

Editorial

## Von Menschen und Modellen

Dirk Fox, Stefan Falk

Die Entstehung eines fischertechnik-Modells kann ganz unterschiedlich verlaufen. Die eine oder der andere steckt zuerst Teile zusammen, probiert, nimmt sie wieder auseinander, kombiniert sie neu, spielt damit, baut eine klemmende Stelle wieder und wieder um – bis das Werk den Ansprüchen genügt. Manche zerlegen es gleich wieder, andere spielen eine Weile damit, wieder andere stellen es ins Regal oder verbessern ihre Schöpfung sogar über Jahre.

Eine andere Spezies geht Wochen und Monate mit einer Idee schwanger, plant im Kopf dieses Prinzip und jenes Detail, fertigt gar Zeichnungen, Berechnungen und Diagramme an, bevor der erste Baustein angefasst wird. Dann wird der Plan abgearbeitet, dies und das im Lichte der Einsicht korrigiert, dass von der schönen Theorie doch nicht alles in der Praxis funktioniert, und tastet sich so ans Ziel.

Solange ein Modell höchstens der Demonstration auf einem fischertechnik-Treffen dient, genießen beide Herangehensweisen einen großen Luxus: Sie dürfen sich nach Herzenslust im eigenen Fundus bedienen, Originalteile veredeln und Fremdteile hinzumischen. Niemand stört es, wenn der letzte Schrei aus Waldachtal genutzt wird, den noch kaum jemand besitzt. Längst nicht mehr produzierte und sogar gebraucht vergriffene Teile, in großer Zahl verbaut, ernten sogar offene Bewunderung. Schlimmstenfalls wird der Erbauer auf ein seltenes Teil angesprochen, das jemand noch nie gesehen hat.

Das ändert sich schlagartig bei Modellen, die zum Nachbauen anregen sollen oder

sogar genau dafür bestimmt sind. Große Freiheiten kann man sich nicht mehr leisten, will man nicht einen Großteil der „Kundschaft“ aussperren. Darf man Spezialteile aus vergangenen Jahrzehnten verwenden, die jüngere fischertechniker kaum kennen, geschweige denn besitzen? Darf man die Verfügbarkeit neuer Bauteile aus den aktuellen Kästen voraussetzen? Kann man bedenkenlos jede Controller-Generation nutzen, auch wenn sich vielleicht viele diese Produkte nicht leisten können? Jede Schritt-für-Schritt-Anleitung riskiert, dass viele aufgeben, wenn ihnen ein Bauteil fehlt.

Zum Glück für die Autoren – und hoffentlich auch für die Leser – ist die Förderung des Tüftelns erklärtes Ziel der ft:pedia, und nicht der rezeptartige Nachbau. Anders als Baukasten-Anleitungen zielen wir auf das Verstehen der Funktionsweise eines Modells – und daher ist es kein Mangel, sondern Programm, wenn ihr mit eurem Teilebestand eine abgewandelte Konstruktion entwickeln müsst. Die ft:pedia ist eine einzige Zumutung: Ihr müsst Texte lesen, Bilder im Detail betrachten, über die Funktionsweise eines Modells nachdenken und es dann umbauen – oder sogar verbessern! Aber erst so entfaltet der Genuss des tüftelnden Forschens seine ganze Wirkung.

In diesem Sinne: Schöne, friedvolle Weihnachten – und viel Zeit für eure Modelle!

Beste Grüße,  
euer ft:pedia-Team

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

## Inhalt

Von Autoren, Menschen und Modellen .....	2
Achteck-Variationen.....	4
Die elektrifizierte Mini-Eisenbahn (Spur N) .....	10
Spielautomat .....	12
Tür zu, bitte! .....	17
Gottwald MK500 .....	26
Anwendungen für Magneten (3): Magnetskupplungen und Magnetrührer .....	31
Anwendungen für Magneten (4): Lautsprecher.....	39
Der Elektromagnet: Was kann er (vertragen)?.....	46
Funktionsmodelle von Gleich- und Wechselstrommotoren .....	52
Auf zu neuen Ufern: Die Geschichte der „Community- Firmware“ für den TXT .....	59
Frische Apps für den TXT-Controller .....	68
Ein universeller I <sup>2</sup> C-Adapter für den TX(T).....	72
LED-Backpack im Retrodesign .....	80
I <sup>2</sup> C mit dem TX(T) – Teil 14: LED-Display (2) .....	84

## Impressum

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

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

**Autoren:** Christian Bergschneider, Gerhard Birkenstock,  
Daniel Canonica, Stefan Falk, Dirk Fox, Stefan Fuss,  
Andreas Gail, Ralf Geerken, Till Harbaum, Peter Krijnen,  
Rüdiger Riedel, René Trapp, Martin Westphal.

**Copyright:** Jede unentgeltliche Verbreitung der unver-  
änderten 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.

Tipps &amp; Tricks

## Achteck-Variationen

Ralf Geerken, Martin Westphal, René Trapp

*Es könnte so einfach sein, etwas Achteckiges mit fischertechnik-Teilen zu bauen. Es fehlt allerdings bislang an Winkelsteinen 45° oder Phantasie.*

Seit über 50 Jahren gibt es nun fischertechnik, und bis vor kurzem fehlte tatsächlich ein Winkelstein 45°. Jetzt gibt es Dank des 3D-Druckers und einer passenden Datei zwar die Möglichkeit, sich selbst zu helfen, aber so ein richtig echter Winkelstein 45° aus Waldachtaler Produktion ist das nicht.

Am Rande der Maker Faire (Hannover 2016) ergab es sich bereits, dass die Autoren in Ralfs Hobbykeller beisammensaßen. Martin berichtete von Problemen beim Bau seines Riesenrades [1]: Seine zweite Nabenkonstruktion (Abb. 1) basiert auf einer achteckigen Geometrie.

