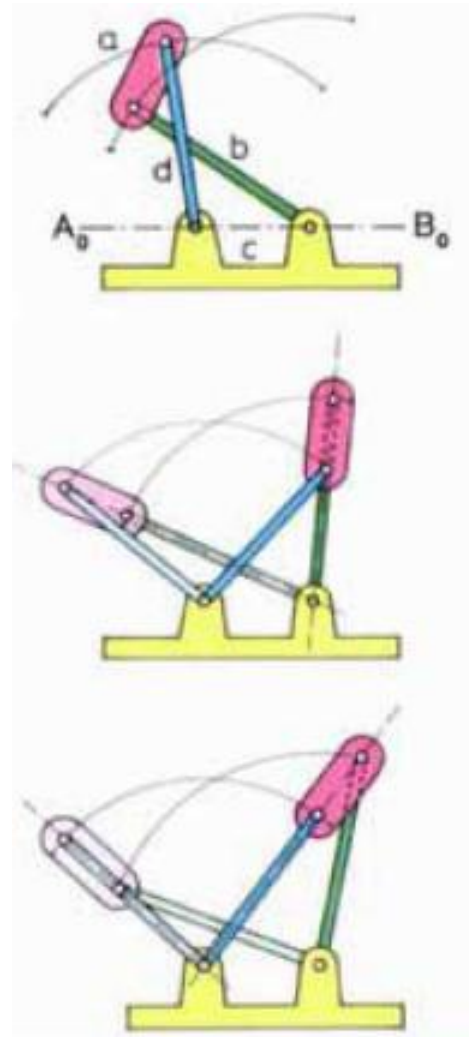


Abb. 3: Viergelenkgetriebe (aus: [1], S. 8)

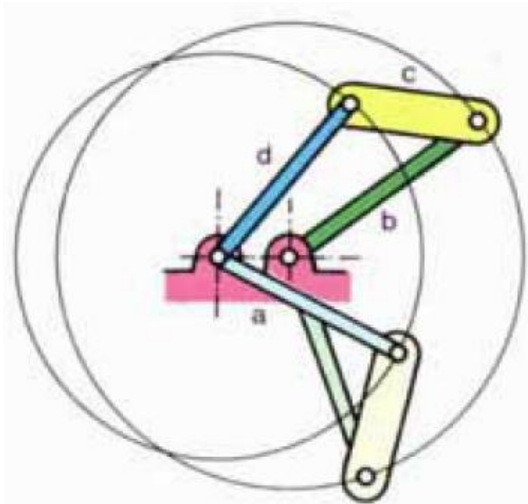


Abb. 4: Viergelenkgetriebe (aus: [1], S. 9)

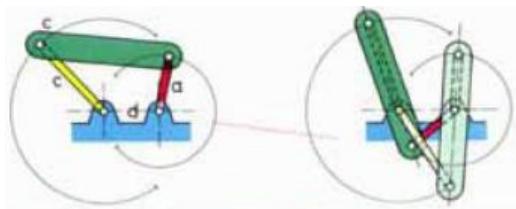


Abb. 5: Viergelenkgetriebe (aus: [1], S. 11)

Weitere Fundstellen und Orte der Inspiration sind z. B. die [Digitale Mechanismen- und Getriebelbibliothek](#) und die Webseite „[How round is our circle?](#)“ von John Brant und Chris Sangwin.



Abb. 6: „Knie im Knie“ beim A350. Man sieht, wer die größten Kräfte aushalten muss (Quelle: Wikipedia, Fotograf: Tangopaso).

Der (echte) [Airbus A350](#) verwendet im Hauptfahrwerk ebenfalls das Kniegelenk als Grundprinzip – allerdings mit den zusätzlichen Verfeinerungen, dass man hier (a) ein „Knie im Knie“ vorfindet, und (b) das Ganze gleich zweifach, in räumlich um 90° verdrehter Anordnung (Abb. 6). Damit ist das ein räumliches Koppelgetriebe, im

Gegensatz zu all den ebenen Vertretern, die hier erwähnt werden.

Wir kommen zurück zum einfachen, ebenen Kniegelenk: in Stufe 3 ist der Oberschenkel das einzige angetriebene Element. Der Antrieb kann über alle denkbaren Arten, also etwa Seilzug, P-Zylinder oder Zahnstange erfolgen. An dieser Stelle kommen aber zwei weitere Aspekte hinzu.

So findet man erstens in einem Flugzeug immer beengte Platzverhältnisse vor. Das Fahrwerk soll unterhalb des Rumpfbodens verschwinden (da soll aber auch Frachtraum oder Elektronik hin) oder in den Flügeln passen (der aber nicht sehr dick werden darf) oder in eigene (auch nicht zu große) Gondeln einfahren.

Zweitens sind Zahnstangen (ft-Hubgetriebe) und P-Zylinder sperrig und brauchen außerdem noch Raum zum Schwenken. Ein ft-P-Zylinder braucht einen geraden Einbauraum von mindestens 105 mm Länge, bei gerade einmal 30 mm Hubweg. Seilzüge sind vom Platzbedarf her sehr verlockend. Allerdings braucht man immer zwei davon, und man muss irgendwie verhindern, dass sie schlaff durchhängen und ihre Führungsbahn verlassen. Das geht am einfachsten, wenn man Seilzüge über runde Scheiben führt.

Ist man schon so weit gekommen, kann man die Seilscheiben auch gleich als Zahnräder ausführen, und landet schließlich bei Dreh-Antrieben als den am ehesten Platz sparenden Varianten.

In Zusammenfassung ergibt sich für ft-Modelle als zweckmäßigster Aufbau ein Viergelenkgetriebe, dessen einziges angetriebenes Glied entweder aus einem Zahnrad besteht (vgl. Abb. 7) oder mittels einer ft-Kurbel mit einem Drehantrieb verbunden ist. Wohlgermerkt – diese Aussage gilt für den ft-Modellmaßstab. Die echte Flugwelt wird klar von Hydraulikzylindern beherrscht, da sie im hoch skalierten Verhältnis deutlich kompakter da stehen und enor-

me Kräfte aufbringen können. Ein maßstabsgetreu hoch skaliertes ft-Z30 wäre im echten Flugzeugbau zu schwer, zu sperrig und hätte mit jedem seiner Zähne eine mögliche Ausfallursache, insbesondere bei Vereisungsgefahr.

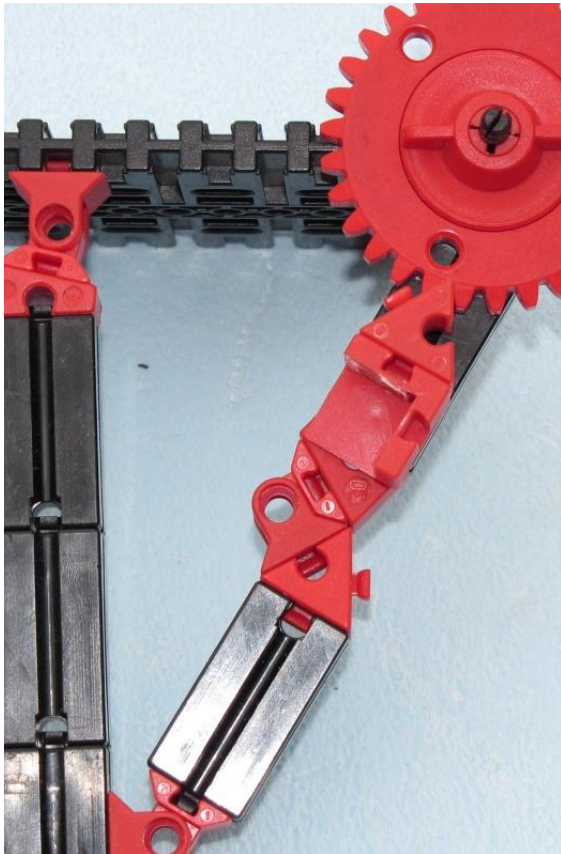


Abb. 7: Fahrwerksbein mit Knie-Verriegelung und Zahnradantrieb

Auf diesem Prinzip beruht das Modell [FWH12](#), das als Ausgangspunkt für viele weitere meiner Muster gedient hat. Die etwas zu groß geratene und dann [eingestellte A380](#) enthielt diesen Standard-Aufbau in allen dreien ihrer Fahrwerke, wenn auch in sehr unterschiedlichen Realisierungen.

Die Abbildungen 8, 9, 10, 11 zeigen den „Standard“-Aufbau einmal mit Z30 und einmal in Mini-Ausführung mit der Kurbel 38235, jeweils in aus- und eingefahrenem Zustand.

Vom Standard-Aufbau ausgehend, oder auf ganz anderen Wegen wandernd, gibt es

natürlich viele weitere Möglichkeiten. Angesichts der dünnen Angelschnüre aus der ft-Packung 48667 (Weihnachtsdekoration) ist das letzte Wort bezüglich der Seilzüge auch noch nicht gesprochen. Passt doch so eine Angelschnur ganz locker in eine ft-Vollnut, selbst wenn schon Bausteine mit Zapfen darin stecken.

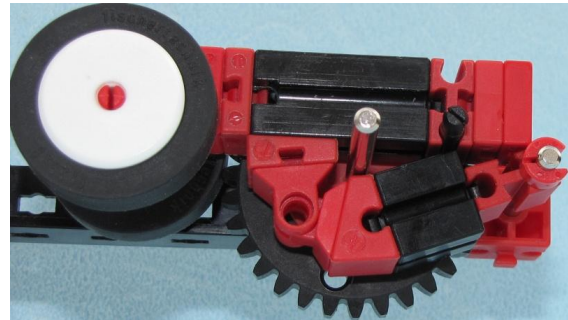


Abb. 8: Fahrwerk mit Antrieb über Drehachse (Z30), eingefahren ...



Abb. 9: ...und ausgefahren.

Weiterhin kann man da, wo ein Zahnrad ist, auch mal ein zweites einsetzen. Dieser Gedanke führte auf zwei kämmende Z30, die mit hinreichend vielen Verstreben

zu einem Klapperatismus verbunden wurden, der sich bei näherer Betrachtung als ... ein Fünfgelenk-Getriebe mit zwei gekoppelten Elementen herausstellt. Dazu später mehr.

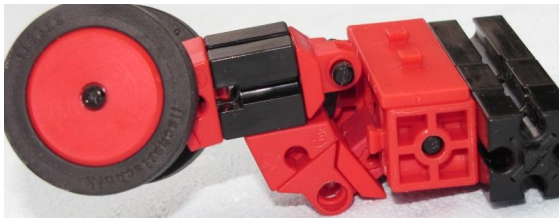


Abb. 10: Fahrwerk mit Antrieb über Drehachse (Kurbel), eingefahren ...



Abb. 11: ... und ausgefahren.

Ausbaustufe 4

In den wenigsten Fällen wird man an einem Viergelenk Freude haben, das ein zum vollen Umlauf fähiges Fahrwerksbein aufweist. Es müsste die „ausgefahrene“ Lage ja trotzdem gezielt anfahren und dort verriegeln können. Also baut man besser ein Getriebe mit zwei Endlagen und treibt es mit einem Motor an. Stufe 4 hat dafür

zu sorgen, dass bei Erreichen der Endlage entweder der Motor abgeschaltet wird oder der Antrieb „leer“ weiter läuft und keinen Schaden anrichtet. Der Leerlauf kann über eine Rutschkupplung erfolgen oder etwa dadurch, dass eine Zahnstange vom Antriebsrad herunter fährt und damit ausgekuppelt ist. Auch bei Leerlauf sollte natürlich irgendwann der Motor abgeschaltet werden. Es müssten also Endschalter für beide Endlagen eingebaut werden.

Ausbaustufe 5

Mit Stufe 4 hat ein Fahrwerk alles, was es zu seiner Funktion braucht. Alles Weitere ist „Sahnehäubchen“. In Stufe 5 ist das ein Landescheinwerfer, der mit dem Fahrwerk ausfährt und dabei eingeschaltet wird. Ein Beispiel hierfür ist das Fahrwerk der A340, bei der das Bugfahrwerk den Schalter betätigt, während sich die Scheinwerfer an den Hauptfahrwerksbeinen befinden.

Ausbaustufe 6

Stufe 6 enthält Fahrwerksluken oder -verkleidungen, die vom Fahrwerk selbst betätigt werden. Dahin kommt das Hauptfahrwerk [FWH13](#) der Antonov AN-124.

Ausbaustufe 7

Schließlich und ganz oben steht Stufe 7, in der Bug- und Hauptfahrwerk vom gleichen Motor angetrieben werden und mit Landescheinwerfer und Fahrwerksluken ausgestattet sind. Das war bei der [A340](#) dank gemeinsamem Antrieb mit Rutschkupplung knapp erfüllt, bis auf die nicht vorhandenen Fahrwerksluken – aber immerhin wurden Teile der Verkleidung mitgeschwenkt. Der Transporter hatte ein [Hauptfahrwerk mit voll integrierter Verkleidung](#) (dessen Antrieb nie richtig funktioniert hat), sowie ein separat angetriebenes Bugfahrwerk.

Mein neuester Anlauf in dieser Richtung ist bislang noch etwas „dünn“: eine Fahr-

werksluke, die komplett auf Leichtbau getrimmt ist. Das verwendete Material stammt (leider, leider) nicht von fischer-technik, ist aber gleichwohl gut erhältlich: es ist Moosgummi von 2 mm Stärke. Moosgummi gibt es in den ft-Farben, es lässt sich sehr einfach zuschneiden, und wenn man Löcher von 3 mm Durchmesser hinein stanzt (etwa per Lochzange), kann man es mit S-Riegeln befestigen.



Abb. 12: Moosgummi mit Lochzange



Abb. 13: Fahrwerksluke, mit Moosgummi beplankt

Fünfgelenkgetriebe

Mangels einer tief schürfenden theoretischen Ableitung liegen beim Fünfgelenk-Getriebe derzeit bei meinen Experimenten die „Prinzen“ (Erfolge) und die „Kröten“ (nicht verwertbare Entwürfe) dicht beieinander. Wenn man nicht starr auf Flugzeugbeine fixiert ist, findet man bei Variation der Geometrie auch interessante Dinge wie Parallelverschiebe-Mechanismen, Greifer und etliches mehr (siehe z. B. Abb. 14). Ein Beispiel für eine „Kröte“ zeigt Abb. 15.

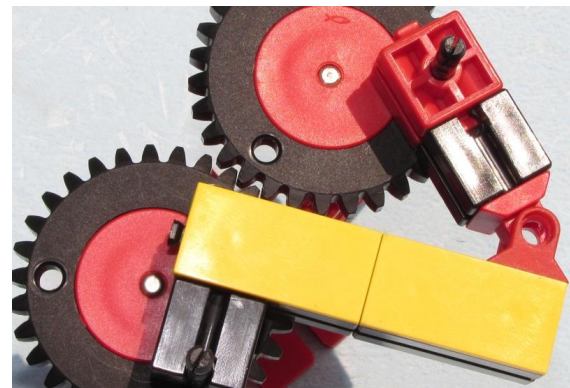


Abb. 14: Fünfgelenk mit Parallelverschiebung (der gelbe Balken verfährt ohne nennenswertes Kippen auf einer gekrümmten Bahn)

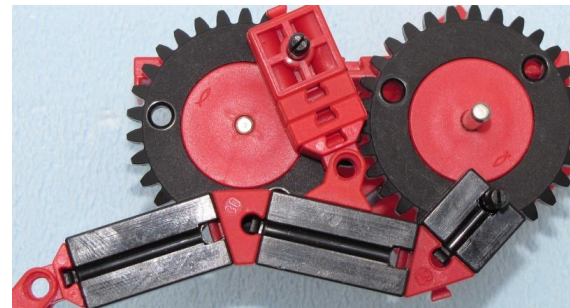


Abb. 15: Diesem Entwurf konnte noch keine sinnvolle Funktion zugeordnet werden

Dafür erwies sich einer der gefundenen „Prinzen“ sogar als „zwei Fahrwerke“: es kann in zwei um 90° verdrehten Lagen und nach passender Einstellung der Z30 zueinander als Fahrwerksbein eingesetzt werden (siehe die Abbildungen 16 bis 19).

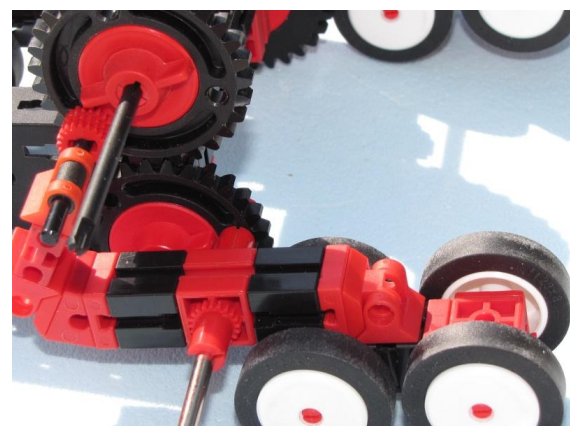


Abb. 16: Das IchBinZweiFahrwerke-Fahrwerk mit zwei Z30 übereinander, eingefahren ...



Abb. 17: ... und ausgefahren.

So gewappnet kann man sich jetzt die Fahrwerke der A380 zu Gemüte führen. Da ist von nicht-ebenen (räumlichen) Kniegelenken über Knie-im-Knie bis hin zu Mehrfach-Koppelgetrieben alles vertreten (Abb. 20). Und weiter als bis zum ft-Bilderpool muss man nicht suchen, um die Details zu finden, denn es gibt ft-Freaks, die sich auch an den [originalgetreuen Nachbau](#) eines derartigen technischen Sahnstücks heran wagen. Schließlich, nachdem einer der Gründe für den Abbruch des Projekts „A380“ die Verklei-

dung all der gewölbten Außenflächen war, könnte man angesichts der Möglichkeiten, die so ein Stück Moosgummi bietet, beinahe auf dumme Gedanken kommen. Noch obsiegt die Vernunft...



Abb. 18: Das IchBinZweiFahrwerke-Fahrwerk mit zwei Z30 nebeneinander, eingefahren ...



Abb. 19: ... und ausgefahren.

Quellen

- [1] *Viergelenkgetriebe*, fischertechnik hobby 2, [Band 5](#), Fischer-Werke, Tümlingen 1973, S. 4-17.



Abb. 20: Fahrwerk des Airbus A380 (Quelle: Wikipedia, Foto: Florian Lindner)

Elektromechanik

Vom Zählen und Abzählen (2)

Stefan Falk

In dieser Ausgabe stellen wir das im vorherigen Beitrag versprochene Modell vor, in dem wir einmal alle Register ziehen wollen: Ein Bausteinspender, der – natürlich ganz ohne Computer – eine einstellbare Anzahl von Bausteinen ausgibt.