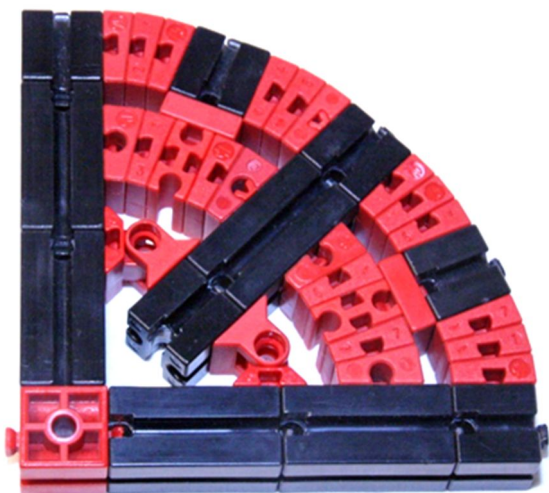


Abb. 1: Ein Viertel der Riesenradnabe 2.0

Allerdings war es sein Ziel, ein 16-Eck zu bauen, um 16 Speichen für das Riesenrad zu bekommen. Wie man sieht, wird der Winkel von 90° zweimal halbiert. Aufgrund der vielen Verbindungsstellen ist die Stabilität

recht gering. Kräfte parallel zur Drehachse verschieben die zahlreichen Winkelsteine gegeneinander, und nur vier der 16 Speichen sind mit dem Achslager (der Baustein 15 mit Loch – BS15-L) fest verbunden.

Auf Ralfs Arbeitstisch lag just an diesem Abend seine interessante achteckige Grundkonstruktion, bestehend aus je acht Winkelsteinen 60° ([31010](#)) und Winkelsteinen 15° ([31981](#)).



Abb. 2: Achteckring aus Winkelsteinen

Mit nur 15 mm Baubreite ist dieser Ring extrem schmal, und auch der Durchmesser ist minimal. Die Konstruktion bietet im Inneren nicht nur jede Menge Platz für Durchführungen aller Art; sie besitzt genau die richtige Öffnungsweite für einen BS15 (z. B. [31005](#)), BS30 (z. B. [31003](#)) oder ein entsprechendes Aluprofil (z. B. [31220](#)).

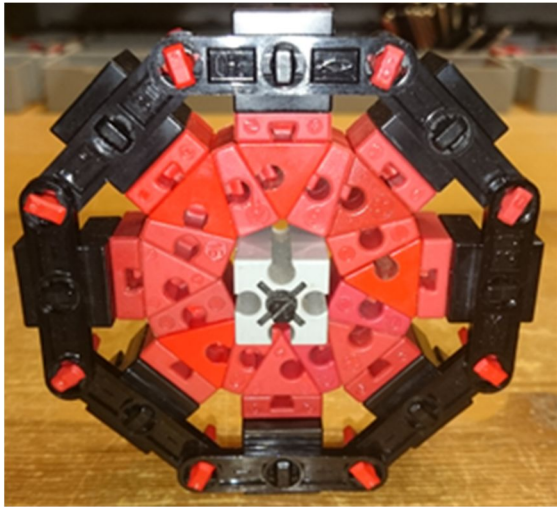


Abb. 3: Achteckring mit Möglichkeiten

Dabei gleitet der achteckige Ring sehr leicht und absolut klemmfrei auf dem verwendeten BS30, wobei ein Drehmoment jedoch formschlüssig übertragen wird! Wofür man das alles einsetzen könnte, sei der Phantasie des Lesers überlassen. Spontan kommt einem Autor die Anwendung an einem Fahrgeschäft-Modell in den Sinn.

In Renés Schatzkiste findet sich ebenfalls ein Achteck. Seit 2014 ist es im Einsatz, allerdings nicht als Nabe eines Riesenrades, sondern für eine Stützkonstruktion mit acht Armen.

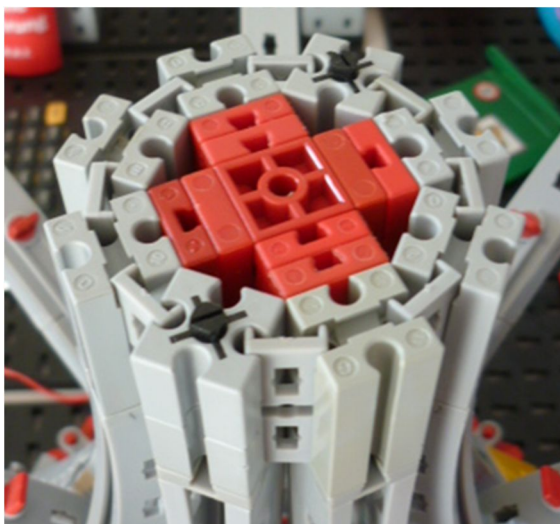


Abb. 4: Achteckiger Sockel

Die Baubreite ist mit mindestens 30 mm allerdings doppelt so breit wie bei Ralfs Umsetzung. Auch der Durchmesser ist im

Vergleich größer. Zusammengesetzt wird dieses Achteck aus zwei identischen klauenförmigen Hälften. Zentral wurde ursprünglich je Klauenhälfte ein BS15 mit rundem Zapfen (z. B. [31059](#)) verwendet, nach außen sitzen je vier BS5 mit zwei Zapfen ([37238](#)) und BS5 ([37237](#)) aufeinander. In Abb. 4 ist allerdings eine Version mit BS15-L anstelle des BS15-2Z zu sehen. Es ergibt sich zunächst ein kreuzförmiges Kernstück. Zwei oder auch mehr dieser Kreuze werden mit den runden Zapfen in das nächste Kreuz eingeschoben und abwechselnd um etwa 45° gegeneinander verdreht; Augenmaß ist hier ausreichend. Winkelsteine werden nicht benötigt, BS30 (z. B. [31003](#)) schließen jede Klauenhälfte nach außen hin ab. Sie werden erst aufgeschoben, nachdem die Kreuze miteinander verbunden sind. Für einen zusätzlichen festen Halt werden die Klauenhälften mit Statiklaschen 15 (z. B. [36326](#)) in den Nuten der BS30 zusammengesteckt.

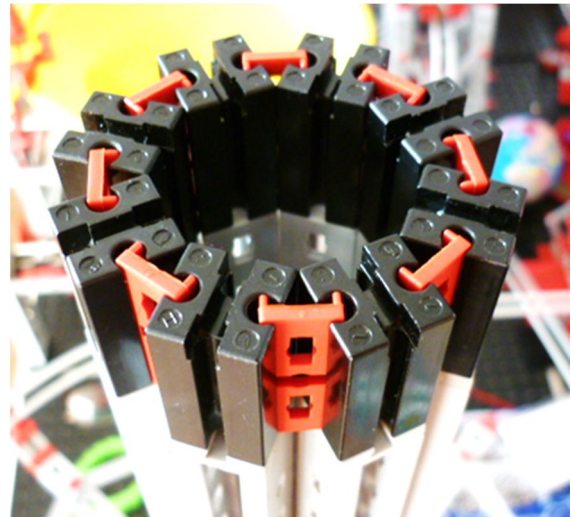


Abb. 5: Bausteine 30 und Statiklaschen 15

Wie man in Abb. 5 sieht, geht das auch als hohler Ring. Die Stabilität gegen radial einwirkende Kräfte ist dann allerdings gering. Die Geometrie kann beliebig verlängert werden und bildet im Beispiel den Schaft eines Turmes.