Bisher hatten wir in der ft:pedia ja schon eine ganze Reihe von elektrischen Schaltungen, elektromechanischen und elektronischen Bauteilen kennengelernt: Taster als Öffner, Schließer und Wechsler

verschaltet, Polwendeschalter, Lampen, Elektromagnete, Relais mitsamt Schaltungen für Selbsthaltung und Selbstsperrung, Dioden, das E-Tec-Modul für lichtgesteuerte Modelle und natürlich Motoren.

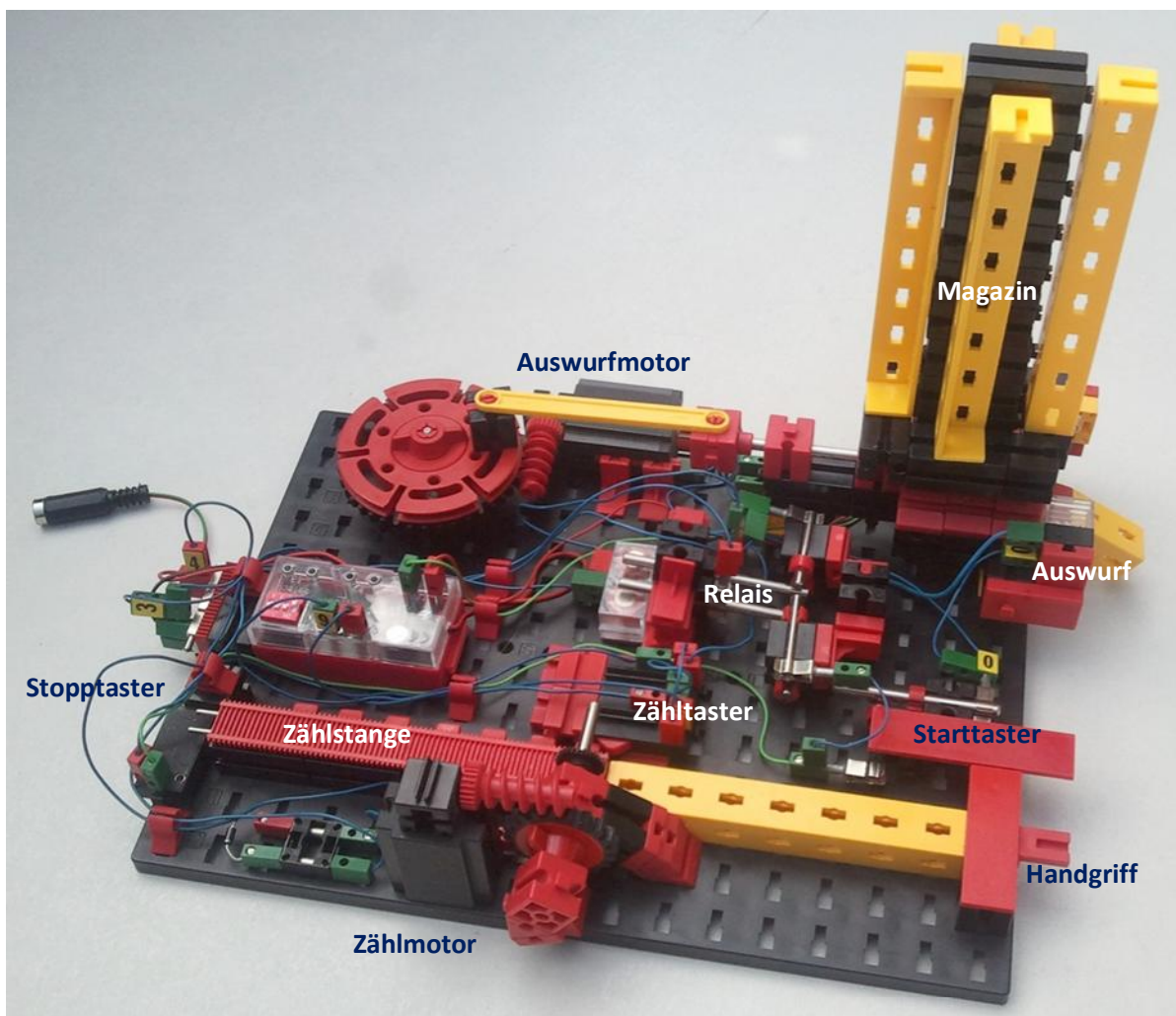
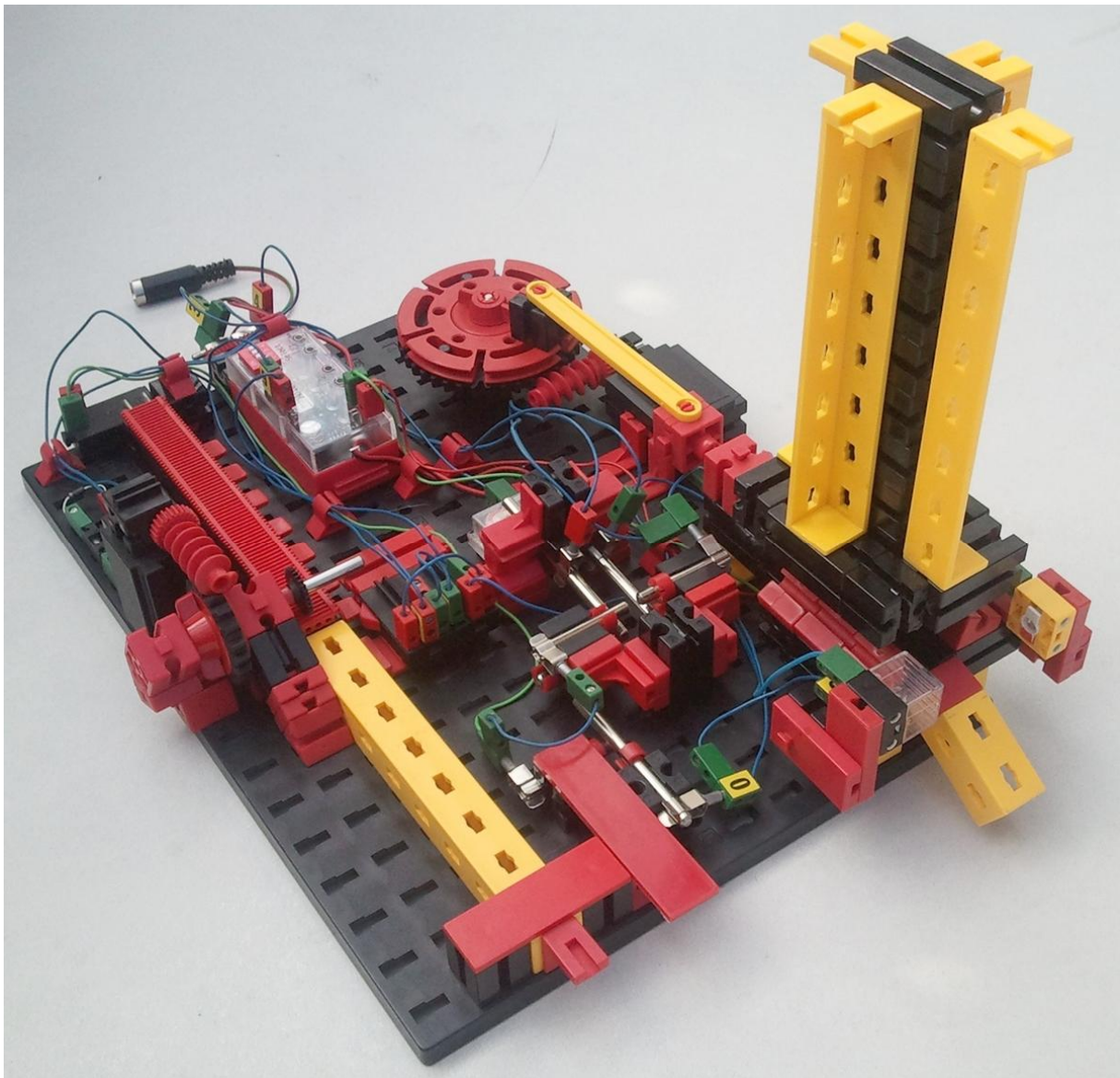


Abb. 1: Übersicht über das Modell



In diesem Motorsteuerungs- und Zähl-Artikel wollen wir ein etwas komplizierteres Modell in Angriff nehmen, bei dem tatsächlich all diese Elemente vorkommen.

Deshalb gleich vorweg der Hinweis: Dieses Modell verlangt beim Aufbau und beim Verstehen Geduld und Ausdauer. Damit ihr die Beschreibungen nachvollziehen könnt, solltet ihr die bisherigen Elektromechanik-Artikel der ft:pedia durchgearbeitet und verstanden haben. Wenn ihr aber durchhaltet, verspreche ich euch einige spannende Stunden und eine gut funktionierende, raffinierte Maschine. Wollen wir?

Die Aufgabenstellung

Das herzustellende Modell soll fischertechnik-Bausteine aus einem Magazin spenden. Das Besondere ist, dass wir die gewünschte Anzahl von Bausteinen vorab einstellen können, und genau diese Anzahl soll ausgegeben werden. Die Abbildungen geben euch von allen Seiten einen Überblick über das Modell. Wenn wir ganz fertig sind, wird das Modell so funktionieren:

Die gewünschte Anzahl Bausteine stellt man ein, indem man die lange Zählstange an dem kleinen Handgriff so weit aus dem

Modell heraus zieht, bis so viele Löcher des Statikträgers außerhalb des Modells sichtbar sind, wie man Bausteine haben möchte. Daraufhin drückt man kurz den Start-Taster (hier ein selbstgebauter, ein fertiger fischertechnik-Taster funktioniert natürlich genauso gut).

Ab dann wird die Maschine den Auswurfmotor und den Zählmotor immer abwechselnd so ein- und ausschalten, dass die Zählstange für jeden durch den Exzenter ausgeworfenen Baustein um genau ein Loch zurück in die Maschine hinein gezogen wird. Dieses Spiel wiederholt sich, bis die Zählstange den Stopptaster betätigt, der das Modell schließlich wieder abschaltet.

Aber der Reihe nach. Zum Aufbau des Modells gehen wir schrittweise vor:

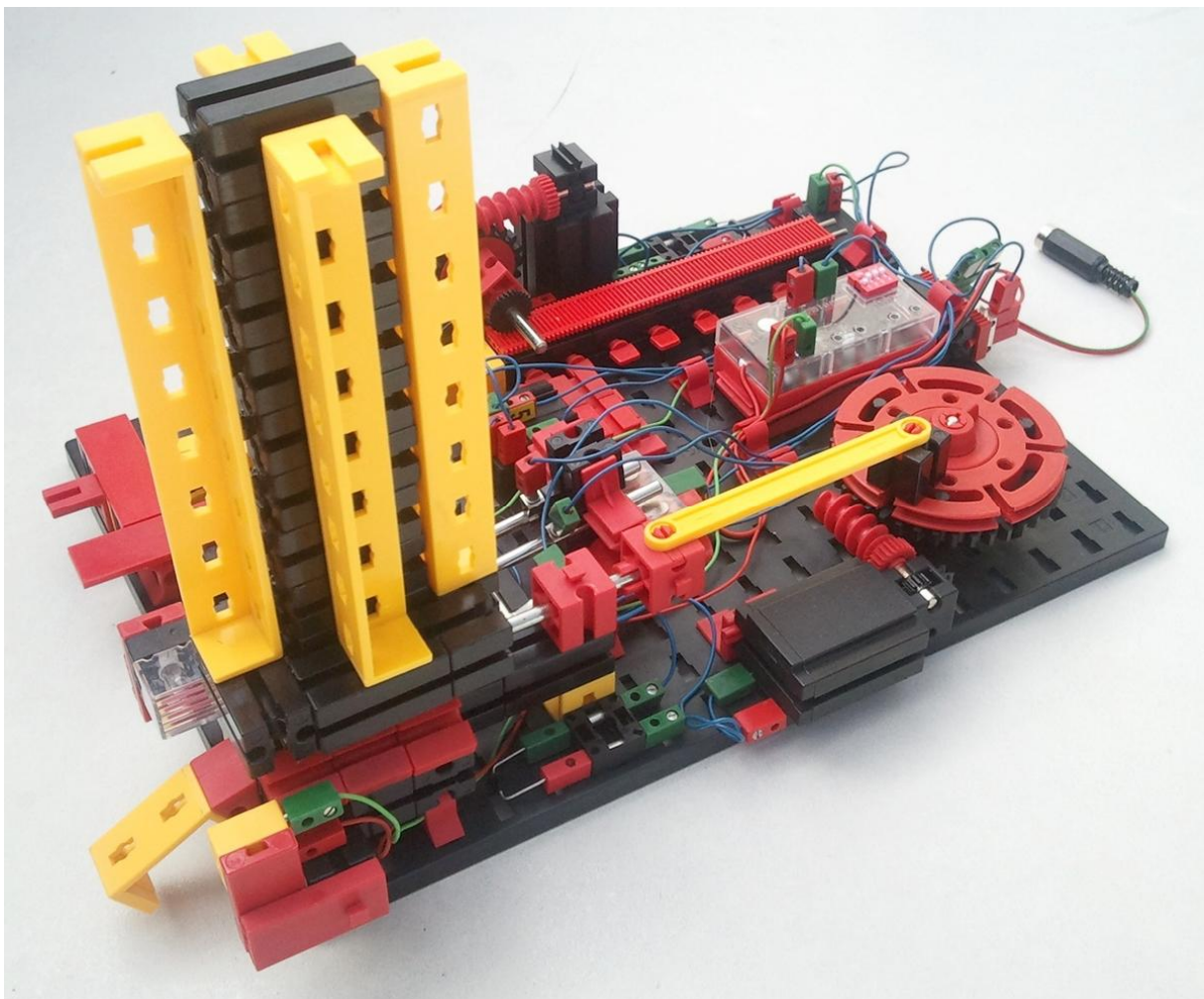
Der mechanische Aufbau

In den folgenden Abschnitten beschreiben wir zunächst den Aufbau des Modells, bevor wir die elektrische Schaltung besprechen.

Das Magazin

Das auffallendste Element des Modells ist das Bausteinmagazin, in dem bis zu neun Bausteine eingelagert werden können.

Das Magazin besteht oben nur aus einem kleinen Rahmen von Bausteinen, auf dem vier Statikträger 120 angebracht sind. Es ist nur auf seiner Rückseite (in Abb. 2 rechts) mit einem einzigen roten Baustein 15 mit Bohrung mit dem Fuß verbunden. Der Fuß wiederum besteht aus einem Gestell von Bausteinen, die bis zum Auswurf vorne reichen. An seinen Seiten



befinden sich einige weitere Bausteine mit drei Winkelsteinen 30 pro Seite als Führung für die herab fallenden Bausteine.

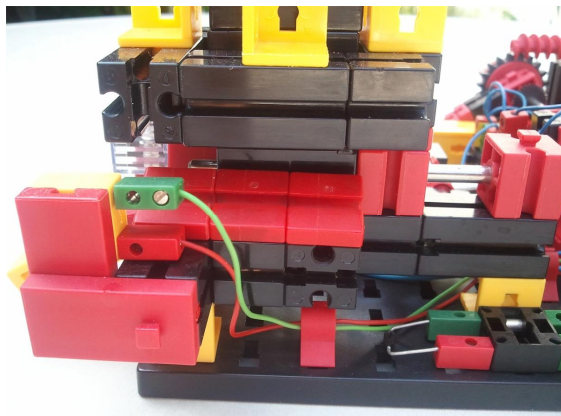


Abb. 2: Detailansicht aufs Magazin

Eine lange Metallachse kommt von der Rückseite, durch einen zweiten Baustein 15 mit Bohrung geführt. Abb. 2 zeigt sie ganz ins Magazin eingefahren: Man sieht ihr Ende über dem linken der drei Winkelsteine. Sie trägt in dieser Stellung sämtliche Bausteine im Magazin. Erst wenn sie durch den Exzenter nach hinten gezogen wird, fallen die Bausteine auf den Magazinboden, und beim nächsten Vorschub der Metallachse wird der unterste Baustein nach vorne ausgegeben.

Durch den von einem S-Motor angetriebenen Exzenter wird pro Umdrehung der Drehscheibe genau ein Baustein ausgeworfen. Am Auswurf vorne überwacht eine Lichtschranke den Ausstoß des Bausteins – der herausgeschobene Baustein unterbricht diese Lichtschranke so lange, bis er auf der Ausgaberutsche nach unten fällt.

Der Exzenter

Das periodische Vor und Zurück der Metallachse für den Bausteinauswurf wird ganz klassisch über einen Exzenter realisiert: Ein S-Motor dreht über ein Zahnrad Z40 eine Drehscheibe, auf der am Rand – exzentrisch eben, also nicht in der Mitte – eine Strebe angebracht ist, deren anderes Ende die besagte Metallachse treibt.

Abb. 3 zeigt, wie die Drehscheibe einfach mit einer kurzen Achse in einem Baustein 15 gelagert ist:



Abb. 3: Lagerung des Exzenter

Das Relais

Für die elektrische Schaltung benötigen wir ein Relais, also einen per Elektromagnet betätigten Schalter. Auf den Abbildungen seht ihr, wie unser *ft:pedia-Relais* [1] mitten auf der Bauplatte untergebracht werden kann.

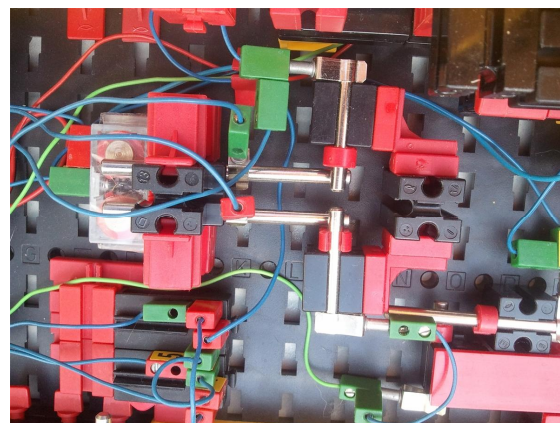


Abb. 4: Das Selbstbau-Relais

Wer eines besitzt, kann natürlich auch eines der fertigen fischertechnik-Relais verwenden. Für eine ausführliche Beschreibung der Relais schaut bitte in Ausgabe 4/2011 der *ft:pedia* nach [1]. Dort findet ihr auch den genauen Aufbau

unseres selbstgebauten Relais. Die elektrische Schaltung besprechen wir später in diesem Artikel noch ausführlich.

Die Zählstange

Das nächste wichtige Element unseres Modells ist die Zählstange: Ein Statik-Flachträger 120, beidseitig versteift mit je einem Flachstück, ist auf der Vorderseite mit einem als Handgriff gedachten Winkelstück versehen. Auf der anderen Seite wird die Stange mit vier Bausteinen 30 verlängert, auf denen je zwei Hubzahnstangen 60 angebracht sind.

Die Zählstange wird vorne am Modell von zwei Bausteinen 15 mit einer aufgeschobenen Platte 15 x 45 geführt, und etwa in ihrer Mitte von je zwei auf der Grundplatte befestigten Bausteinen 5 (Abb. 5 und 6).

Für den Zählvorgang sind auf der Seite sieben S-Riegel im Abstand von 15 mm angebracht. Beim Verschieben der Zählstange betätigen die ein Paar zweier fischertechnik-Minitaster. Wir brauchen an dieser Stelle nämlich eine Polwendschaltung, wie wir sie in ft:pedia 4/2011 auf Seite 7 beschrieben haben [1]. Dabei ist es besonders nützlich, dass man die fischertechnik-Minitaster einfach hintereinander befestigen kann, so dass durch einen Tastendruck mehrere Taster gleichzeitig umgeschaltet werden.

Damit wir die Taster in jeder benötigten Richtung gut justieren können, hängen sie an zwei Bausteinen 7,5. Diese wiederum sind mit Federnocken auf zwei Bausteinen 5 15 x 30 angebracht, die in der Bauplatte 500 stecken. Durch diese Anordnung können wir die Zählstange sowohl längs als auch quer zur Zählstange fein justieren.

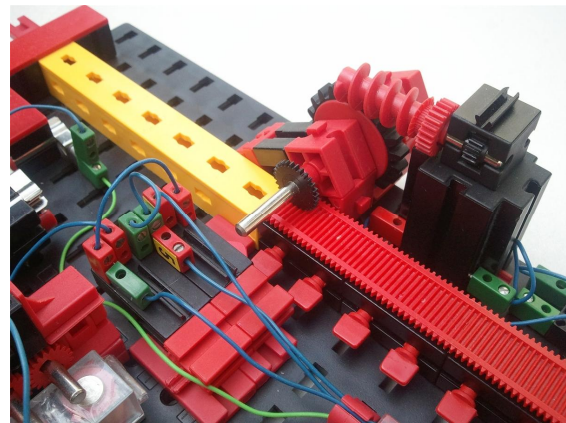


Abb. 5: Zählstange und Zählmotor

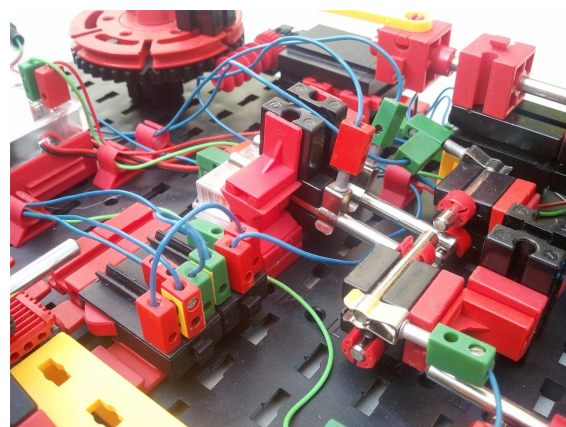


Abb. 6: Zählstange und Relais

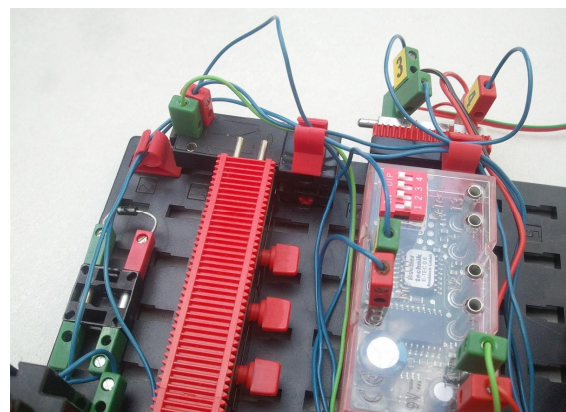
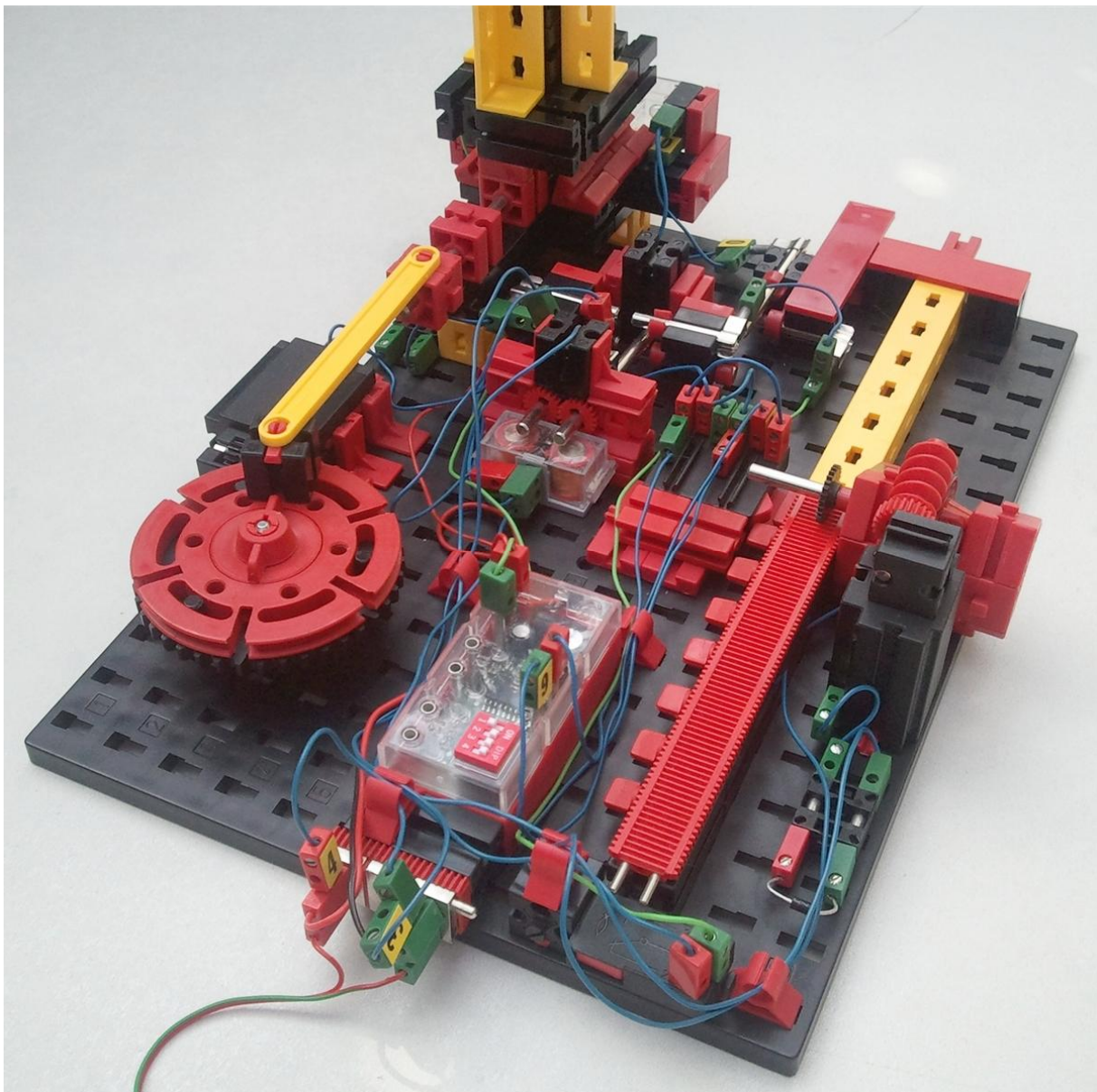


Abb. 7: Der Stopptaster

Am hinteren Anschlag befindet sich der Stopptaster. Damit der auch gut justiert werden kann, steckt er in einem via Federnocken auf der Bauplatte befestigten Baustein 15. Dessen Nuten liegen in Längsrichtung der Zählstange.



Auf diesem Bild seht ihr auch einen Vorschlag, wie wir eine der beiden später benötigten Dioden sauber anbringen können (die andere sitzt neben dem Ausgabemagazin).

Der Zählmotor

Der zweite verbaute Motor treibt die Zählstange an und hat die Aufgabe, diese schrittweise zurück ins Modell zu ziehen. Er ist einfach mit zwei Federnocken auf der Bauplatte angebracht. Seine Schnecke geht auf ein Zahnrad Z20, das auf der

Achse einer [31063](#) U-Achse 60 + Zahnrad Z28 m0,5 mini angebracht ist. Für den richtigen Abstand sitzt auf der Seite des kleinen Zahnrads noch ein [31579](#) „Abstandsring 3“.

Der Antrieb ist sehr stabil so angebracht, dass das kleine Zahnrad genau in die feinen Zahnstangen eingreift. Das gelingt durch die Kombination Winkelstein 30° – Baustein 15 – Winkelstein 60° – Baustein 15 mit Bohrung. Auf der Innenseite wird das noch durch einige Bausteine verstärkt, wie Abb. 8 zeigt:

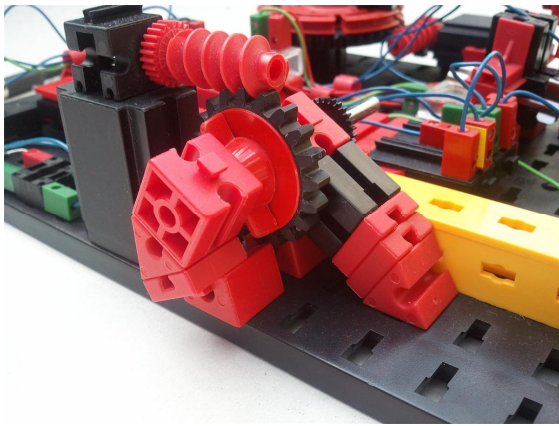


Abb. 8: Der Zählmotor

Um die Antriebsachse gut zu führen, ist sie auf der Außenseite nochmal gelagert. Dabei steckt der äußere Winkelstein 30° nur lose in der Bauplatte (das Z20 muss ja vorbei passen). Das macht aber nichts, denn auf dieser Seite wird die Konstruktion nur auf Druck nach unten beansprucht. Das feine Zahnrad wird nämlich von den Zahnstangen hochgedrückt. Dadurch ergibt sich auf der anderen Seite des Achslagers ein Druck nach unten. Den abzufangen dient die äußere Achsaufhängung.

Die Rutschkupplung

Hier gibt es ein kleines Problem: Einerseits wollen wir die Zählstange ja mit Muskelkraft aus dem Modell heraus ziehen, um damit festzulegen, wie viele Bausteine wir ausgeworfen haben möchten: Pro Baustein muss ein Loch des Statikträgers sichtbar sein, die Zählstange also 15 mm herausgezogen werden.

Andererseits soll der Motor die Zählstange wieder zurück ins Modell ziehen. Das kleine Zahnrad ist aber ständig im Eingriff mit den feinen Zahnstangen. Wir müssten also schon mit Gewalt ziehen, und das Zahnrad würde an den Zahnstangen einfach nur entlang rattern. Sehr unschön!

Aber die Lösung ist ganz einfach: Wir dürfen die Nabe des Z20 einfach nicht ganz fest ziehen! Dabei müssen wir genau das richtige Maß treffen: Die Zählstange muss bitte ganz leicht durchs Modell

gleiten können, bevor ihr den Motor und den Antrieb anbringt. Das Z20 muss dann gerade so fest angezogen werden, dass die Zahnstange vom Motor zuverlässig eingezogen werden kann. Sie muss aber so locker angezogen werden, dass das feine Zahnrad und damit die Antriebsachse im Z20 einfach „durchdrehen“ bzw. „durchrutschen“ können, wenn man bei stillstehendem Motor die Zählstange heraus zieht. Wegen dieser Wirkungsweise nennt man eine solche Konstruktion *Rutschkupplung*.

Es erfordert also etwas Geduld und Justierarbeit, bis alles sauber läuft. Baut bitte erst weiter, wenn die Zählstange sowohl hinreichend leicht herausgezogen werden kann als auch zuverlässig vom Motor eingezogen wird. Bis dieser Zustand erreicht ist, müsst ihr die Führung der Zählstange und die Anzugsstärke der Nabe des Z20 justieren.

Ein Tipp dazu: Bringt die Bausteine 15 am vorderen Ende und die Bausteine 5 in der Mitte der Zählstange so an, dass sie ein ganz kleines Bisschen mehr Platz als die 15 mm Breite der Zählstange bieten. Die oberen Nuten der Bausteine 15 müsst ihr dazu quer zur Zählstange ausrichten. Erst wenn die Zählstange leichtgängig läuft, bringt den Antrieb an und stellt die Stärke der Rutschkupplung ein.

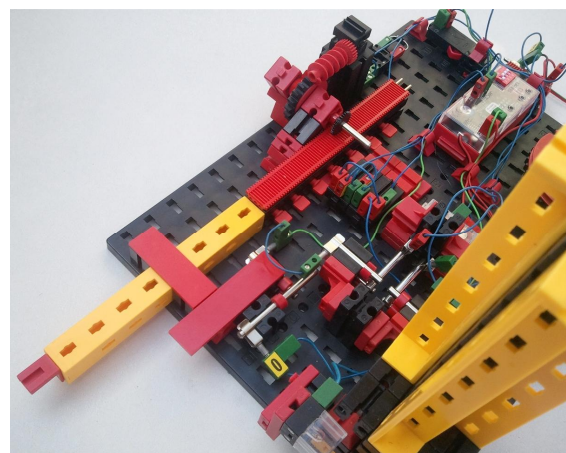


Abb. 9: Blick auf den Starttaster und die auf „4“ herausgezogene Zählstange

Der Starttaster

Das letzte Stückchen Mechanik ist einfach: Ein Selbstbautaster, später lediglich als Schließer verschaltet, passt vorne noch auf die Bauplatte 500. Seine Konstruktion ist in ft:pedia 4/2011 auf Seite 9 beschrieben [1].

Die elektrische Schaltung

Nachdem wir den mechanischen Aufbau des Modells gemeistert haben, müssen wir uns nun eine elektrische Schaltung überlegen, die alle beschriebenen Baugruppen wie gewünscht „zum Leben erweckt“.

Relais mit Selbsthaltung

Ganz grob funktioniert das Modell so, dass durch das Drücken des Start-Tasters das Relais anzieht und mittels Selbsthaltung angezogen bleibt (zur Selbsthaltung eines Relais könnt ihr nochmal in ft:pedia 4/2011 nachschauen [1]). Damit wird die Stromversorgung des restlichen Modells aktiviert, und zwar genau so lange, bis die Zählstange am Ende den Stopptaster betätigt. Der unterbricht dann die Stromzufuhr des Relais, es fällt also ab und schaltet das Modell aus. Soweit kennt ihr die Schaltung schon, wenn ihr die letzten Ausgaben der ft:pedia mitverfolgt habt.

Die zu lösende Kernaufgabe ist aber, dass der Exzenter genau so oft drehen und damit einen Baustein auswerfen muss, wie die Anzahl außen sichtbarer Löcher auf der herausgezogenen Zählstange anzeigt. Da die beiden Motoren wegen der völlig anderen Getriebeaufbauten nicht gleich schnell arbeiten werden, genügt es aber nicht, einfach beide gleich lange laufen zu lassen. Wir müssen sie schritt- bzw. Bausteinweise synchron laufen lassen.

Zwei Motoren steuern sich gegenseitig

Das realisieren wir so, dass immer nur einer der Motoren läuft, bis er einen

bestimmten Schritt vollendet hat. Dann soll nur der andere Motor laufen, bis auch dieser einen solchen Schritt gemacht hat. Die beiden Motoren warten also auf einander. Nach jedem ausgeworfenen Baustein haben beide Motoren einen kompletten Zyklus abgearbeitet: Genau ein Baustein wurde ausgeworfen, weil der Exzenter genau eine volle Umdrehung machte. Die Zählstange wurde um genau eine Zähl-Länge eingezogen, weil genau der nächste zur Zählung angebrachte S-Riegel vor den beiden Zählrastern ansteht.

Eine kleine Zeitreise

Eine ganz ähnliche Aufgabe wird übrigens in einem Modell „Bohrautomat“ im fischertechnik hobby-Begleitbuch 3-2 ab Seite 56 behandelt [3]. Dort werden ebenfalls zwei Motoren immer abwechselnd betrieben. Allerdings wird im damaligen Modell nichts gezählt; es läuft einfach endlos.



Abb. 10: Modell „Bohrautomat“ aus dem hobby-Begleitbuch 3-2 von 1973 [3]

Ihr erkennt auf Abb. 10 zwei der älteren fischertechnik-Motoren mit den großen Schnecken: Einer dreht den Drehteller mit den Werkstücken, der andere senkt bzw. hebt den Bohrer. Der Bohrer selbst wird von einem [mini-mot 1](#) angetrieben, der

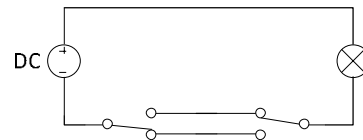
tatsächlich nur so groß wie ein Baustein 30 war und den man auf diesem Bild deshalb nicht sieht. Außerdem gibt es zwei Taster: Einer wird von Nocken auf dem Drehteller betätigt, der andere in der oberen Endlage des Bohrschlittens (oben in der Mitte in Abb. 10).

Der raffinierte Trick, die beiden Motoren genau synchron zu steuern, besteht in einer Schaltung, die bewirkt, dass einer der Motoren genau dann läuft, wenn *beide* Taster gedrückt oder *beide* nicht gedrückt sind. Der andere Motor läuft stattdessen genau dann, wenn jeweils *genau einer* der beiden Taster gedrückt ist. Das bewirkt folgenden Ablauf:

1. Kein Taster ist gedrückt: Motor 1 läuft, bis er „seinen“ Taster betätigt. Jetzt ist also genau ein Taster gedrückt. Deshalb ...
2. ... geht Motor 1 aus und Motor 2 läuft so lange, bis der auch „seinen“ Taster drückt. Dann sind beide Taster gedrückt, deshalb ...
3. ... geht Motor 2 wieder aus und Motor 1 wieder an. Bis der seinen Taster loslässt. Wiederum ist also nur ein Taster gedrückt und ...
4. ... Motor 1 geht aus und Motor 2 läuft, bis auch der seinen Taster loslässt. Und damit sind wir wieder beim Zustand 1: Motor 1 läuft, Motor 2 ist aus, und das ganze Spiel wiederholt sich.