Beim Übergang auf eine Zwölfeckstruktur passen die unterschiedlichen Grundgeome-

trien an vier Seiten durchaus maßhaltig aneinander. Abb. 6 zeigt Verbinder 45, die in vier der acht BS30 stecken. Das Zwölfeck liegt auf dem Achteck auf, die vier Verbinder 45 ragen in jeweils vier der 12 BS5 und halten so das Zwölfeck auf dem Sockel. Das Zwölfeck besteht aus jeweils sechs Stück WS60° und sechs Stück BS15 in der bekannten Ringkonstruktion. Zusätzlich sind auf den 12 Außenflächen noch BS5 angebracht.

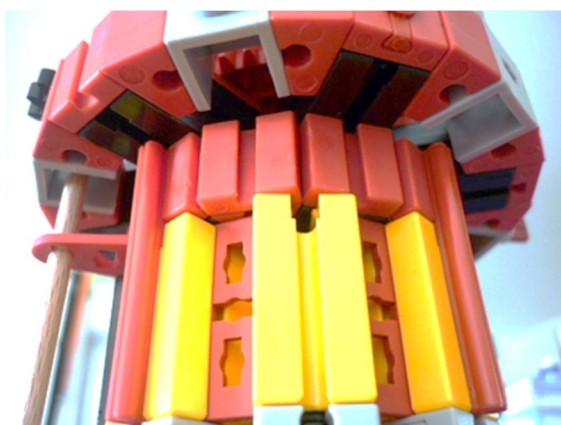


Abb. 6: Übergang von Achteck auf Zwölfeck

Ursprünglich für die Stützarme der Kugelbahn-Laufschienen erdacht, kann dieses Achteck auch mit einer Aufnahme für Metallachsen konstruiert werden. Zu diesem Zweck werden die BS15 mit rundem Zapfen durch BS15-L (z. B. [32064](#)) ersetzt. Zwei BS5 mit zwei Zapfen sowie sechs BS5 vervollständigen den Kern des Ensembles.

Die Stabilität der Konstruktion ist sehr hoch, und so eignet sie sich auch als achteckige Nabe für schwere Objekte. Ungelöst ist hier bislang noch die Übertragung von – nicht allzu großen – Drehmomenten auf die Metallachse. Aber eines ist noch interessant: Anstelle der Statiklaschen würden tatsächlich 3D-Druck-Winkelsteine zwischen die BS30 passen und könnten die Statiklaschen ersetzen. Die Stabilität der Anordnung würde sich dadurch nochmals deutlich erhöhen.



Abb. 7: Leuchtturm mit achteckigem Sockel

Spontan übernahm Martin eine der Ideen, um eine neue, stabilere Nabe für das Riesenrad zu konstruieren.

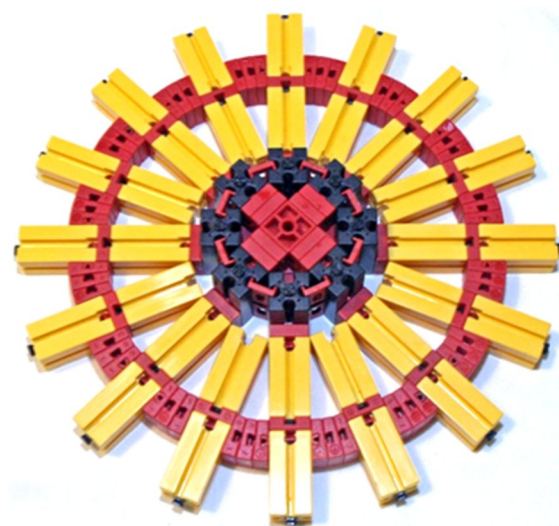


Abb. 8: Riesenradnabe 3.0

Bei diesem Kranz (Abb. 8) bildet das Achteck aus BS30 den stabilen Kern und trägt direkt acht Speichen. Weitere acht Speichen sind allerdings wieder nur mit Winkelsteinen fixiert und biegen sich unter Last seitlich weg.

Darüber noch nicht müde geworden, verabredeten die Autoren, zur Convention ihre Achtecke für einen gemeinsamen Fototermin mitzubringen. Im Trubel der Convention blieben die „Models“ dann anschließend unbeachtet auf dem Tisch liegen und es fanden sich andere kreative Köpfe, die mit den Strukturen etwas experimentierten. Eigentlich sollte wohl nur ein Turm aus den Achtecken entstehen, und beim Stapeln gerieten durch Zufall zwei Achtecke ineinander – nahezu exakt passend (Abb. 9). Die Bildqualität entspricht nicht dem Anspruch der ft:pedia, allerdings ist es das einzige existierende Foto, entstanden während der Convention 2016.

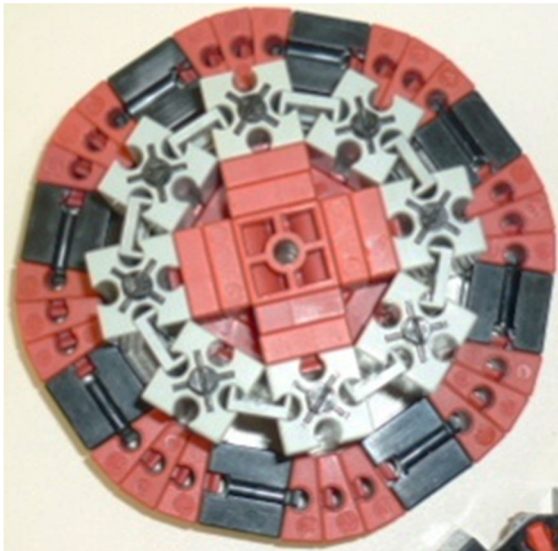


Abb. 9: Achtecke im Verbund

Der äußere Ring (schwarz/rot, Ralf) gleitet axial auf dem inneren Ring (grau/rot, René) entlang. Die Achtecke können sich auch – leicht schleifend allerdings – ineinander drehen. Ersetzt man den BS15-L durch eine Schneckenmutter, ergibt sich schon die nächste Kaulquappe für etwas Ausfahrbares – dafür sind die beiden Konstruktionen natürlich in der Höhe deutlich zu verlängern. Mittels der Lücken in und zwischen den Ringen lassen sich sogar Seilzüge, Kabel oder Schläuche unterbringen.