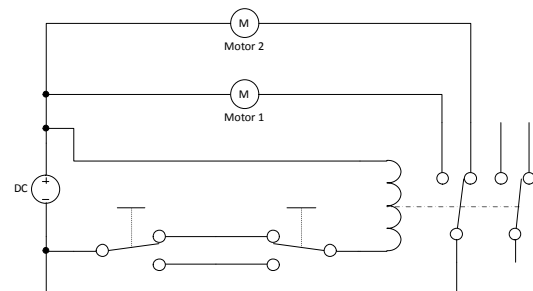
Die Motoren arbeiten also in „Halbschritten“: Immer vom ungedrückten bis zum gedrückten Taster bzw. vom gedrückten zum ungedrückten. Ganz egal, welcher der Motoren also evtl. schneller läuft: Jeder wartet mit seinem nächsten Halbschritt, bis der jeweils andere „nachgekommen“ ist. Genial, oder?

Eine solche Wechselschaltung hatten wir schon einmal in der ft:pedia. Schaut mal in ft:pedia 4/2011 auf Seite 6 Schaltung 3 an, mittels der wir eine Lampe von zwei Stellen aus schalten können [1]:



Schaltung 1: Wechselschaltung aus ft:pedia 4/2011

Die Lampe leuchtet, wenn beide Taster gedrückt oder wenn beide nicht gedrückt sind. Wir brauchen nur noch etwas, was bei genau einem gedrückten Taster einen anderen Verbraucher (den zweiten Motor nämlich) einschaltet. Das geht aber leicht mit einem Relais, wie *Schaltung 2* zeigt:



Schaltung 2: Wechselschaltung für zwei Motoren

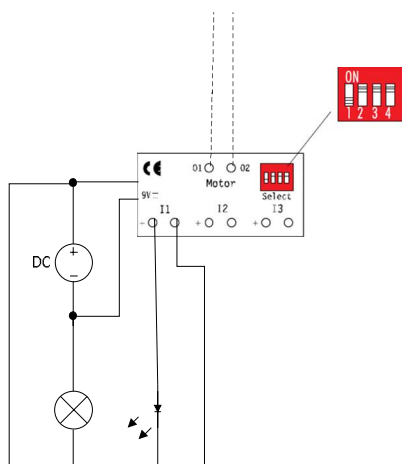
Nun läuft Motor 1 genau dann, wenn beide Taster im selben Zustand sind (also beide gedrückt oder beide ungedrückt), und Motor 2 läuft dann, wenn sie sich in unterschiedlichen Zuständen befinden. Bedenkt, dass die Schaltzeichen immer den Ruhezustand darstellen. In der gezeigten Tasterstellung ist das Relais aber angezogen, deshalb läuft dann Motor 1 und nicht Motor 2.

In der Schaltung des Bohrautomaten wird diese Schaltung lediglich um die Kurzschlussbremse ergänzt verwendet – ihr erinnert euch, siehe ft:pedia 1/2011 [2].

Lichtschanke anstatt Taster

Die beiden Motoren sind in unserem Bausteinspender natürlich der des Exzenter und der der Zählstange. Einen Taster können wir bei der Zählstange ja leicht von den darauf angebrachten S-Riegeln betätigen lassen. Den zweiten könnten wir direkt am Exzenter befestigen und wären schon fast fertig.

Allerdings ist es eleganter, den Auswurf durch eine Lichtschranke zu überwachen, um festzustellen, ob *tatsächlich* ein Baustein ausgeworfen wurde. Ein Taster am Exzenter würde ja nur die Umdrehungen des Exzenter feststellen, aber nicht merken, wenn das Magazin leer ist. Für die Lichtschranke können wir einfach einen Fototransistor (oder je nach eurem Bauteilbestand auch einen Fotowiderstand) an die I1-Eingänge eines E-Tec-Moduls anschließen. Das E-Tec stellen wir so ein, dass es einfach nur den Zustand der Eingänge verstärkt und am Ausgang wiedergibt:



Schaltung 3: Lichtschranke mit E-Tec-Modul

Allerdings verhält sich das E-Tec elektrisch natürlich anders als ein Taster. Sein O1-Ausgang liefert „-“ und O2 „+“, solange I1 kein Signal erhält. Wenn der Fototransistor beleuchtet wird und also durchschaltet, liefert O1 „+“ und O2 „-“.

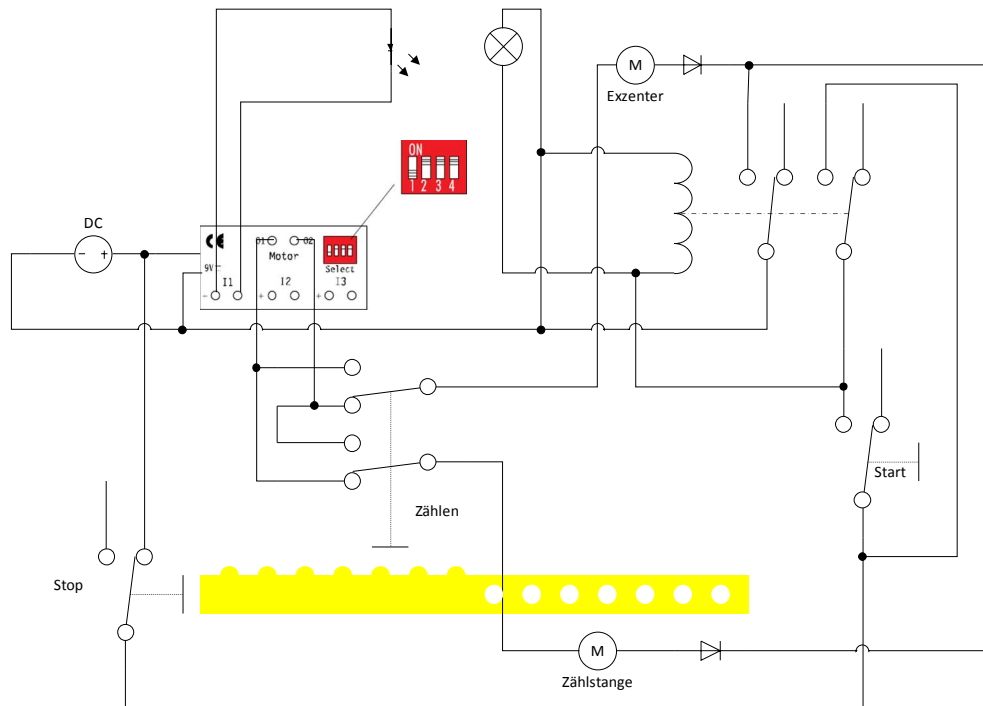
In der vollständigen *Schaltung 4* des Modells seht ihr, wie man das mit zwei

immer gleichzeitig betätigten Tastern ergänzen kann: Die beiden Zählaster sind ja direkt hintereinander angebracht, sodass einer den anderen mit betätigt. Verschaltet sind sie wie ein Polwendeschalter: In unbetätigtem Zustand (wenn die Taster also gerade zwischen zwei S-Riegeln stehen) sind der Zählmotor mit O1 und der Exzentermotor mit O2 verbunden. Bei gedrückten Zählastern ist es genau umgekehrt.

Die jeweils anderen Anschlüsse der Motoren sind über je eine Diode und das Relais mit dem Minuspol der Stromversorgung verbunden. Das Relais übernimmt wie schon beschrieben das Ein- und Ausschalten des ganzen Modells. Wozu aber die Dioden?

Denken wir uns die Dioden mal weg und die Motoren über das gerade angezogene Relais mit dem Minuspol der Stromversorgung verbunden. Wenn nun die Lichtschranke beleuchtet ist, weil gerade kein ausgeschobener Baustein sie versperrt, liegt also an O1 des E-Tecs „+“ an. Durch die gerade unbetätigt angenommenen Zählaster geht das also durch den Zählmotor zurück zum Minuspol der Stromversorgung. Der Zählmotor läuft also.

Aber vergesst nicht, dass am *selben* Anschluss ja auch noch der Exzentermotor angeschlossen ist. Das angezogene Relais verbindet zwar mit dem Minuspol, aber es hat doch einen gewissen Übergangswiderstand an der Kontaktstelle. Deshalb ist das Spannungspotenzial an der Verbindung der beiden Motoren und dem Relais nicht ganz identisch mit dem Minuspol der Stromversorgung. Es ist noch „ein bisschen positiv“. Deshalb fließt ein klein wenig Strom tatsächlich auch durch den Exzentermotor zurück zum E-Tec – an O2 nämlich. O2 liegt aber gerade wirklich auf „-“. Der überraschende Effekt ist deshalb, dass sich der Exzenter ganz langsam dreht, und zwar rückwärts!



Schaltung 4: Die vollständige Schaltung der Maschine

Das können wir natürlich nicht gebrauchen und bauen deshalb die beiden Dioden als Sperre ein. Die verhindern, dass ein wenig Strom unerwünscht und auch noch in der falschen Richtung durch den Motor fließt, der gerade ausgeschaltet sein sollte. Probiert aber ruhig aus, wie sich das Modell ohne die Dioden verhält!

Eine Alternative wäre übrigens, nur einen Zählstaster einzusetzen, der an O1 und O2 angeschlossen ist und dessen Zentralkontakt ein zweites Relais ansteuert. Mit diesem könnte man dann einen der Taster in *Schaltung 2* ersetzen. Vermutlich haben aber mehr von euch drei fischertechnik-Minitaster als zwei Elektromagnete oder Relais.

Zur Justage des Modells

Damit euer Modell wirklich korrekt funktioniert, hier ein paar Tipps, wie alles ineinander greifen muss:

- Nicht vergessen: Die Zählstange muss noch ohne ihren Antrieb leichtgängig gezogen werden können.
- Die Rutschkupplung des Zählstangenantriebs muss so eingestellt werden, dass man die Stange manuell herausziehen und die Antriebsachse in der Nabe des Z20 durchdrehen kann. Sie muss aber auch hinreichend fest angezogen sein, sodass der Zählmotor die Stange wieder einziehen kann.
- Möglicherweise werden die beiden Zählstaster nicht beide satt durchgedrückt, sondern nur einer, bevor der Zählmotor anhält. Ihr könnt dann versuchen, die Nabe des Z20 etwas

fester anzuziehen. Klappt das nicht, vertauscht die beiden Zähltaster miteinander, sodass der andere von den S-Riegeln betätigt wird.

- Die S-Riegel müssen so angebracht sein, dass die Taster nach dem Herausziehen auf eine bestimmte Anzahl sichtbarer Löcher unbetätigt sind, also zwischen zwei S-Riegeln stehen.
- Beide Zähltaster müssen ganz aneinander anliegen, damit sie gleichzeitig betätigt werden.
- Die Zählstange muss nach dem letzten Baustein noch ein kleines Stückchen Weg fahren, um den Stopptaster zu betätigen, aber nicht so weit, dass die Zähltaster nochmal umschalten. Dazu könnt ihr die Justiermöglichkeit des Stopptasters in der Nut seines Bausteins 15 ausnutzen.
- Je nach Lichtverhältnissen müsst ihr den Fotowiderstand evtl. mit einer schwarzen Störlichtkappe oder zur Not einer roten Kappe mit Loch abdecken, damit die Lichtschranke sauber schaltet. Das könnt ihr leicht testen, denn das E-Tec-Modul blinkt anders, wenn der Fototransistor durchschaltet bzw. sperrt.
- Vergesst nicht, dass der Fototransistor richtig herum angeschlossen werden muss: Sein roter Pluspol muss mit dem Pluspol von I1 des E-Tec verbunden werden. Bei einem Fotowiderstand („Fotozelle“) ist die Polung egal (der hat auch keinen rot markierten Anschluss).
- Justiert das Relais sauber, so dass beide Kontakte satt durchschalten (siehe ft:pedia 4/2012 [1]).
- Es gibt ein youtube-Video vom fertigen Modell [4]. Dort könnt ihr vergleichen, wie das Modell funktioniert.

Geschafft!

Wenn alles richtig klappt, ist der Gesamt- ablauf damit so:

1. Die Zählstange ist ganz eingeschoben und betätigt den Stopptaster. Die Zähltaster stehen vor dem ersten S-Riegel und sind unbetätigt. Der Exzenter steht so, dass die Stange ganz in den Auswurf hineingeschoben ist. Auf ihr drauf liegen die Bausteine im Magazin.
2. Der Strom wird eingeschaltet. Nichts passiert.
3. Der Mensch zieht die Zählstange um so viele Statiklöcher heraus, wie er Bausteine haben möchte. Die Zähltaster müssen danach zwischen zwei S-Riegeln stehen und also wiederum unbetätigt sein. Der Stopptaster ist jetzt unbetätigt, würde also über seinen Ruhekontakt Strom durchlassen.
4. Der Starttaster wird kurz gedrückt. Das Relais zieht an und versorgt das Modell mit Strom. Es bleibt auch nach dem Loslassen des Starttasters angezogen, weil es sich über die Selbsthalteschaltung selber weiter mit Strom versorgt. Die zum Relaismagneten parallel geschaltete Lampe der Lichtschranke geht gleichzeitig mit an und beleuchtet den Fototransistor.
5. Da die Lichtschranke gerade frei ist, liefert das E-Tec an O1 „+“ und der Zählmotor läuft an.
6. Nach einem kurzen Augenblick betätigt der erste S-Riegel die Zähltaster. Dadurch geht der Zählmotor aus und der Exzentermotor läuft los. Er zieht die Stange zunächst ganz aus dem Magazin heraus. Die Bausteine fallen nach unten. Der Exzenter läuft weiter und schiebt mit der wieder

- einziehenden Stange den untersten Baustein auf den Auswurf.
7. Eben dieser ausgeschobene Baustein unterbricht, bevor er herunterfällt, die Lichtschranke. Augenblicklich schaltet das E-Tec um: Der Exzentermotor wird wieder stillstehen und der Zählmotor laufen. Die Zählstange wird also weiter eingezogen, bis ...
 8. ... die Zählstange wieder zwischen zwei S-Riegeln stehen und also losgelassen werden.
 9. Und schon steht der Zählmotor wieder still und der Exzenter läuft wieder an. Es stößt den halb ausgeschobenen Baustein nun endgültig aus; der fällt die Rutsche hinab.
 10. Dadurch wird aber die Lichtschranke wieder freigegeben. Das E-Tec schaltet wieder um, und der Exzenter läuft weiter.
 11. Solange die eingestellte Anzahl Bausteine noch nicht ausgeworfen ist, wiederholen sich die Schritte 5 bis 10. Mit jedem Durchgang wird genau ein Baustein ausgeworfen und die Zählstange um genau einen S-Riegel eingezogen. Da die Riegel im 15-mm-Abstand angebracht sind, entspricht das den Löchern des Statikträgers.
 12. Wenn aber die eingestellte Anzahl erreicht wurde, läuft der Zählmotor nach Schritt 10 nur noch das letzte

kleine Stück, bis die Zählstange den Stopptaster betätigt. Der unterbricht damit die Stromversorgung des Relais. Es fällt also ab, hebt damit seine eigene Selbsthaltung auf (bleibt also aus) und schaltet alle Motoren ab.

Wenn ihr dieses Modell nachgebaut und zum Funktionieren gebracht habt, kann ich mich wieder nur herzlich bedanken und euch gratulieren! Gebt bitte im Forum Rückmeldung, ob ihr euch weitere solch komplexe Modelle in der ft:pedia zum Nachbauen wünscht. Auf jeden Fall werden wir hier noch viele Anwendungen der mittlerweile angesammelten Schaltungstricks sehen – es gibt noch so viel zu entdecken!

Quellen

- [1] ft:pedia Ausgabe 4/2011, <http://www.ftcommunity.de/ftpedia>
- [2] ft:pedia Ausgabe 1/2011, <http://www.ftcommunity.de/ftpedia>
- [3] fischertechnik, *hobby-Begleitbuch* 3-2, 1973. <http://www.ft-datenbank.de/search.php?keyword=39532>
- [4] Stefan Falk, *Bausteinspender aus ft:pedia 2/2012*, youtube-Video des fertigen Modells, 2012. <http://www.youtube.com/watch?v=uqTwxKc2xQQ>

Tipps & Tricks

ft-Spezialteile made by TST (Teil 1)

Andreas Tacke

In einer lockeren Reihe wird TST einige von ihm entwickelte Spezialteile vorstellen, die so manche Lücke beim Bauen mit Fischertechnik schließen.

In meinem ersten Beitrag geht es um Teile, die die Verwendung des Differentials 31500 aus den 80er Jahren mit den aktuellen Motoren möglich machen.

Vor ca. sieben Jahren begann ich mit dem Bau eines Trucks (siehe Bild unten). Für die Antriebsachsen wollte ich die Differential 31500 verwenden. Leider sind die Antriebseinheiten so selten, dass ich mir etwas Anderes einfallen lassen musste.

Für den Antrieb wollte ich die Powermotoren nutzen. Nur: Mit den Standardteilen war dies nicht möglich. Da die Differential einen speziellen Anschluss haben, blieb mir nichts anderes übrig, als selbst

etwas zu entwickeln.

Als erstes baute ich die Antriebswellen 31416, 31417 und 31418 aus Messing nach, da die originalen Teile nicht mehr erhältlich waren (Abb. 1).



Abb. 1: Nachbau der Antriebswellen 31416, 31417 und 31418 aus Messing

Als nächstes musste ich eine Möglichkeit



finden, diese Antriebswellen mit den Powermotoren zu verbinden. Nach etlichen Versuchen habe ich dann eine Lösung gefunden: Ich fertigte mir Messingadapter, womit das Problem gelöst war (Abb. 2, 3).



Abb. 2: Adapter für Power- und XM-Motor



Abb. 3: Adapter für den Anschluss der Antriebswellen

Mit diesen Teilen ließ sich dann ein stabiler Antrieb herstellen, und ich konnte die aktuellen Motoren verwenden (Abb. 4).



Abb. 4: Power- und XM-Motor mit Adapter

Wie in dem Versuchsaufbau in Abb. 5 und Abb. 6 zu sehen, passen die Teile perfekt ins Raster und lassen sich mit den Originalteilen spielfrei verbinden.

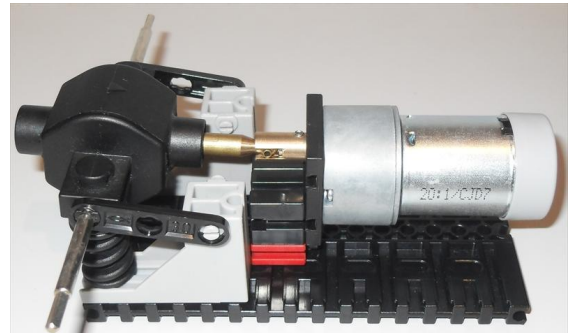


Abb. 5: Differential31500 mit Power Motor

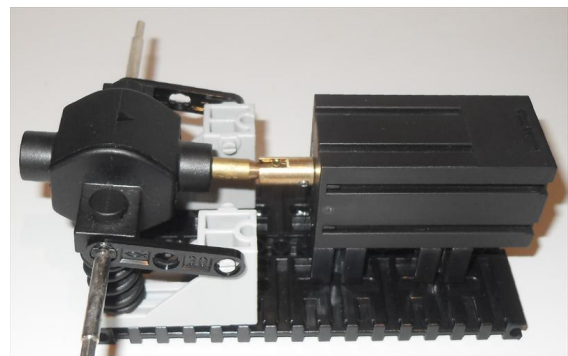


Abb. 6: Differential31500 mit XM Motor

Da sieht man mal wieder: Geht nicht gibt es nicht – mit handwerklichem Geschick und dem richtigen Werkzeug lässt sich so manches Problem im ft-Modellbau lösen.

Und wem eines von beidem (oder auch beides) abgeht, dem fertige ich die Teile gerne.

Einsteigermodell

Bootsbau mit fischertechnik

Michael Tilli

Der Sommer ist da, und noch kein außentaugliches Modell ist in Sicht? Mit diesem Artikel wird Abhilfe geschaffen: Hier werden die Grundlagen des Schiffbaus mit fischertechnik gezeigt und zum Schluss ein kleines Rennboot gebaut.

Schiffsrumpf

Fangen wir also gleich an mit den verschiedenen Möglichkeiten ein ft-Schiff zu bauen. Und was ist das wichtigste bei einem Boot? Der Rumpf, logisch ohne ihn fährt kein Schiff.

fischertechnik EPP-Rumpf

Der einzige Schiffsrumpf von fischertechnik ist der gut 31 cm lange Rumpf aus expandiertem Polypropylen, kurz EPP, einem Kunststoff, aus dem zum Beispiel auch Mehrwegverpackungen hergestellt werden. Dieses Material wird im Modellbau besonders aufgrund seiner Stabilität und dem geringen Gewicht geschätzt. Der Rumpf war unter der Artikelnummer 126927 in den Baukästen *Basic Boats* und *Advanced Ships + More* enthalten. Weder in den Kästen noch als Einzelteil ist der Rumpf heute noch über Knobloch zu beziehen. Die kurze Verkaufsdauer von 2007 bis 2009 ist wohl vor allem auf die geringe Nachfrage aufgrund des Fehlens einer Motorisierung der Schiffe zurückzuführen [3, 4].

Sortierkasten 1000

Auch als Sofort-fertig-Rumpf lässt sich der Grundkasten 1000 verwenden, allerdings sind die daraus entstehenden Boote sehr groß und langsam. Er eignet sich eher für eine Bohrplattform oder ein Baggerschiff.

Pneumatik und Hydrauliktanks

Auch eine Möglichkeit für Schwimmkörper aus fischertechnik: Man nehme mehrere Tanks, befestige diese aneinander und schließe die Anschlüsse mit Pneumatik-Stopfen, oder man verwendet kurze Schlauchstücke, mit denen man immer zwei Anschlüsse eines Tanks verbinde. Es besteht allerdings das Risiko, dass sich ein Stopfen löst und dadurch im Wasser verschwindet. Obendrein kann dann Wasser in einen Behälter eindringen (hier bleibt aber noch Zeit, das Boot an Land zu steuern).

Fischer Dübelboxen

Auf der ft:community wird von Harald Steinhaus vorgeschlagen, diese Boxen zum Bau von Bootsrümpfen einzusetzen. Eventuell muss der Deckel festgeklebt werden, damit kein Wasser durchkommt. Der Vorteil dieser Boxen ist, dass sie an beiden Seiten zu fischertechnik kompatible Nuten besitzen, allerdings sind sie 5 mm zu kurz um exakt ins ft-Raster zu passen.

Mir ist kein fischertechnik-Modell bekannt, in dem diese Boxen eingesetzt werden.

Plastikflaschen

Die Plastikflaschen sind leicht, billig, relativ stabil und wasserfest. Deshalb bilden sie gute Schwimmkörper. Gerade

für schnell aufzubauende Boote sind sie ideal, allerdings muss man sie mit einem Gummi am Boots Aufbau befestigen und sie sehen nicht besonders schön aus.

Folien und Tüten

Hier wird zuerst das Skelett des Schiffsrumpfes aus fischertechnik-Teilen gebaut. Dann wird entweder eine Einkaufstüte oder ein Müllsack aufgeschnitten und am Schiff entweder mit Statikriegeln oder mit Klebeband befestigt. Selbstverständlich dürfen keine Löcher in der Tüte vorhanden sein. Danach wird Klebeband von außen auf den Rumpf geklebt. Es verhindert ein Aufschlitzen des Rumpfes und macht ihn stabiler. Diese Bauweise ähnelt stark der Skelett- oder Spantenbauweise, die zum Bau von hölzernen Schiffen seit dem 8. Jahrhundert angewendet wird. Im fischertechnik-System sind die Vorteile dieser Bauweise eine hohe Flexibilität bei der Formgebung des Rumpfes, die Nachteile sind ein verhältnismäßig hohes Gewicht und die Gefahr, dass Wasser durch eine undichte Tüte eindringt.

Styropor

Styropor besitzt ähnliche Eigenschaften wie der ft-Rumpf: Es ist leicht, stabil und einfach zu verarbeiten. Wer eine Stichsäge verwenden kann, hat die Möglichkeit, aus Styroporplatten passgenaue und leichte Schwimmkörper herzustellen. Für einfache Modelle genügt oft auch schon ein Bastelmesser, um das Material zu schneiden. Vorsicht bei der Befestigung von Schrauben im Styropor, sie können herausbrechen. Deshalb empfehle ich, die Aufbauten mit Kleber oder Gummibändern zu befestigen. Ein weiteres Problem ist besonders bei grobem Styropor, dass sich Kügelchen ablösen und die Wohnung verdrecken.

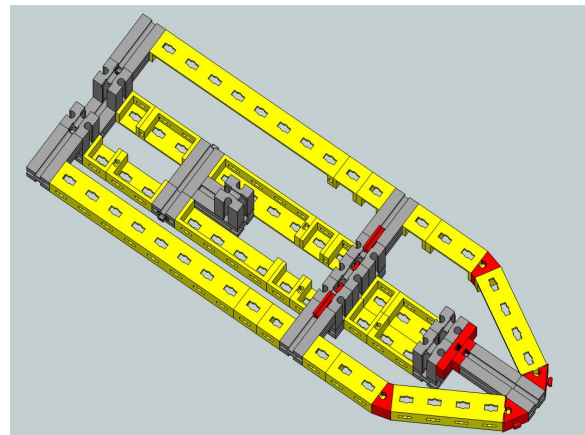


Abb. 1: Der Rumpf unseres Schiffchens

Das Schiffmodell

Das oben gezeigte Bild ist eine mit Google SketchUp erstellte Grafik des Rumpfes. Hinten ist der „Kiel“ etwas breiter, vorne wird er schmaler. Zwischen dem breiten Heck und dem schmalen Kiel befindet sich eine Trennung, die zur Stabilisierung des Schiffes dient und gleichzeitig den Akku hält. Vorne läuft der Bug spitz zu und bildet in etwa die übliche Schiffsform nach. Wenn das Boot fertig ist wird der Rumpf dann mit einer Plastiktüte bespannt.

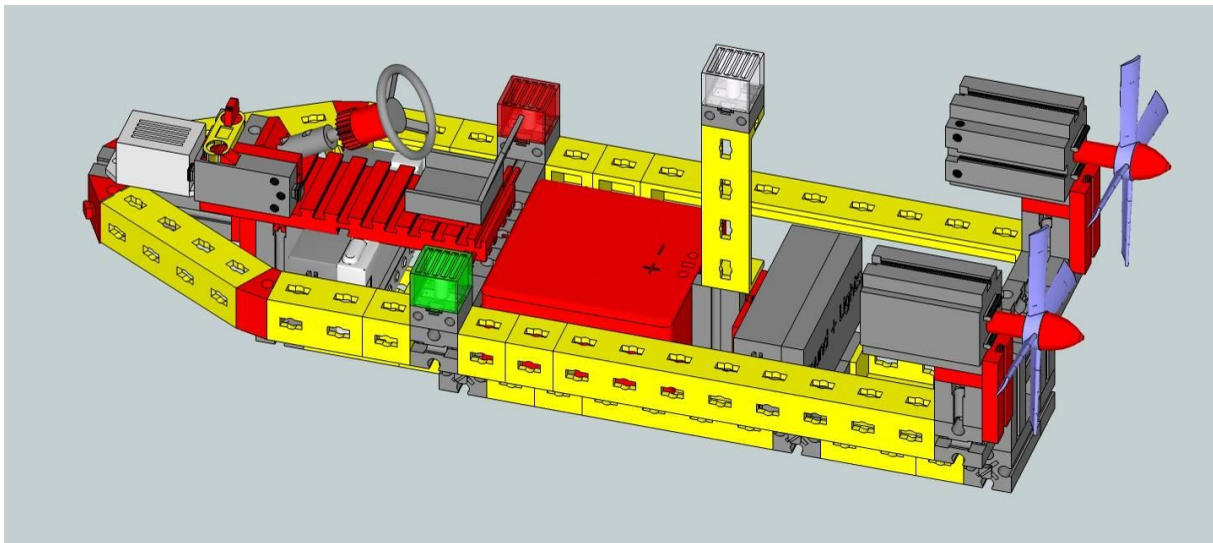
Antrieb

ft-Luftschrauben 36337 und 36559

Die Luftschrauben von fischertechnik können sowohl als Propeller wie bei einem Airboat oder auch als Schiffsschraubenersatz eingesetzt werden. Ich ziehe in der Regel den vierflügeligen, kleineren Propeller vor, da er weniger Platz benötigt. Beide Propeller sind ungefähr gleich schnell.

Schiffsschraube

Für möglichst realistische Schiffsantriebe empfehle ich die Schraubenlösung von Harald, welche schon in der ft:pedia vorgestellt wurde [2].



Schaufelradantrieb

Wer ein eher langsames und dafür großes Schiff bauen möchte, ist mit Schaufelrädern bestens bedient. Man kann sie direkt an einen Power- oder XM-Motor anschließen. Für noch schönere Schaufelräder kann man auch das Malteser-Schaltrad verwenden.

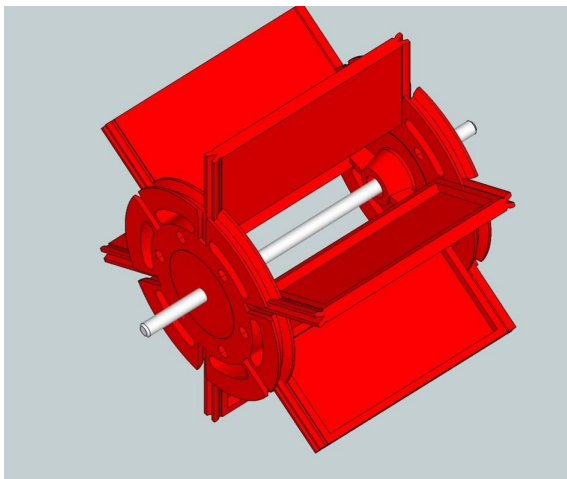


Abb. 2: Schaufelrad aus Standardteilen

Lenkung

Gut, wir haben jetzt einen Rumpf, und wir haben die verschiedenen Antriebe kennen gelernt. Aber wie lenkt man das Schiff? Es bietet sich an, entweder ein großes Ruder zu verwenden, oder man kombiniert Lenkung und Antrieb.

„Raupenlenkung“

Wer schon einmal einen Bagger gesehen hat, weiß, dass dieser auf Ketten fährt, die einzeln angetrieben werden. Der Bagger lenkt, indem eine Kette vorwärts läuft und die andere rückwärts. Für ein Schiff bedeutet das, dass man zwei Motoren verwendet, die je eine Schraube, ein Schaufelrad, oder einen Propeller antreiben. Praktischerweise kann man beim IR Control Set direkt eine Raupefunktion auswählen: Drückt man „M1↑“, drehen sich beide Motoren in die gleiche Richtung und das Schiff fährt vorwärts. Bei „M1↓“ fährt das Schiff rückwärts. Beim Lenken mit „M2→“ und „M2←“ dreht jeweils ein Motor vorwärts, der andere rückwärts. Es dreht immer der Motor in Lenkrichtung rückwärts.

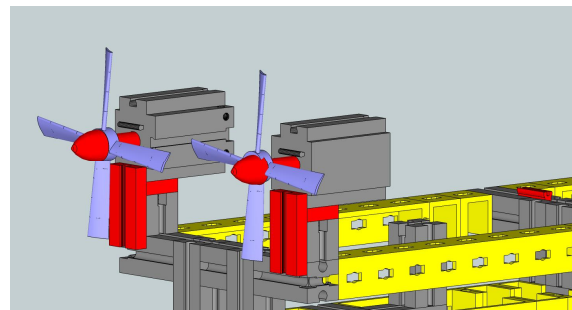


Abb. 3: Unser Boot mit zwei Mini-Motoren

Antrieb drehen

Eine Möglichkeit, mit nur einem Antriebsmotor zu lenken ist die Schraube oder den Propeller zu drehen.

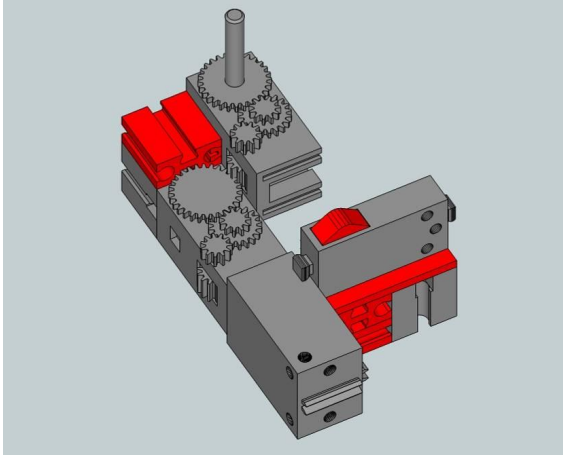


Abb. 4: Der Motorblock

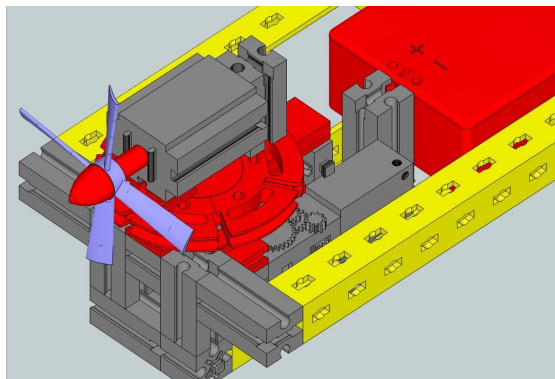


Abb. 5: Fertig eingebauter, drehbarer Motor

Der fertige Motorblock wird in den Rumpf geschoben. Anschließend wird der Propellermotor mit der Drehscheibe aufgesteckt.

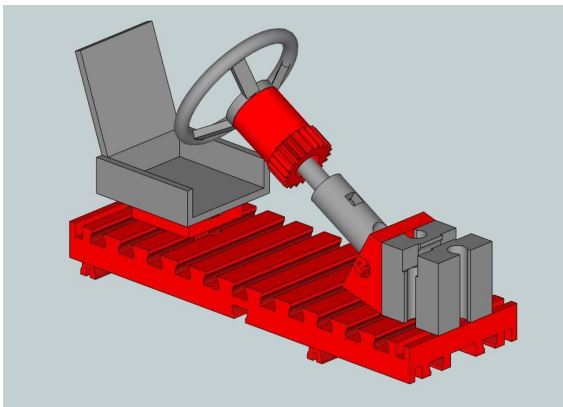


Abb. 6: Der Fahrersitz

Kleine Extras

Für fischertechnik-Fans mit mehr Bauteilen oder dem Wunsch, dem Boot noch mehr „Leben“ einzuhauchen, habe ich mir einige kleine Zusatzfunktionen und Verschönerungen überlegt.

Sound and Lights-Modul

Mit dem Sound and Lights-Modul kannst du dafür sorgen, dass das Boot Geräusche macht. Besonders interessant ist für uns die Möglichkeit, dass beim nach rechts Fahren der Ruf „Hart Steuerbord“ und beim nach links drehen „Hart Backbord“ ertönt. Wie man diese Funktion erreicht musst du schon selber herausfinden, ich sage dir nur, dass du höchstwahrscheinlich eine eigene Schaltung benötigst. Zusätzlich kann man auch den Betrieb mindestens eines Antriebsmotors mit einem lauten Brummen verbinden.

Positionslichter

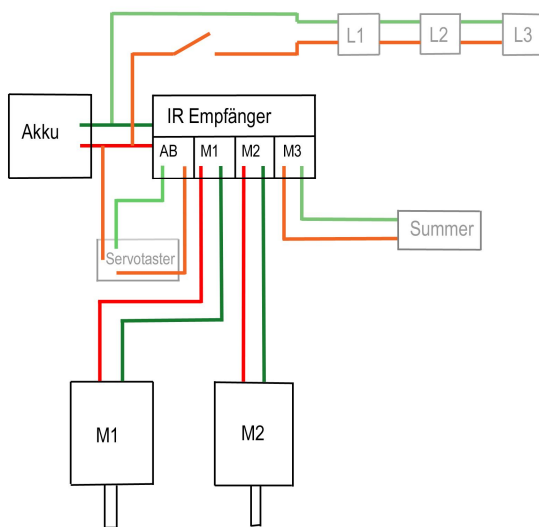
Positionslichter zeigen anderen Schiffen an, wo das eigene Schiff liegt. Dazu besitzt das Boot ein weiße Rundum-Leuchte, rechts eine grüne Lampe und links eine rote. Die Positionslichter müssen von Sonnenuntergang bis Sonnenaufgang eingeschaltet sein [1].

Schiffssirene

Mit einem Summer, der an den ungenutzten Motorausgang M3 angeschlossen wird, erhält man ein lautes Schiffshorn.