Nun sind dies nicht die ersten und einzigen Versuche, aus fischertechnik-Teilen etwas

Achteckiges zusammenzusetzen. Im Bilderpool finden sich weitere dokumentierte Ideen für achteckige Naben oder Konstruktionen [2, 3], zum Teil allerdings mit selbstgebauten Komponenten.

Eigentlich könnte die Geschichte jetzt hier enden, aber Ralf erleidet eine heftig kreative Phase und (er)findet noch mehr achteckige Konstruktionen. Im Prinzip bestehen auch diese meist aus klauenförmig zusammengesteckten Hälften. Allerdings sind auch schmale Ringe dabei. Abb. 10 zeigt zunächst eine Version, bei der die Speichen an zwei Positionen miteinander verbunden sind.

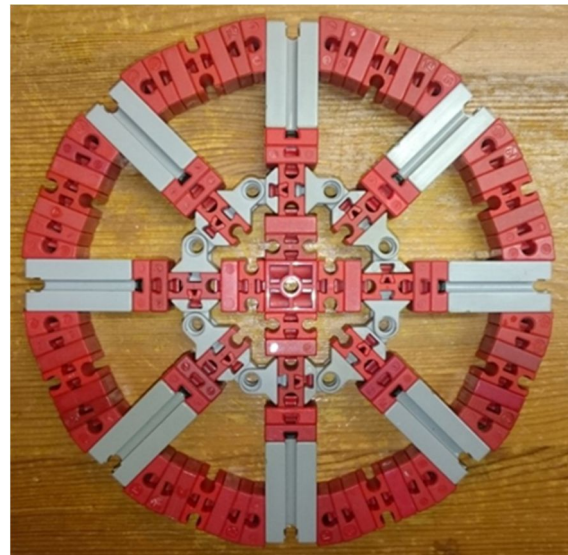


Abb. 10: 16-Eck-Nabe mit innenliegender Achteck-Nabe

Die folgende Variante bietet rasterkonforme Abmessungen auf den Armen. Diese Version (Abb. 11) besteht aus zwei identischen Viereck-Hälften und hält einfach dadurch zusammen, dass beide zusammengesteckt werden.

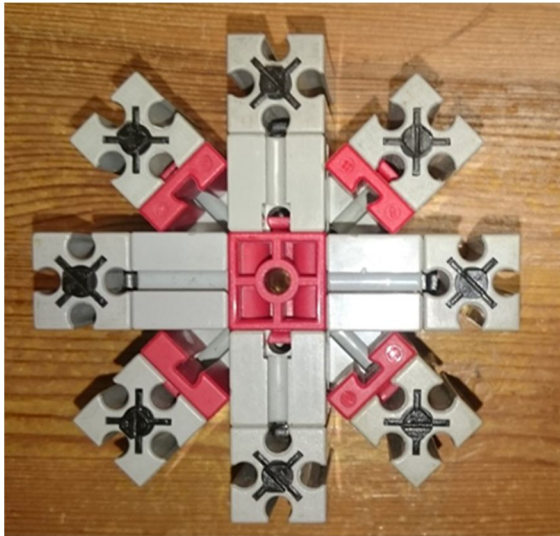


Abb. 11: Variante mit zwei steckbaren Hälften

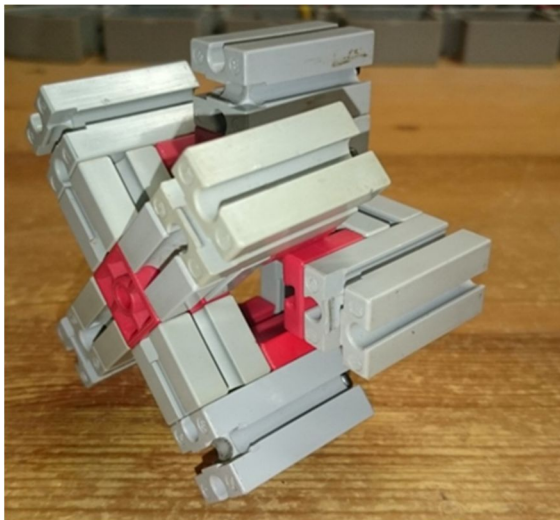


Abb. 12: Die beiden Klauenhälften

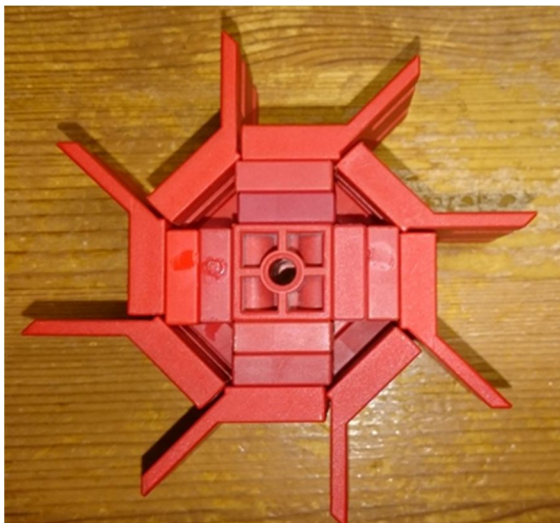


Abb. 13: Schaufelrad

Die Idee mit den Klauenhälften scheint Potenzial zu haben. Abb. 13 enthüllt ein Schaufelrad mit, wer hätte es gedacht, acht Schaufeln.

Hier sind die Klauen nicht ganz identisch, auch wenn sie jeweils die gleichen Teile verwenden. Bei einer Hälfte müssen die Schaufeln andersherum montiert werden.

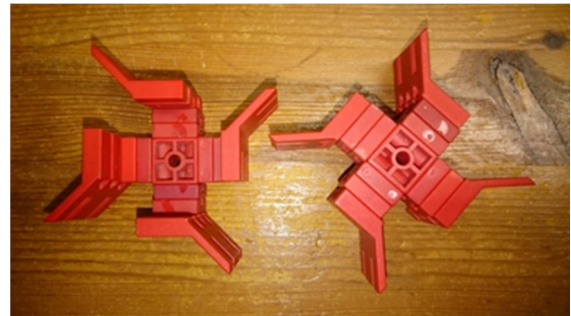


Abb. 14: Schaufelradhälften

Mit Speichenrädern, BS15 und Achsen gelingt auch die sehr ungewöhnliche Konstruktion in Abb. 15. Naheliegender ist es, die Speichenräder zu verwenden, da jedes acht Speichen besitzt. Werden sie entsprechend gegeneinander verdreht, ergibt sich eine Art 16-eckiger Grundriss. Besondere Beachtung verdient die Anordnung der Achsen, die jeweils knapp 11 mm Abstand zueinander haben.



Abb. 15: Zwei Speichenräder werden zum 16-Eck



Die beiden Hälften werden hier durch die Kunststoffachsen miteinander verbunden. Diese Anordnung kann daher auch auseinandergezogen werden, um weitere Elemente im Zwischenraum unterzubringen.

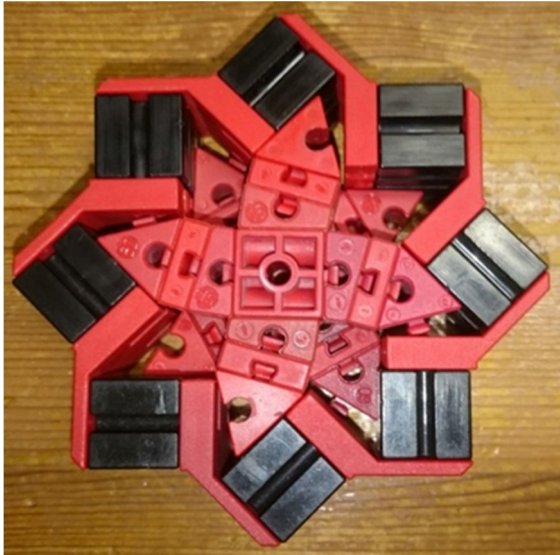


Abb. 16: Weihnachtliches Achteck

Weihnachtlich mutet die Konstruktion an, mit der dieser Artikel seinen Abschluss finden soll (Abb. 16):

Dieses Achteck sieht nicht nur elegant aus, die beiden Hälften sind durch die V-Schaufelhalter (so heißen diese 45°-Winkel offiziell) fest miteinander verbunden. Allerdings wird es wegen der verkippten BS15 nicht ganz einfach, dieses Achteck als Nabe mit Speichen zu verwenden. Ein prima Grundstock für ein großes Schaufelrad ist es aber allemal.

Erwähnenswert ist schließlich das auf Rollenböcken beruhende Achteck aus Haralds „Kaulquappen“ in der ft:pedia 2/2011 [4].

Fortsetzung folgt...

### Quellen

- [1] Westphal, Martin: [Riesenrad 2015](#) im ftc Bilderpool.
- [2] Baumgart, Kai: [Kastanienmühle](#) im ftc Bilderpool.
- [3] Brickwedde, Wilhelm: [Riesenrad](#) im ftc Bilderpool.
- [4] Steinhaus, Harald: *Kaulquappen (Teil 2)*. ft:pedia 2/2011, S. 9-13.

## Tipps &amp; Tricks

## Die elektrifizierte Mini-Eisenbahn (Spur N)

Gerhard Birkenstock

*fischertechnik* hatte im Juli des vergangenen Jahres zum 50. Geburtstag nach Waldachtal eingeladen. Diesem Ruf sind viele bis zur Spritzgießmaschine gefolgt. Und wer etwas Geduld mitbrachte, konnte sich einen Bausatz der überarbeiteten kleinen Mini-Eisenbahn (Spur N) aus dem Jahre 1979 mitnehmen. Nur zum Hinstellen ist diese aber zu schade.

Eines Abends ist mir in meiner Post mal wieder eine Verkaufsbroschüre in die Hände gefallen. Lokomotiven, Waggon, kleine Häuser. Ah, der Winter kommt: Die Kaufhäuser wollen unsere freie Zeit mit nützlichen Dingen bereichern.

Beim Durchblättern ist mir ein motorisiertes Spur-N-Lok-Chassis ohne Aufbau aufgefallen. Das ist mal was Schönes: Man kann kreativ werden. Nicht nur einfach kaufen und fahren lassen (das hat mich noch nie ausgefüllt). Nur, welcher Aufbau soll aufgesetzt werden?

Ein paar Tage später bin ich an den Tüten der kleinen fischertechnik Mini-Eisenbahn vorbeigekommen. Da kam mir das Lok-Chassis wieder in den Sinn.

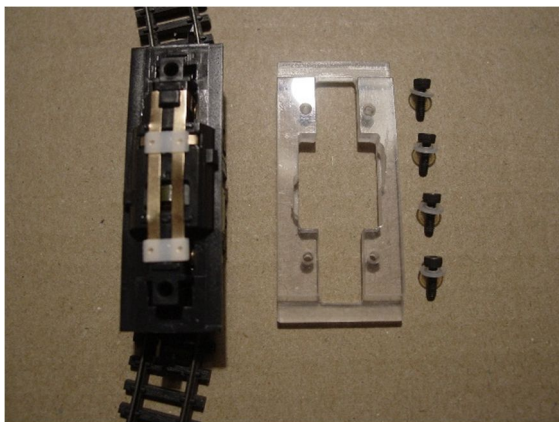


Abb. 1: Motorisiertes zweiachsiges Chassis der Spur N im Maßstab 1:160 bei Conrad Electronic ([Best.-Nr. 246473](#), Firma KATO, Japan)

Leider gab es keine technischen Zeichnungen zu dem Angebot. Daher musste ich mir das Chassis mit der Ungewissheit eines möglichen Fehlkaufes zulegen...

Bis das Chassis eintraf, hatte ich in der Zwischenzeit die fischertechnik-Lok mit Anhänger zusammengesetzt.

Die Länge von Chassis und Lok passten recht gut zusammen. Die Zusammenfügung geht aber nicht ohne Adapter: Bei der Lok wird angeklickt, bei der fischertechnik-Lok wollte ich schrauben. Und das, ohne den fischertechnik-Teilen etwas anzutun.

### Realisierung

Abb. 2 zeigt den von mir entworfenen Adapter als Maßzeichnung.

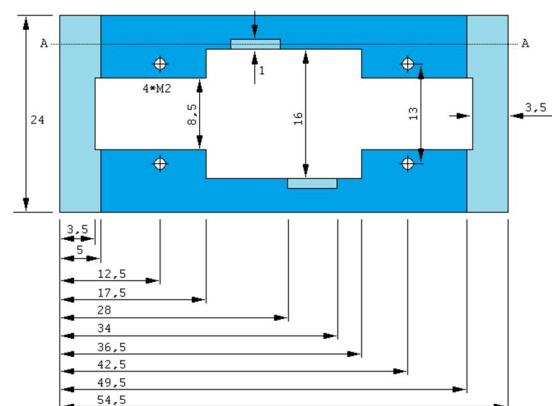


Abb. 2: Draufsicht mit Bemäßung

Das Chassis hat zwei diagonal angeordnete Nasen. Diese sind zum Anklipsen des

Chassis gedacht; auf der Zeichnung oben mit 1 mm bezeichnet. Der Aufbau des Motors und der Antrieb zu den Achsen nehmen recht viel Raum in Anspruch. Der Adapter wird dadurch an zwei Stellen recht dünn. Da der Adapter aus Polycarbonat (PC) hergestellt ist, ist die Festigkeit trotzdem gut.