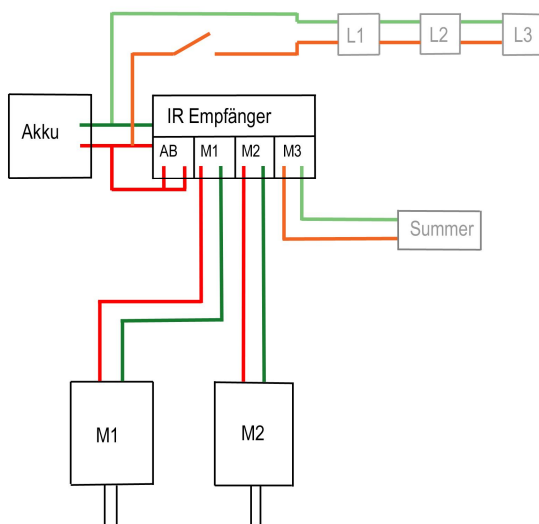
Schaltpläne

In den folgenden Schaltbildern werden die schwarzen Bauteile zwingend gebraucht; die grauen zeigen die Möglichkeiten zum weiteren Ausbau des Schiffes. Die roten und grünen Linien symbolisieren natürlich die Kabel, wobei beim IR-Empfänger und dem Summer auf die Polung zu achten ist.



Schaltung 1: Modell mit Drehlenkung

Beim Modell mit drehbarer Lenkung empfehle ich, den Servotaster zu verwenden, um einen besseren Geradeauslauf zu gewährleisten. Damit das Steuer beim alten IR Control Set nicht herumschlägt, muss man in der Regel den Geschwindigkeitsknopf für M2 drücken. Verwendet man das neue Control-Set, ist es leider nicht mehr möglich, auf diese Art einen Servo zu bauen. Deshalb sollte man hier einfach den Lenkhebel nur vorsichtig betätigen.



Schaltung 2: Raupensteuerung

Bei der Schiffsversion mit zwei Motoren wird beim IR Control Set die eingebaute Raupenfunktion genutzt. Beim neueren Control Set steuert man mit beiden Steuerknüpfeln.

Hinweise

Das hier gebaute Boot solltest du nur in der Badewanne oder in strömungsfreien Gewässern fahren lassen, und nach Möglichkeit auch so, dass du das Boot jederzeit wieder herausholen kannst, falls Batterie oder Fernbedienung ausfallen oder Wasser durch die Plane kommt. Wenn die Tüte, die den Rumpf bildet, beschädigt wird, musst du eine neue verwenden, da man Löcher darin nur sehr schlecht wieder schließen kann.

Literatur und Ideen

- [1] Weitere Informationen zu den Positionslichtern:
<http://www.segeln-wissen.de/segelwissen/seezeichen/positionslichter.html>
- [2] Vorschlag zum Nachbauen, der Speedantrieb von Harald:
http://www.ftcommunity.de/categories.php?cat_id=2253
- [3] Designtechnisch sehr gelungenes Schaufelradboot von schnaggels:
http://www.ftcommunity.de/details.php?image_id=11165
- [4] Ein Feuerlöschboot, ebenfalls von Schnaggels:
http://www.ftcommunity.de/details.php?image_id=33007



Computing

The fischertechnik Interface for the Rest of us

Erik Andresen

Wer sein fischertechnik-Interface unter Linux nutzen möchte, stellt fest, dass es von Robo Pro keine Linux-Version gibt. Um nicht doch auf eine Windows-Installation zurückgreifen zu müssen, gibt es Bibliotheken, mit denen das Interface auch von Linux aus programmiert werden kann. Dieser Beitrag beschreibt die Installation so einer Bibliothek unter einem Ubuntu Linux mit Inbetriebnahme des Robo Explorers.

fischertechnik und Linux

Als „unwirtschaftlich“ wurde eine versprochene Linux Version bezeichnet: Wer ein Robo-Interface kaufte in dem Glauben, dass die angekündigte Linux-Version mit der Artikelnummer 93297 [3] wirklich erscheint, wurde eines Besseren belehrt. Da sich der typische Linux-Anwender aber nicht mit rein mechanischen Modellen zufrieden gibt, wurden mit der *ftApi* [5] und der *libroboint* [1] (früher unter dem Namen *libft*) zwei Bibliotheken entwickelt, mit denen sich das Robo-Interface unter Linux betreiben lässt. Bei der *ftApi* handelt es sich um eine von Mitarbeitern der Goethe-Universität in Frankfurt geschriebene objektorientierte C++-Bibliothek.

Die alternative C-Bibliothek *libroboint* ist der im C-Compiler-Paket für Windows enthaltenen Knobloch-Bibliothek *ftlib* [7] nachempfunden: Programme, die für die *ftlib* geschrieben wurden, sollten ohne große Änderungen mit der *libroboint* lauffähig sein.

Beide Bibliotheken unterstützen nur das Arbeiten im „Online“-Modus, können die Modelle also nur steuern, wenn das Modell mit dem PC verbunden ist. Für den „Download“-Betrieb kann das *RoboInt-Basic* [6] verwendet werden. Das neue TX

Interface wird bislang von keiner der beiden unterstützt.

Eigenschaften der *libroboint* sind unter anderem:

- Kompatibilität mit der Knobloch-Bibliothek *libft*.
- In der Programmiersprache C geschrieben, was die Anbindung von vielen Programmiersprachen ermöglicht.
- Durch die Verwendung von *libusb* [8] ist kein Kernel-Modul erforderlich. Damit ist der Betrieb unter vielen Betriebssystemen, wie Linux, BSD-Varianten, Mac OS-X und Android möglich.
- Alternative Anbindung in der höheren Programmiersprache Python [10].
- Unterstützt werden Intelligent Interface mit Extension, Robo Interface, Robo I/O Extension und Robo RF Data Link.
- Grafisches Diagnose- und Testprogramm *ftdiagnose* zum schnellen Testen eines Interface.
- Upload von mit RoboIntBasic erstellten Programmen über das Konsolenprogramm *ftuploader*.

Wie ihr Name schon sagt, ist die *libroboint* eine Bibliothek: Sie ist damit kein eigen-

ständiges Programm wie Robo Pro, mit ihr können aber eigene Programme erstellt werden, die Modelle kontrollieren. In diesem Beitrag soll die *libroboint* unter Linux installiert und ein Robo Explorer damit kontrolliert werden.

Installation der libroboint

Aktuell ist *libroboint 0.5.0*, die mit einem Browser direkt als gepacktes Archiv unter dem Namen *libroboint-0.5.0.tar.gz* von der Homepage [1] herunter geladen werden kann und hier unter einem Ubuntu 12.04 [2] installiert wird. Unter anderen Distributionen verläuft die Installation ähnlich.

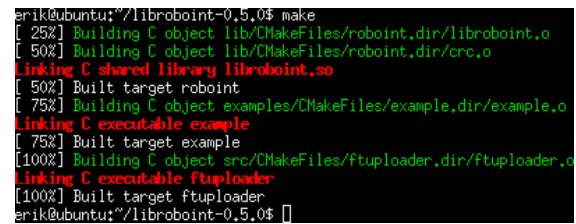
Die Software muss vor der Installation kompiliert werden: Dazu müssen ein C-Compiler sowie weitere Hilfsprogramme vorhanden sein. Unter Ubuntu sind dies die Pakete *cmake*, *gcc*, *libc6-dev*, *libusb-dev* und *python-qt4*. Bei anderen Linux-Distributionen sollten die Pakete einen ähnlichen, wenn nicht identischen Namen haben.

Für die Installation wird eine Konsole benötigt. Diese findet man in Ubuntu's Unity-Oberfläche unter dem normalerweise links oben positionierten Dash-Knopf → *Installierte Anwendungen* → *Terminal*.

```
> tar xvzf ~/Downloads/libroboint-0.5.0.tar.gz
> cd libroboint-0.5.0
>
> cmake .
> make
> sudo make install
> sudo ldconfig
>
> sudo cp udev/fischertechnik.rules /etc/udev/rules.d/
> sudo udevadm control --reload-rules
>
> sudo make diag
> sudo cp diag/ftdiagnose /usr/local/bin
> sudo chmod 755 /usr/local/bin/ftdiagnose
```

Listing 1: Vollständige Installation der Bibliothek mit Diagnoseprogramm. Das Kommandoprompt, üblicherweise „username@hostname:/Arbeitsordner\$“, wird hier verkürzt durch ein „>“ dargestellt.

Listing 1 zeigt den kompletten Vorgang; die Befehle werden alle nacheinander in die Konsole eingegeben. Dabei wird die *libroboint* zuerst aus dem vom Browser Firefox standardmäßig verwendeten Ordner *Downloads* entpackt und mit dem Dreisatz *cmake*, *make* und *make install* kompiliert und installiert.



```
erik@ubuntu:~/libroboint-0.5.0$ make
[ 25%] Building C object lib/CMakeFiles/roboint.dir/libroboint.o
[ 50%] Building C object lib/CMakeFiles/roboint.dir/crc.o
Linking C shared library libroboint.so
[ 50%] Built target roboint
[ 75%] Building C object examples/CMakeFiles/example.dir/example.o
Linking C executable example
[ 75%] Built target example
[ 75%] Building C object src/CMakeFiles/ftuploader.dir/ftuploader.o
Linking C executable ftuploader
[100%] Built target ftuploader
erik@ubuntu:~/libroboint-0.5.0$ []
```

Abb. 1: Kompilierung der libroboint mit make. Erscheint hier kein Fehler, kann die Bibliothek installiert werden.

CMake prüft dabei, ob auf dem Computer alle benötigten Bibliotheken und Programme vorhanden sind und erstellt abschließend die Anweisungen, die *make* benutzt, um die Software zu kompilieren (vergleiche Abb. 1). Mit dem als *root* aufgerufenen *make install* wird die Bibliothek in das Verzeichnis */usr/local* installiert und mit *ldconfig* wird dem Betriebssystem mitgeteilt, dass eine neue Bibliothek vorhanden ist. Dazu sollte der Pfad */usr/local/lib* in die Datei */etc/ld.so.conf* eingetragen sein. Unter Ubuntu ist dies bereits der Fall.

Damit man als Benutzer die Berechtigungen für den Zugriff auf die ft-Hardware erhält, wird eine *udev*-Regel kopiert und anschließend *udev* durch neu laden mitgeteilt, die Regel gleich anzuwenden. *udev* ist ein Systemdienst, der bei Ereignissen wie dem Einstecken eines USB-Gerätes Regeln, z. B. das Setzen von Berechtigungen, ausführt.

Mit *make diag* wird das Python-basierte Diagnoseprogramm kompiliert und installiert. Die letzten zwei Zeilen kopieren das Diagnoseprogramm und machen es ausführbar.

Mit einem abschließenden Test wird alles überprüft. Dazu wird ein Robo-Interface mit Sensoren und Aktoren (z. B. einem Taster und einem Motor) über USB mit dem Computer verbunden. Das Diagnoseprogramm wird über die Konsole mit *ftdiagnose* gestartet.

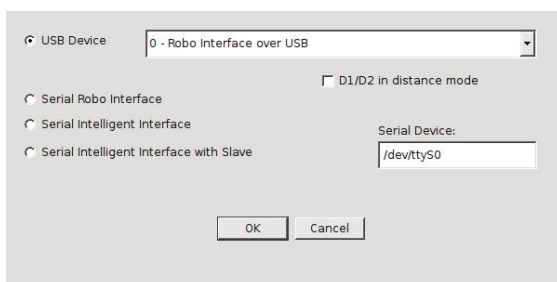


Abb. 2: Öffnen-Dialog des Diagnoseprogramms, das an USB erkannte Interface wird verwendet.

Im Dialog *File* → *Open Interface* zum Verbinden mit dem Interface können die verschiedenen Interface-Arten geöffnet werden. Zur Verfügung stehen die Interfaces aus der Robo-Familie über USB und RS232 (seriell) sowie das Intelligent Interface mit und ohne Extension. Da wir das Interface über USB angeschlossen haben, wird es, wie in Abb. 2 dargestellt, unter den USB-Geräten angezeigt und kann durch Drücken auf den Knopf „OK“ geöffnet werden.

Ist das Interface korrekt angeschlossen, kann es, wie in Abb. 3 zu sehen, getestet werden: Dabei können Aktoren wie Motoren und Lampen gesteuert und alle digitalen und analogen Eingänge ausgelesen werden. Sollte etwas nicht wie erwartet funktionieren, lohnt sich ein Blick in die Datei *README* der *libroboint*. Zusätzlich kann z. B. im Forum der ft Community [4] um Hilfe gebeten werden.

Besteht das Interface alle Tests, kann nun ein komplettes Modell in Betrieb genommen werden. Ein C-Beispiel ist in der *libroboint* im Ordner *examples/* zu finden. Die erweiterte Dokumentation im Ordner *doc/html/* kann mit dem Befehl *make doc* generiert werden.

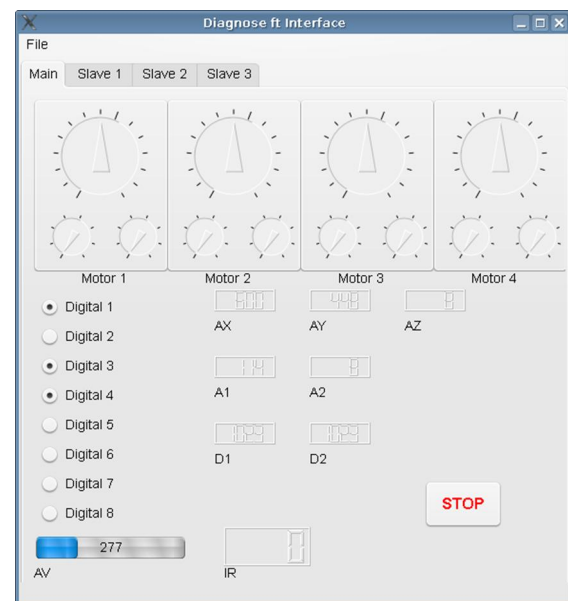


Abb. 3: Diagnoseprogramm, Auslesen der Eingänge und Steuern von Motoren ohne erst ein Programm zu schreiben.

Die Welt erkunden

Für Programmieranfänger ist die Sprache C schwer zugänglich, deswegen wird hier beispielhaft ein Robo Explorer (Bild 4) mit Python gesteuert. Wer noch nie mit Python gearbeitet hat, dem sei das Tutorial [9] auf der Homepage empfohlen. Es gibt keinen Unterschied, ob der Explorer dabei direkt

über USB oder mit dem Robo RF Data Link mit dem Rechner verbunden ist.

Die Dokumentation zur Python-Schnittstelle bekommt man auf die bei Python übliche Weise durch die Anweisungen `import robointerface` und `help(robointerface)` aus dem Python-Interpreter.

Die Python-Schnittstelle kennt einfache Befehle zum Steuern einzelner Motoren. Für komplexere Befehle wie „fahre vorwärts“ fügen wir der Bibliothek die neue Funktion `setSpeed(l, r)` aus Listing 2 hinzu: Mit `setSpeed(7, 7)` fährt unser mobiler Roboter mit maximaler Geschwindigkeit vorwärts, mit `setSpeed(-7, -7)` mit maximaler Geschwindigkeit rückwärts, mit `setSpeed(-3, 3)` dreht er mit mittlerer

Geschwindigkeit nach links usw. Dabei wird auf die Funktion `setMotor(iMotor, cDirection, iSpeed)` zurückgegriffen: `iMotor` ist die Nummer des Motors, der angesteuert wird, im Bereich von eins bis vier, `cDirection` ist „l“ für Links, „r“ für Rechts und „s“ für Stopp. Das letzte, optionale Argument ist die Geschwindigkeit als Zahl von eins bis sieben (Standardwert ist sieben).

Mit Hilfe der Funktion `setSpeed()` soll der Roboter jetzt

1. solange kein Hindernis vorhanden ist, vorwärts fahren,
2. bei einem Hindernis links drehen, bis die Sicht frei ist, und
3. anschließend wieder wie in 1. vorwärts fahren.

```

1 def setSpeed(self, l, r):
2     """Set the speed and direction of the Explorer in one command
3
4     l = speed of left engine -7 <= l <= 7
5     r = speed of right engine -7 <= r <= 7
6     """
7     # handle left engine
8     if l > 0:
9         # left forward
10        self.SetMotor(1, 'l', l)
11    elif l < 0:
12        # left reverse
13        l*=-1
14        self.SetMotor(1, 'r', l)
15    else:
16        # left stop
17        self.SetMotor(1, 's', l)
18
19    # handle right engine
20    if r > 0:
21        # right forward
22        self.SetMotor(2, 'l', r)
23    elif r < 0:
24        # right reverse
25        r*=-1
26        self.SetMotor(2, 'r', r)
27    else:
28        # right stop
29        self.SetMotor(2, 's', r)
30 # add new function to lib
31 RoboInterface.setSpeed = setSpeed

```

Listing 2: Erweiterung der robointerface-Bibliothek durch die für den Robo Explorer geschriebene Funktion `setSpeed()`.

```

1 from time import sleep
2 from robointerface import *
3
4 DIST = 50
5 SPEED = 7
6 SPEED_TURN = 7
7
8 ri = RoboInterface(bEnableDist=1)
9 while True:
10     # normally forward
11     ri.setSpeed(SPEED, SPEED)
12     # forward until we see an obstacle
13     while ri.GetD1() > DIST:
14         sleep(0.1)
15     # obstacle ahead, stop
16     ri.setSpeed(0, 0)
17
18     # turn left
19     ri.setSpeed(-SPEED_TURN, SPEED_TURN)
20
21     # until free
22     while ri.GetD1() < DIST:
23         sleep(0.1)

```

Listing 3: Python-Quelltext, der den Robo Explorer fahren lässt. Hindernissen wird durch eine Drehung nach links ausgewichen.

Den Algorithmus, realisiert in Python, zeigt Listing 3.

Zeile 8 öffnet das Robo Interface: da die Distanzsensoren standardmäßig deaktiviert sind, müssen sie mit `bEnableDist=1` aktiviert werden. Ausgelesen wird der erste Distanzsensor mit `getD1()`, was die Entfernung zum ersten Ziel in Zentimetern zurückgibt. Die `sleep()`-Befehle verhindern, dass der Prozess die komplette CPU belegt.

Erweiterungen sind immer möglich. So nutzt Listing 4 in einem separaten Thread den Fotowiderstand an AX, um je nach Helligkeit die Lampen an- und auszusprechen.