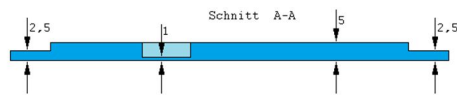


Abb. 3: Schnittdarstellung an Position -A-

Leider ist der Kondensator am Motor der KATO-Lafette im Weg. Den Klemmstein habe ich entfernt und den Kondensator abgenommen. Die Lok wird dadurch etwas mehr elektromagnetische Störung abgeben.

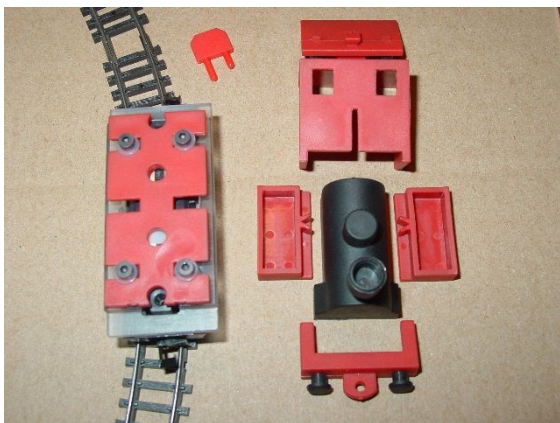


Abb. 4: Lok vor dem Zusammenbau

In Richtung fischertechnik wird die Grundplatte mit vier Schrauben M2×6 mm am Adapter angeschraubt. Die Unterlegscheiben aus Kunststoff (D = 5 mm, d = 2 mm,

h = 0,8 mm) verhindern unschöne Abdrücke auf der fischertechnik-Grundplatte. Wenn man die Positionen der Schrauben übernimmt, können die Aufbauten der Lok ohne Verkleben aufgesteckt werden.

Da der Lok-Aufbau nur um 5 mm angehoben wurde, können noch immer die Steckkupplungen mit den Stiften von fischertechnik verwendet werden. Gleichzeitig schauen auch noch die Haken der Spur N am Chassis heraus. Somit können die Waggon der Spur N parallel verwendet werden. Es passt alles zusammen.

### Tipps zum Betrieb

Die Waggon der fischertechnik-Bahn haben für die Spur N sehr schlechte Laufeigenschaften: Der Schwerpunkt liegt ungünstig weit oben, Kurvenfahrten sollte man daher nur vorsichtig durchführen. Rückwärtsfahren durch eine Kurve geht nicht, Rückwärtsfahren in gerader Richtung ist machbar.

Ein weiteres Problem sind die Randbildungen, die beim Spritzen an den Formrändern der Spritzgießmaschine entstehen. Diese Ränder habe ich mit einem Messer an den Achsen und auf den Laufflächen der Räder entfernt. Die Waggon laufen nun leichter und gleichmäßig. Etwas Nivea-Creme auf der Achse verbessert die Laufeigenschaften noch erheblich.

Na, wer baut es nach? Vielleicht mit einem der aktuellen 3D-Filament-Drucker?

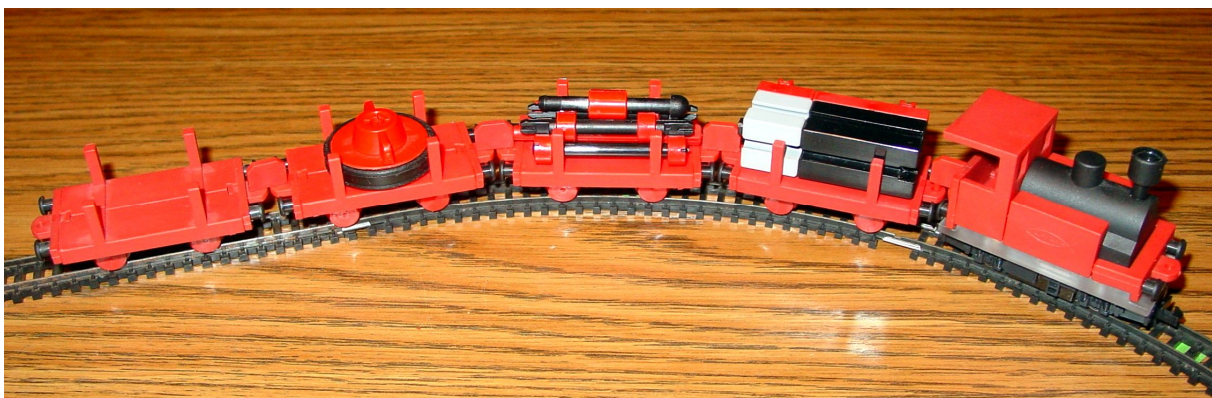


Abb. 5: Güterzug mit elektrifizierter fischertechnik-Lok

Modell

# Spielautomat

Daniel Canonica

*Die „Einarmigen Banditen“ mit dem magischen Klimpern der Münzen haben wohl viele Menschen in den Ruin getrieben und einigen unerwartetes Glück gebracht. Mit fischertechnik kann man schöne Modelle davon bauen, elektromechanisch oder elektronisch gesteuert.*

Dieses Modell hat eine sehr, sehr lange Geschichte. So konnte man 1975 eine Anleitung mit dem folgenden Titel bestellen:



Abb. 1: Bauanleitung von 1975

Leider fehlten mir damals nicht nur ein paar, sondern viele Teile, um auch nur annähernd ein solches Modell zu bauen:

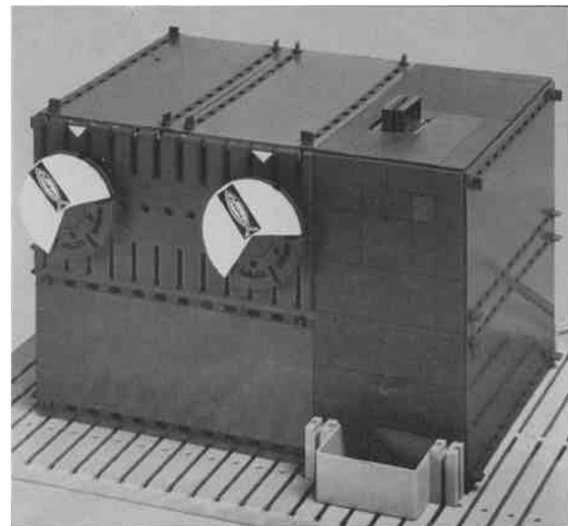


Abb. 2: Das Ur-Modell von 1975 [1]

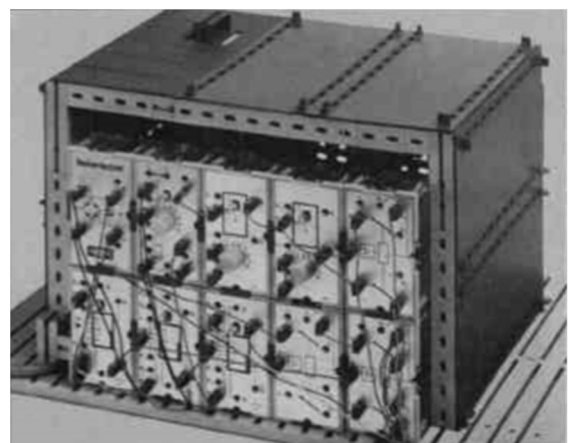


Abb. 3: Die Rückseite mit der Elektronik

41 Jahre später und um ein paar Erfahrungen reicher habe ich im Bilderpool auf [ftcommunity.de](http://ftcommunity.de) einige schöne Modelle entdeckt. Nun war der Antrieb da, selber etwas zu bauen.

## Etwas Geschichte

Der sogenannte *Einarmige Bandit*, auch *slot machine* genannt, wurde bereits im 19. Jahrhundert erfunden und rasch weiterentwickelt. Hier ist eines der ältesten Exemplare abgebildet:



Abb. 4: Einarmiger Bandit, 19. Jh. [2]

Schon bald wurden die geldgierigen Geräte in vielen Bundesstaaten der USA verboten, was auch dazu führte, dass es immer mehr Spielfreudige und -süchtige nach Las Vegas zog.

## Das Modell

Mein erster Ansatz war ein rein elektromechanisches Modell mit drehenden Walzen, verschiedenen Farben und sehr vielen Tastern. Dies funktionierte leidlich.

Im zweiten Anlauf habe ich Lampen eingebaut und die alten Silberlinge hervorgeholt. Die beiden von Motoren angetriebenen Drehscheiben mit aufgesetzten BS15 bedienen in rascher Folge die Taster, wodurch die Lampen von rot zu grün und zurück wechseln.

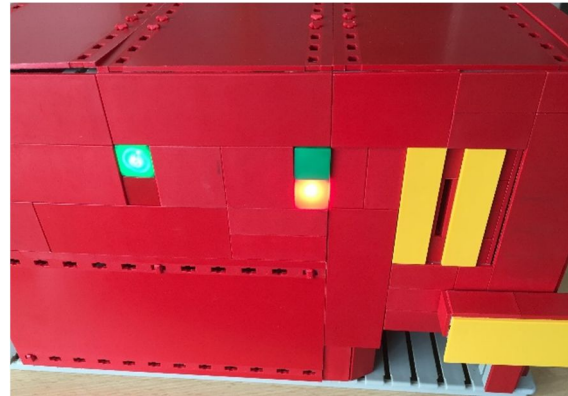


Abb. 5: Der Spielautomat, rechts der Geldeinwurf und der Hebel zum Beenden des Spiels

Wenn beim Stopp beide Lampen auf Grün stehen, hat man gewonnen und die Kassette mit dem bisher eingeworfenen Geld wird ausgefahren. (Die Maschine behält, im Gegensatz zu kommerziellen Geräten, nichts für sich!)



Abb. 6: Der Geldbehälter ausgefahren

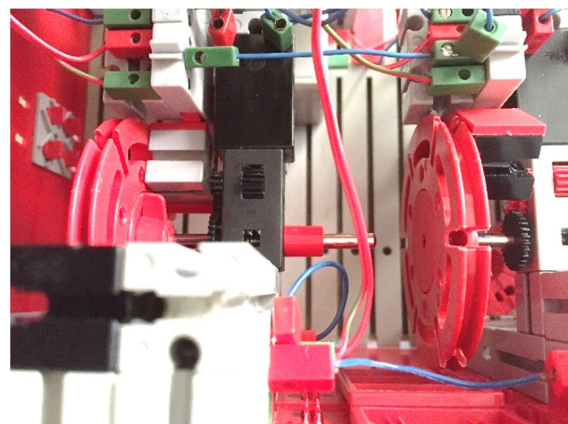


Abb. 7: Motoren und Ansteuerung der Taster

Ich musste pro Drehscheibe zwei Minitaster hintereinander schalten. Einer bedient die Lampen, der zweite schaltet das Signal für die Gewinnstellung durch. Bei Verwendung eines AND-Bausteins würde ein Taster pro Drehscheibe genügen.

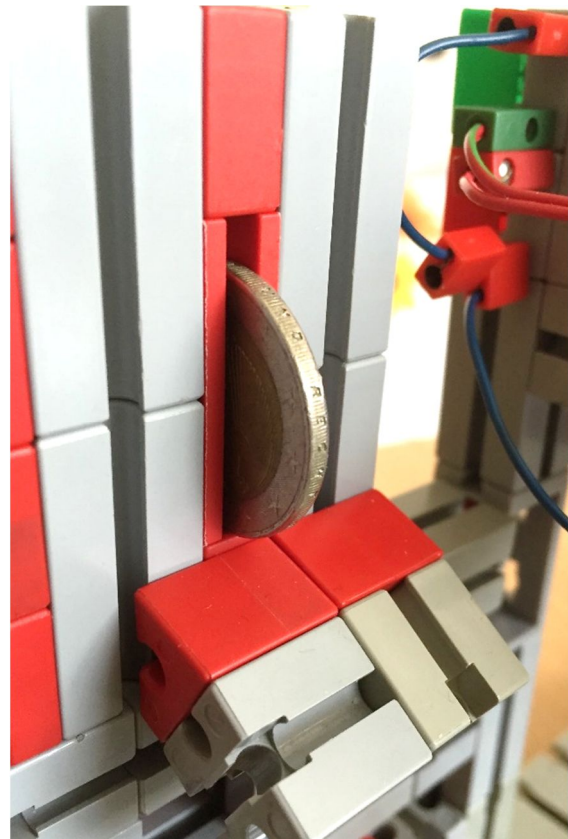
Natürlich kann man die Gewinnchancen erhöhen oder verringern, indem mehr oder weniger Bausteine auf die Drehscheiben gesteckt werden.