Listing 5 zeigt das komplette Programm. Es kann als Basis für eigene Programme verwendet werden, damit der Robo Explorer – auch von Linux aus – die Welt erkunden kann.

```

1 def light_thread():
2     while True:
3         if ri.GetAX() > 1000:
4             # dark: turn on lights
5             ri.SetOutput(5, 7)
6             ri.SetOutput(6, 7)
7         else:
8             # bright: turn off lights
9             ri.SetOutput(5, 0)
10            ri.SetOutput(6, 0)
11            sleep(0.1)
12
13 thread.start_new_thread(light_thread, ())

```

Listing 4: Ein- und ausschalten der Lampen am Explorer, je nach Helligkeit.

Quellen

- [1] Andresen, Erik: *ROBO Interface Bibliothek für Unix-artige Systeme*. <http://defiant.homedns.org/~erik/ft/libft/>
- [2] Canonical Ltd.: *Ubuntu*. <http://www.ubuntu.com>
- [3] ft Community: *Computing ROBO Pro Linux*, ArtNr 93297. <http://www.ft-datenbank.de/search.php?keyword=93297>
- [4] ft Community: *Forum der fischertechnik Community*. <http://forum.ftcommunity.de>
- [5] Holger Friedrich, David D.: *ftApi, c++ Bibliothek für Linux und MacOS-X*. <http://www.vsi.cs.uni-frankfurt.de/fischer/>
- [6] Kaiser, Thomas: *Basic für das Robo Interface von fischertechnik*. <http://www.ft-fanpage.de/roboint/robobas.html>
- [7] Knobloch GmbH: *C-Compiler für ROBO Interface Version 1.66*. <http://www.fischertechnik.de/ResourceImage.aspx?raid=4363>
- [8] Libusb.org: *Bibliothek für den Zugriff auf USB-peripherie vom Userspace*. <http://www.libusb.org>
- [9] Python.org: *Einführung in die Programmiersprache Python*. <http://docs.python.org/tutorial/> (Deutsche Übersetzung: <http://tutorial.pocoo.org/>)
- [10] Python.org: *Homepage der Programmiersprache Python*. <http://www.python.org>



Abb. 4: Robo Explorer (Foto: fischertechnik)

```

#!/usr/bin/env python

import thread
from time import sleep
from robointerface import *

DIST = 50
SPEED = 7
SPEED_TURN = 7

def setSpeed(self, l, r):
    """Set the speed and direction of the Explorer in one command
    l = speed of left engine -7 <= l <= 7
    r = speed of right engine -7 <= r <= 7
    """
    # handle left engine
    if l > 0:
        # left forward
        self.SetMotor(1, 'l', l)
    elif l < 0:
        # left reverse
        l*=-1
        self.SetMotor(1, 'r', l)
    else:
        # left stop
        self.SetMotor(1, 's', l)

    # handle right engine
    if r > 0:
        # right forward
        self.SetMotor(2, 'l', r)
    elif r < 0:
        # right reverse
        r*=-1
        self.SetMotor(2, 'r', r)
    else:
        # right stop
        self.SetMotor(2, 's', r)

# add new function to lib
RoboInterface.setSpeed = setSpeed

def light_thread():
    while True:
        if ri.GetAX() > 1000:
            # dark
            ri.SetOutput(5, 7)
            ri.SetOutput(6, 7)
        else:
            # bright
            ri.SetOutput(5, 0)
            ri.SetOutput(6, 0)
        sleep(0.1)

if __name__ == "__main__":
    ri = RoboInterface(bEnableDist=1)
    thread.start_new_thread(light_thread, ())
    while True:
        # normally forward
        ri.setSpeed(SPEED, SPEED)
        # forward until we see an obstacle
        while ri.GetD1() > DIST:
            sleep(0.1)
        # obstacly ahead, stop
        ri.setSpeed(0, 0)
        # turn left
        ri.setSpeed(-SPEED_TURN, SPEED_TURN)
        # until free
        while ri.GetD1() < DIST:
            sleep(0.1)

```

Listing 5: Komplettes Robo Explorer Programm aus dem Beitrag.

Projekt

HP-GL-Plotter (Teil 3)

Dirk Fox

Im ersten Teil des Beitrags wurde die Konstruktion eines fischertechnik-HP-GL-Plotters vorgestellt [1]. Der zweite Teil des Beitrags führte in Version 1 der Steuersoftware ein, die einfache HP-GL-Grafiken aus .csv-Dateien einlesen und plotten kann [2]. Im dritten Teil wird die Steuersoftware um HP-GL-Vektorgrafik-Befehle erweitert, mit denen Kreise, Kreisbögen, Kreissektoren, Rechtecke und regelmäßige N-Ecke geplottet werden können.

Version 1 der Steuersoftware für unseren ftpedia-Plotter war auf die Implementierung der fünf Befehle des HP-GL-Standards beschränkt, die für das Plotten einfacher HPGL-Dateien (wie z. B. in HP-GL konvertierter Zeichnungen aus gängigen CAD-Formaten) ausreichen: die HP-GL-Kommandos IN, PU, PD, PA und PR [2].

Tatsächlich umfasst der gesamte HP-GL-Standard mehr als 60 Befehle [3]. Einige davon sind bei unserem HP-GL-Plotter funktionslos, wie die Auswahl des Stifts und seiner Eigenschaften (SP: *Select Pen*, PT: *Pen Thickness*, PW: *Pen Width*) oder der Papierauswurf (PG: *Advance Full Page*), denn der funktioniert nur manuell.

Weitere Befehlsgruppen wie die Füll-Befehle (RA, RR: *Fill Rectangle*) oder die Stricharten (LT: *Line Type*, FT: *Fill Type*) lassen sich mit unserem einfachen Stiftplotter nicht sinnvoll nutzen, denn Schraffuren sind viel zu aufwändig. Auch diese Befehle habe ich daher ausgeklammert.

Verzichtet habe ich – bisher – auch auf die Implementierung der Schreibkommandos (*Character Plot Instructions*), da Robo Pro keine String-Verarbeitung beherrscht. Eine HP-GL-Standard konforme Implementierung des Befehls LB (*Label*) würde nämlich die Übergabe der zu plottenden Zeichen als Zeichenkette (*String*) erfordern.

Vektorgrafik-Befehle

Will man mit einem Plotter geometrische Formen zeichnen – wie z. B. Kreise, Quadrate, Rechtecke, regelmäßige Vielecke – oder ansprechende Computergrafiken (Balken- oder Kreisdiagramme) erzeugen, ist das mit den einfachen Plot-Befehlen PA und PR aufwändig. Daher beherrscht HP-GL komplexere Vektorgrafik-Befehle, die das Zeichnen von geometrischen Körpern erheblich vereinfachen.

Die wichtigsten dieser Vektorgrafik-Befehle sind die **Kreisbogen-Befehle** (AA, AR) und das **Kreis-Kommando** (CI). Mit letzterem können auch gleichmäßige N-Ecke geplottet werden. Dazu zählen außerdem die **Rechteck-Befehle** (EA, ER) und der **Kreissektor-Befehl** (EW). Diese sechs Befehle haben eine Gemeinsamkeit: Sie plotten immer – unabhängig davon, ob der Stift zuvor gesenkt oder angehoben war.

In Version 2 der Steuersoftware habe ich diese sechs Befehle ergänzt, sodass die Implementierung nun die in der folgenden Tabelle zusammengefassten elf HP-GL-Kommandos beherrscht (Tab. 1).

Nr.	Kürzel	Funktion	Param.
0	IN	<i>Initialize</i>	0
1	PU	<i>Pen Up</i>	0
2	PD	<i>Pen Down</i>	0
3	PA	<i>Plot Absolute</i>	2
4	PR	<i>Plot Relative</i>	2
5	AA	<i>Plot Arc Absolute</i>	3-4
6	AR	<i>Plot Arc Relative</i>	3-4
7	CI	<i>Circle</i>	1-2
8	EA	<i>Edge Absolute Rectangle</i>	2
9	ER	<i>Edge Relative Rectangle</i>	2
10	EW	<i>Edge Wedge</i>	3-4

Tab. 1: Liste der in Version 2 der Steuer-
software implementierten HP-GL-Befehle

Die neuen Befehle erfordern jedoch ein paar grundsätzliche Änderungen an der Robo Pro-Implementierung.

Erweiterter Parser

Die Ergänzung des HP-GL-Befehlsumfangs der Steuersoftware zwang mich zu einer Überarbeitung des Parsers (Hauptprogramm): Wie man an der vierten Spalte von Tabelle 1 erkennen kann, muss der Parser nun auch HP-GL-Befehle mit vier Parametern verarbeiten können. Dazu muss er fünf Listenelemente einlesen – den Befehl selbst und die bis zu vier Parameter.

Damit dies nicht die Anlage von fünf verschiedenen Dateien erfordert, habe ich in Version 2 des Steuerprogramms den Einlesemechanismus modifiziert: Befehle und Parameter werden aus einer einzigen .csv-Datei beliebigen Namens eingelesen.

Diese Datei muss mit der Zeile

Befehl;X;Y;Alpha;Beta

beginnen; anschließend folgen zeilenweise die Befehle mit je vier Parametern.³ Eine solche Datei lässt sich leicht durch einen csv-Export aus einer Excel-Tabelle erzeugen. Vor dem Starten des Programms wird diese Datei über die Menu-Option „Datei“ / „.csv-Speicher für Listen laden“ ausgewählt und eingelesen.⁴

Das in der Datei verwendete Trennzeichen (Semikolon, Komma oder Tabulator) kann vorher unter „Datei“ / „Trennzeichen für .csv-Dateien“ gewählt werden – anders als frühere Versionen von Robo Pro verarbeitet die aktuelle Version 3.1.3 das nun korrekt. Exportiert man die .csv-Datei aus Excel, wird als Default-Trennzeichen ein Semikolon verwendet. Die dem Steuerprogramm beiliegenden Beispiel-Dateien verwenden daher alle einheitlich ein Semikolon.

Neue Programmstruktur

Die zahlreichen Unterprogramme und Parameter des Steuerprogramms machen die Nutzung globaler Variablen zu einer Fehlerquelle – zu leicht übersieht man dabei einen Seiteneffekt. Daher werden in Version 2, wie es sich für eine saubere Programmierung gehört, Parameter an die Unterprogramme übergeben. Innerhalb der Unterprogramme wird fast ausschließlich mit lokalen Variablen gearbeitet.

Ausnahmen sind weiterhin die aktuellen Koordinaten (X, Y) des Schreibkopfs und die Position des Stifts (*PenUp*), sowie die Geschwindigkeitsvorgaben für die beiden Encoder-Motoren, die Konstanten *TempoX* und *TempoY*.

³ Sollte ein Befehl weniger Parameter benötigen, wird für die nicht erforderlichen Parameter der Wert „0“ angegeben.

⁴ In der im Download-Bereich der ft-Community verfügbaren [zip-Datei](#) mit dem neuen Steuerprogramm sind die auf fünf Spalten erweiterten Plot-Dateien der Version 1.1 enthalten.

Neue HP-GL-Kommandos

Die sechs neuen Befehle werden im Folgenden kurz vorgestellt – mit etwas mathematischem Hintergrund zum besseren Verständnis der Implementierung und der verwendeten Algorithmen. Wie dabei deutlich wird, ist eine Vektorgrafik-Implementierung eine sehr schöne Anwendung für trigonometrische Funktionen.

Selbstverständlich könnt ihr die Erläuterungen auch überspringen und gleich mit der neuen Programmversion herumexperimentieren. Die am Ende des Beitrags gezeigten Beispiel-Plots sind dafür ein guter Einstieg. Abb. 1 erläutert die wichtigsten Bezeichnungen.

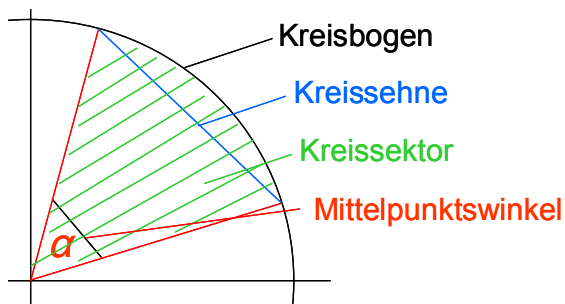


Abb. 1: Kreisbogen, Kreissehne, Kreissektor und Mittelpunktswinkel α

Die neuen HP-GL-Kommandos wurden gemäß der HP-GL-Spezifikation implementiert. Mit einer Einschränkung: Als Parameter werden ausschließlich ganze Zahlen akzeptiert, obwohl nach Spezifikation die Winkel als Fließkomma-Werte angegeben werden.

Diese Einschränkung habe ich vorgenommen, da ich anderenfalls alle Parameter als Fließkomma-Werte hätte einlesen müssen, denn Winkel können je nach Befehl an unterschiedlichen Stellen in der Parameterliste auftreten. Das hätte erheblichen zusätzlichen Speicherbedarf verursacht und eine befehlsabhängige Umrechnung der Koordinatenangaben in Ganzzahlen erfordert. Es zeigte sich aber, dass die Beschränkung auf ganzzahlige Winkel die Genauigkeit unseres Plotters nicht beeinträchtigt.

AA und AR

Die Befehle AA (*Plot Arc Absolute*) und AR (*Plot Arc Relative*) zeichnen ab der aktuellen Position des Zeichenstifts (X, Y) einen **Kreisbogen** bezogen auf den angegebenen Kreis-Mittelpunkt (X_C, Y_C) gegen den Uhrzeigersinn um den Mittelpunktswinkel α (*Alpha*):

$$\mathbf{AA} X_C, Y_C, \alpha [, \delta]$$

Beim AR-Kommando⁵ werden die Koordinaten von (X_C, Y_C) relativ zur Ausgangsposition (X, Y) angegeben:

$$\mathbf{AR} dx, dy, \alpha [, \delta]$$

Nach Durchführung des Kommandos steht der Schreibkopf an der Endposition des Kreisbogens (X_E, Y_E) .

Um den Kreisbogen plotten zu können, müssen wir zunächst aus den beiden gegebenen Punkten – der aktuellen Position (X, Y) und dem Kreis-Mittelpunkt (X_C, Y_C) bzw. dem Abstand (dx, dy) – sowie dem Mittelpunktswinkel α den Endpunkt des Kreisbogens (X_E, Y_E) berechnen (Abb. 2). Dazu benötigen wir ein wenig [Trigonometrie](#). Zum Glück stehen alle erforderlichen mathematischen Funktionen in Robo Pro als Operatoren zur Verfügung; inzwischen auch im Download-Modus.

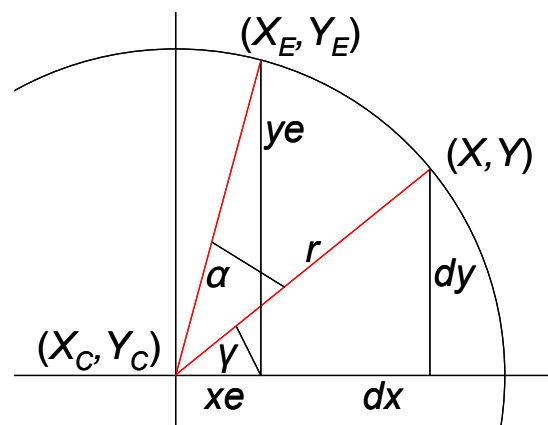


Abb. 2: Winkel und Punkte des Kreisbogens von (X, Y) bis (X_E, Y_E) zum Mittelpunkt (X_C, Y_C)

⁵ Parameter in eckigen Klammern ([]) sind optional, dürfen also fehlen (Wert = 0).

Aus dem Abstand (dx, dy) zwischen den Punkten (X, Y) und (X_C, Y_C) lässt sich zunächst der Radius r des Kreises bestimmen, denn dx und dy bilden die Katheten eines rechtwinkligen Dreiecks mit dem Kreisradius r als Hypotenuse. Nach dem Satz des Pythagoras (1) gilt:

$$dx^2 + dy^2 = r^2 \quad (1)$$

Da die Quadrate von dx und dy ab Werten von 180 den Zahlenbereich einer 16-bit-Zahl übersteigen, müssen wir sie vorher in Fließkommazahlen umwandeln. Die Berechnung des Kreisradius erfordert dann zwei Fließkomma-Multiplikationen und eine Quadratwurzelberechnung (2):

$$r = \sqrt{dx^2 + dy^2} \quad (2)$$

Aus dem Radius r und der Kathete dx können wir jetzt den ‚Startwinkel‘ γ des Kreisbogens berechnen (Abb. 2), denn in unserem rechtwinkligen Dreieck gilt:

$$\cos \gamma = \frac{x}{r} \quad (3)$$

Aber Achtung: Vom Punkt (X, Y) aus gesehen ist der Abstand dx zwischen X_C und X negativ, wenn X im ersten oder vierten, und positiv, wenn X im zweiten oder dritten Quadranten liegt. Damit folgt (4.1):

$$\gamma = \arccos\left(-\frac{dx}{r}\right) \quad (4.1)$$

Und noch ein Aspekt ist zu beachten: Die Cosinus-Funktion nur im Bereich von 0 bis 180° umkehrbar. Falls (X, Y) im dritten oder vierten Quadranten liegt ($dy \geq 0$), berechnet sich γ daher wie folgt (4.2):

$$\gamma = 360^\circ - \arccos\left(-\frac{dx}{r}\right) \quad (4.2)$$

Aus γ lässt sich der Endpunkt des Kreisbogens (X_E, Y_E) nun leicht bestimmen:

$$\begin{aligned} x_E &= r \cdot \cos(\gamma + \alpha) \\ X_E &= r \cdot \cos(\gamma + \alpha) + X_C \end{aligned} \quad (5)$$

$$\begin{aligned} y_E &= r \cdot \sin(\gamma + \alpha) \\ Y_E &= r \cdot \sin(\gamma + \alpha) + Y_C \end{aligned} \quad (6)$$

Der Kreisbogen von (X, Y) bis (X_E, Y_E) wird jetzt allerdings nicht, wie man vielleicht erwarten würde, exakt Punkt für Punkt geplottet, sondern durch **Kreissehnen** angenähert (Abb. 3). Die exakte Bestimmung jedes einzelnen Kreispunkts nach der allgemeinen Kreisgleichung (2) würde die Berechnung von

$$y = \pm\sqrt{r^2 - x^2} \quad (7)$$

für jeden ganzzahligen x -Wert zwischen $[-r, r]$ erfordern. Der Rechenaufwand wäre beträchtlich: Insgesamt wären $4 \cdot r$ Fließkomma-Multiplikationen und $2 \cdot r$ Wurzelberechnungen durchzuführen.

Durch die Annäherung des Kreisbogens mit Kreissehnen sinkt der Aufwand für die Punktberechnung erheblich – nur für die Endpunkte der Sehnen ist der jeweilige Kreispunkt genau zu bestimmen. Allerdings gilt: Je kürzer die verwendeten Sehnen, desto größer die Zahl der zu berechnenden Kreispunkte – und damit die Genauigkeit der Näherung.

Die Länge der Sehne wird durch den Mittelpunktswinkel δ (Delta) bestimmt, der als vierter Parameter des Befehls AA bzw. AR angegeben werden kann. Ist dessen Wert 0, wird nach der HP-GL-Spezifikation als Default-Winkel $\delta = 5^\circ$ gesetzt.

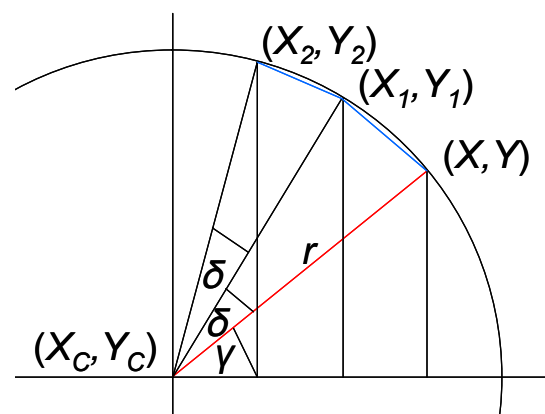


Abb. 3: Annäherung des Kreisbogens über Kreissehnen zum Winkel δ

Im allgemeinen Fall werden also die beiden Punkte (X, Y) und (X_E, Y_E) schrittweise

durch Kreissehnen zum Mittelpunktswinkel δ verbunden. Die Berechnung der Zwischenpunkte (X_i, Y_i) erfolgt analog der Bestimmung des Endpunkts (X_E, Y_E) des Kreisbogens:

$$X_i = r \cdot \cos(\gamma + i \cdot \delta) + X_{i-1} \quad (8)$$

$$Y_i = r \cdot \sin(\gamma + i \cdot \delta) + Y_{i-1} \quad (9)$$

Dies wird bis zum Index i_{\max} wiederholt, bei dem der Restwinkel kleiner ist als δ :

$$\alpha - i_{\max} \cdot \delta < \delta$$

Dann wird im letzten Schritt die Kreissehne zum Restwinkel $\alpha - i_{\max} \cdot \delta$ geplottet.

CI

Das CI-Kommando (*Circle*) erhält als Parameter den Radius r und plottet – durch Kreissehnen angenähert – einen **Vollkreis** um den aktuellen Punkt (X, Y) :

$$\text{CI } r \text{ [, } \delta \text{]}$$

Der Schreibkopf wird zum Schluss wieder am Kreismittelpunkt (X, Y) platziert. Auch hier kann optional ein Kreissehnen-Winkel δ (*Delta*) als zweiter Parameter übergeben werden; ist der Wert 0, wird als Default-Wert $\delta = 5^\circ$ verwendet.

Genau besehen ist der *Circle*-Befehl damit ein Vektorgrafik-Befehl für ein N-Eck. Falls δ ein ganzzahliger Teiler von 360° ist, dann plottet der Befehl sogar ein gleichseitiges N-Eck:

Winkel δ	N-Eck
120°	Gleichseitiges Dreieck
90°	Quadrat
72°	Regelmäßiges Fünfeck
60°	Regelmäßiges Sechseck
...	...
5°	72-Eck (Default)

Tab. 2: Zahl der Ecken des N-Ecks in Abhängigkeit vom Mittelpunktswinkel δ

Je kürzer die Sehne (also je kleiner δ), desto ‚runder‘ die Kreislinie – allerdings

sind dann auch mehr Hilfspunkte zu berechnen und die Zeichnung wird aufwändiger. Ob die Abweichung von der tatsächlichen Kreislinie mit bloßem Auge erkennbar ist oder nicht, hängt dabei nicht nur vom Mittelpunktswinkel δ der Sehne (bzw. der Sehnenanzahl) ab, sondern natürlich auch von der Größe des Kreises (also dem Radius r). Tatsächlich ist die Abweichung von einer exakten Kreislinie beim Default-Wert $\delta = 5^\circ$, also einem 72-Eck, sehr gering, wie sich leicht nachrechnen lässt.

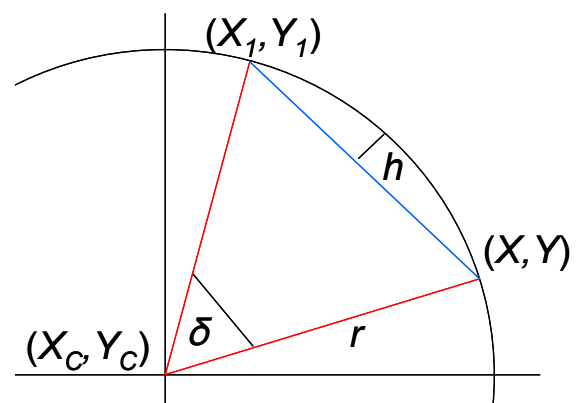


Abb. 4: Höhe h des vom Mittelpunktswinkel δ bestimmten Kreissegments

Denn die Höhe h der von den Kreissehnen gebildeten **Kreissegmente** (Abb. 4) kann nach der folgenden Gleichung aus δ und r bestimmt werden [4]:

$$h = r \cdot (1 - \cos \frac{\delta}{2}) \quad (10)$$

Bei einem Kreis (oder Kreisbogen) mit einem Radius von 1.000 Punkten liegt die maximale Abweichung h von einer exakten Kreislinie bei einem Sehnen-Mittelpunktswinkel von 5° bei knapp einem Punkt (genau: 0,95) – das entspricht 0,02 mm, ist also mit bloßem Auge nicht zu erkennen. Selbst bei einem von unserem Plotter maximal darstellbaren Radius von 2.250 Punkten liegt die maximale Abweichung unter drei Punkten (genau: 2,62), also bei etwa 0,06 mm – bei einer Linienstärke von ca. 0,4 mm (Kugelschreibermine).

Das bedeutet umgekehrt, dass bei kleineren Kreisen auch ein größerer Wert für δ ge-

wählt werden kann, ohne dass die Abweichung von der Kreislinie erkennbar ist. Aus (10) folgt mit ein paar einfachen Umformungen für einen festen Wert h_{\max} :

$$\delta = 2 \cdot \arccos\left(1 - \frac{h_{\max}}{r}\right) \quad (11)$$

Wollen wir eine maximale Abweichung h_{\max} von fünf Punkten (also etwa 0,1 mm) zulassen, dann darf bei einem Kreis mit einem Radius von 500 Punkten (= Kreisdurchmesser von ca. 2 cm) demnach der Sehnen-Mittelpunktswinkel δ bis zu 16° erreichen. Ohne Lupe ist in dieser Größe ein gleichmäßiges 24-Eck ($\delta = 15^\circ$) von einem Kreis nicht zu unterscheiden.

Neben der Annäherung mit Kreissehnen gibt es eine weitere Möglichkeit, das Plotten eines Kreises zu beschleunigen: mit einem von [Berthold Horn](#), heute Professor am MIT, im Jahr 1976 vorgestellten Verfahren [5]. Dabei wird zu jedem Punkt (x_i, y_i) der Kreislinie geprüft, ob $(x_i, y_i + 1)$ innerhalb eines ein Punkt breiten ‚Schlauchs‘ um den idealen Kreisbogen liegt. Wenn ja, wird dieser Punkt, sonst wird $(x_i - 1, y_i + 1)$ geplottet. Dazu genügt es, wie bei *Bresenham's* Algorithmus für schräge Linien [6, 2], einen Fehlerwert f mitzuführen, der größer als Null sein muss:

$$f = (x - \frac{1}{2})^2 + y^2 - r^2 \quad (12)$$

Zur Bestimmung des Fehlerwerts wird f mit dem Kreisradius r initialisiert ($f := r$) und dann bei jedem x -Schritt $2y + 1$ addiert. Ist $f \leq 0$, wird $2x - 2$ subtrahiert. Ein ähnlicher Algorithmus wurde knapp ein Jahr nach *Horn's* Veröffentlichung von *Bresenham* publiziert [7].

Damit lässt sich der Aufwand für die Berechnung der Kreispunkte auf ein bis zwei Ganzzahl-Additionen bzw. -Subtraktionen reduzieren – und so erheblich beschleunigen. Allerdings sind die Algorithmen von *Horn* und *Bresenham* nur für einen Achtelkreis (beginnend am Endpunkt eines Viertelkreises und in der Mitte des Viertel-

kreises endend) definiert; die Kreislinie lässt sich daher nicht an einem Stück, sondern nur in acht Kreisbögen ab je einem Endpunkt eines Viertelkreises plotten.

EA und ER

Die Befehle EA (*Edge Absolute Rectangle*) und ER (*Edge Relative Rectangle*) plotten **Rechtecke**, deren Seiten aus horizontal und vertikal verlaufenden Linien bestehen. Sie werden relativ zur aktuellen Cursor-Position geplottet, d. h. an den Befehl werden die absoluten bzw. relativen Koordinaten (X_C, Y_C) der gegenüberliegenden Ecke übergeben:

EA X_C, Y_C

ER dx, dy

Die Implementierung zeichnet (mit gesenktem Stift) zunächst eine Linie der Länge dx in horizontaler, dann eine der Länge dy in vertikaler, erneut eine in horizontaler ($-dx$) und dann wieder in vertikaler Richtung ($-dy$); der Cursor steht anschließend wieder an der Ausgangsposition (X, Y) .

EW

Der Befehl EW (*Edge Wedge*) zeichnet einen **Kreis Sektor**, ausgehend von der aktuellen Position (X, Y) als Mittelpunkt, zu gegebenem Radius r mit dem Mittelpunktswinkel α bezüglich des ‚Startwinkels‘ γ , d. h. dem Winkel der gegen den Uhrzeigersinn ersten Begrenzungslinie des Kreissektors zur X-Achse (siehe Abb. 1):

EW $r, \gamma, \alpha [, \delta]$

Wird kein Kreissehnenwinkel δ angegeben (also $\delta = 0$), wird der Kreisbogen aus Kreissehnen mit Mittelpunktswinkel 5° angenähert. Anschließend steht der Schreibkopf wieder am Ausgangspunkt (X, Y) .

Die Implementierung des Befehls ist einfach: Zunächst wird eine Linie zum ersten Kreispunkt im Abstand (dx, dy) mit $dx = r \cdot \cos \gamma$ und $dy = r \cdot \sin \gamma$ geplottet. Von dort wird mit dem Kommando

AR dx, dy, α [, δ]

ein Kreisbogen zum Endpunkt (X_E, Y_E) gezogen. Schließlich wird der Kreissektor durch eine Linie zum Mittel- und Startpunkt (X, Y) abgeschlossen.

Beispiel-Plots

Mit dem so erweiterten Befehlssatz lassen sich mit relativ wenigen Kommandos komplexe geometrische Grafiken plotten.

Um die Leistungsfähigkeit des Befehlsumfangs zu demonstrieren, habe ich beispielhaft ein paar optische Täuschungen in HP-GL ‚programmiert‘ (Abb. 5). Die HP-GL-Befehlsdatei, mit der die Plots erzeugt wurden, wurde aus einer Excel-Datei exportiert, die die Befehlskürzel entsprechend der Zuordnung aus Tabelle 1 automatisch durch Befehlsnummern ersetzt. Beide Dateien findet ihr, zusammen mit den an den neuen Parser angepassten HP-GL-Beispielen aus [2] und Version 2 des Steuerprogramms, im [Download-Bereich der ft:c](#).

Quellen

- [1] Fox, Dirk: *HP-GL-Plotter (Teil 1)*, [ft:pedia 4/2011](#), S. 26-34.
- [2] Fox, Dirk: *HP-GL-Plotter (Teil 2)*, [ft:pedia 1/2012](#), S. 4-12.
- [3] Hewlett-Packard: *The HP-GL/2 and HP RTL Reference Guide*. Second Edition, September 1996.
- [4] Wikipedia: [Kreissegment](#).
- [5] Berthold K. P. Horn: *Circle Generators for Display Devices*. Computer Graphics and Image Processing 5, 2 (June 1976), S. 280-288.
- [6] Wikipedia: [Bresenham-Algorithmus](#).
- [7] Jack Bresenham: *A linear algorithm for incremental digital display of circular arcs*. Communications of the ACM 20, 2 (1977), S. 100-106.

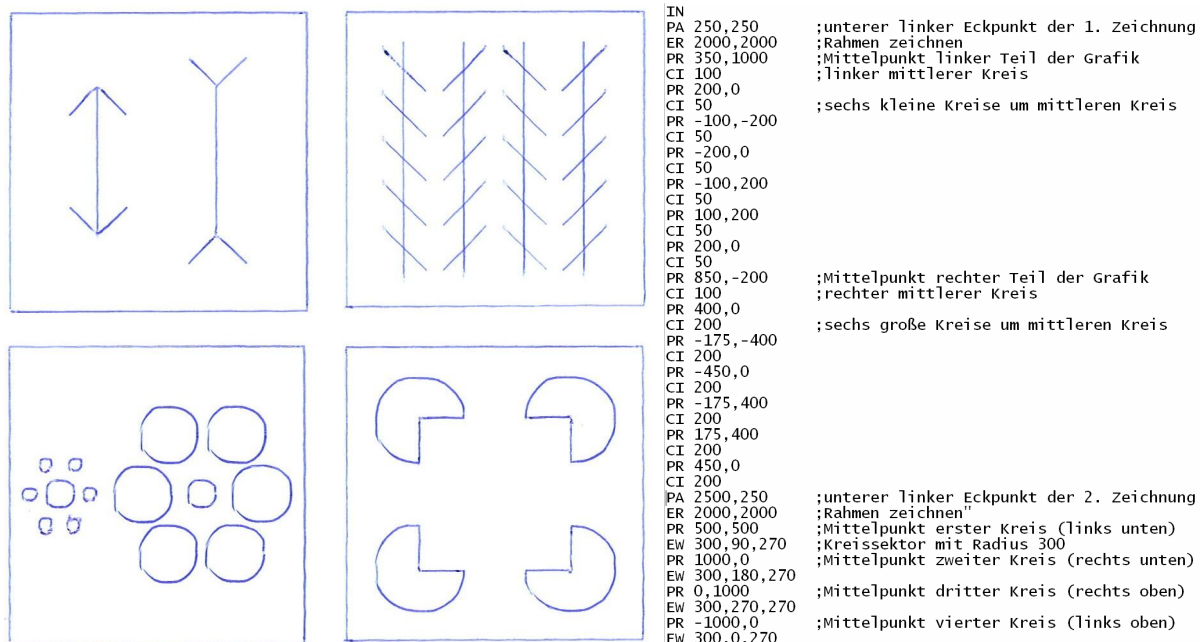
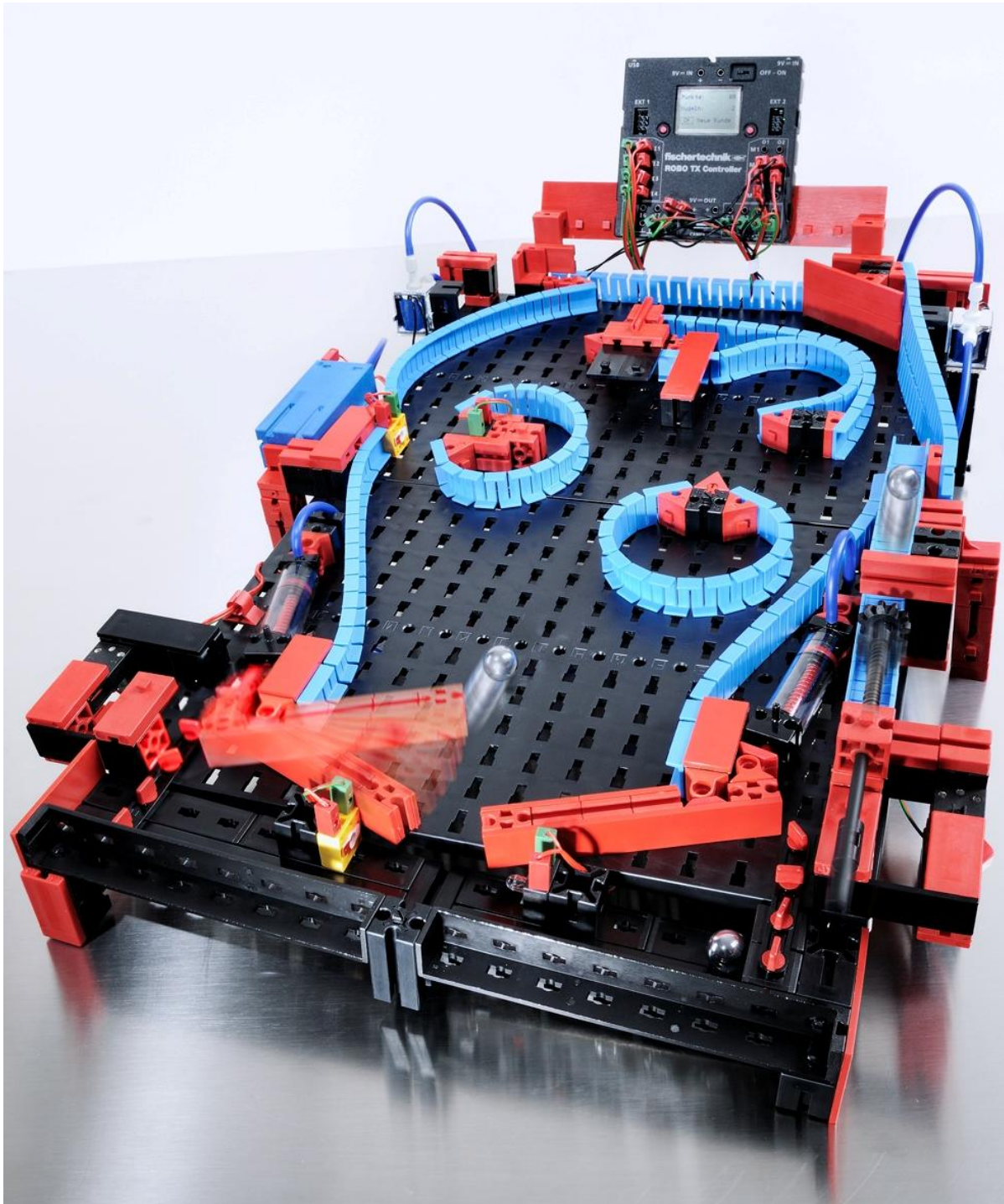


Abb. 5: Optische Täuschungen – links der Plot, rechts die HP-GL-Datei der beiden unteren Zeichnungen



*Der Flipper – aus dem neuen fischertechnik-Kasten
[Robo TX Electro Pneumatik](#) (Foto: fischertechnik)*