Beim Münzeinwurf fällt die Münze in einem 30°-Winkel seitlich auf den Taster. Unten wird sie von einem Statikverbinder festgehalten, der an einem simplen Hebel angebracht ist (nicht ganz im Stile des Einarmigen Banditen, aber mechanisch und vom Platzbedarf her einfacher zu realisieren).

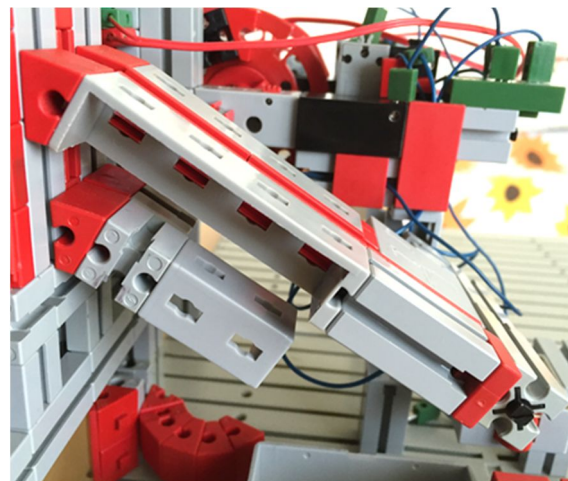


*Abb. 8: Münzeinwurf und Start/Stopp-Taster (von oben)*

Sobald ich den Hebel ziehe, fällt die Münze in den Behälter und der Start/Stopp-Taster geht wieder in den Ruhezustand. Ich betreibe den Mechanismus mit 2-Franken-Münzen, es funktioniert aber auch mit 2-Euro-Münzen. Man muss den Taster genau justieren, damit die Münze nicht eingeklemmt und die Maschine in den ewigen Umlauf versetzt wird.



*Abb. 9: Münzeinwurf von innen, Münzführung ist demontiert*



*Abb. 10: Münzeinwurf von der Seite*

Und hier oben befindet sich noch der Hebel mit dem Statikverbinder, welcher die Münze unter dem Taster festhält, bis der Hebel gezogen wird:



Abb. 11: Hebel

Wenn beide Taster auf Gewinnstellung und der Start-Stopp-Taster in Ruhestellung sind, wird über das Flip-Flop das Relais betätigt, welches als Polwender funktioniert. Der Motor mit dem Hubgetriebe schiebt den Münzbehälter zur Vorderseite, wo er mit einem End-Taster gestoppt wird.

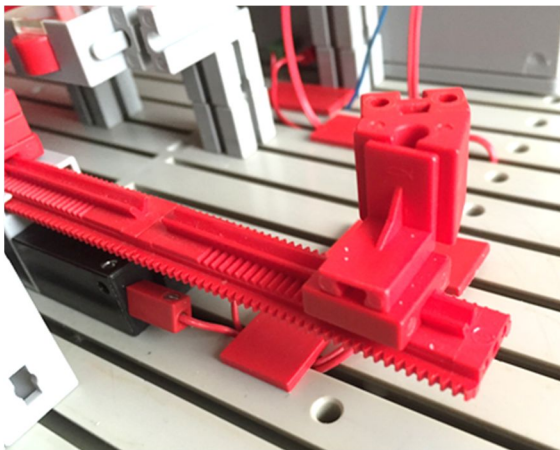


Abb. 12: Motor zum Verschieben des Geldbehälters

Sobald die nächste Münze eingeworfen wird, wird über den Arbeitskontakt des Start/Stopp-Tasters, welcher auch die Motoren steuert, das Flip-Flop zurückgesetzt, das Relais fällt ab und der Motor zieht den Geldbehälter wieder zurück.

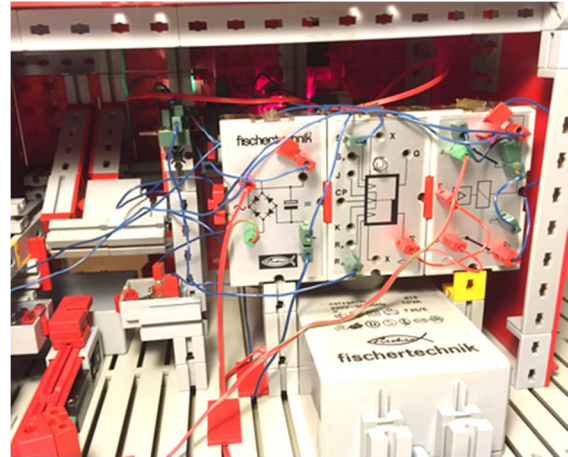


Abb. 13: Elektronik und Trafo

Wie man sieht, wurde das ganze Modell mit den alten Silberlingen und dem Uralt-Trafo konstruiert. Ich betreibe bewusst alles mit ca. 6 Volt ab Gleichrichter. Es ist durchaus erwünscht, dass die Motoren etwas langsamer drehen als mit der Nennspannung von 9 Volt. In Abb. 14 findet ihr den vollständigen Schaltplan.

Es ist wichtig, dass das Setzen des Flip-Flops bei Gewinnschaltung der beiden Taster erst über den Ruhekontakt des Start/Stopp-Tasters erfolgt (also wenn die Münze heruntergefallen ist), sonst ruckelt der Hub-Motor ständig ein bisschen vor und zurück, weil während der Drehung die Gewinnstellung immer wieder mal kurzzeitig erreicht wird.

### Erweiterungsmöglichkeiten

Reizvoll wären zusätzliche Lampen bzw. LEDs in verschiedenen Farben, welche mit zusätzlichen Kontakten angesteuert werden. Dann könnte man mit zusätzlicher Elektronik mehrere, abgestufte Gewinnmöglichkeiten programmieren.

Schließlich könnte man anstelle des mechanischen Hebels eine Zeitsteuerung einbauen, welche die Maschine nach Einwurf der Münze für eine fixe oder variable Zeit zum Laufen bringt.

## Quellen

- [1] fischertechnik: *Club-Modell 2-75: Bauanleitung Spielautomat* [in der ft-Datenbank](#). Fischer-Werke, Tumlingen, 1975.
- [2] Wikipedia: [Slot machine](#). Photo: Nazox, Lizenz CC BY-SA 3.0.

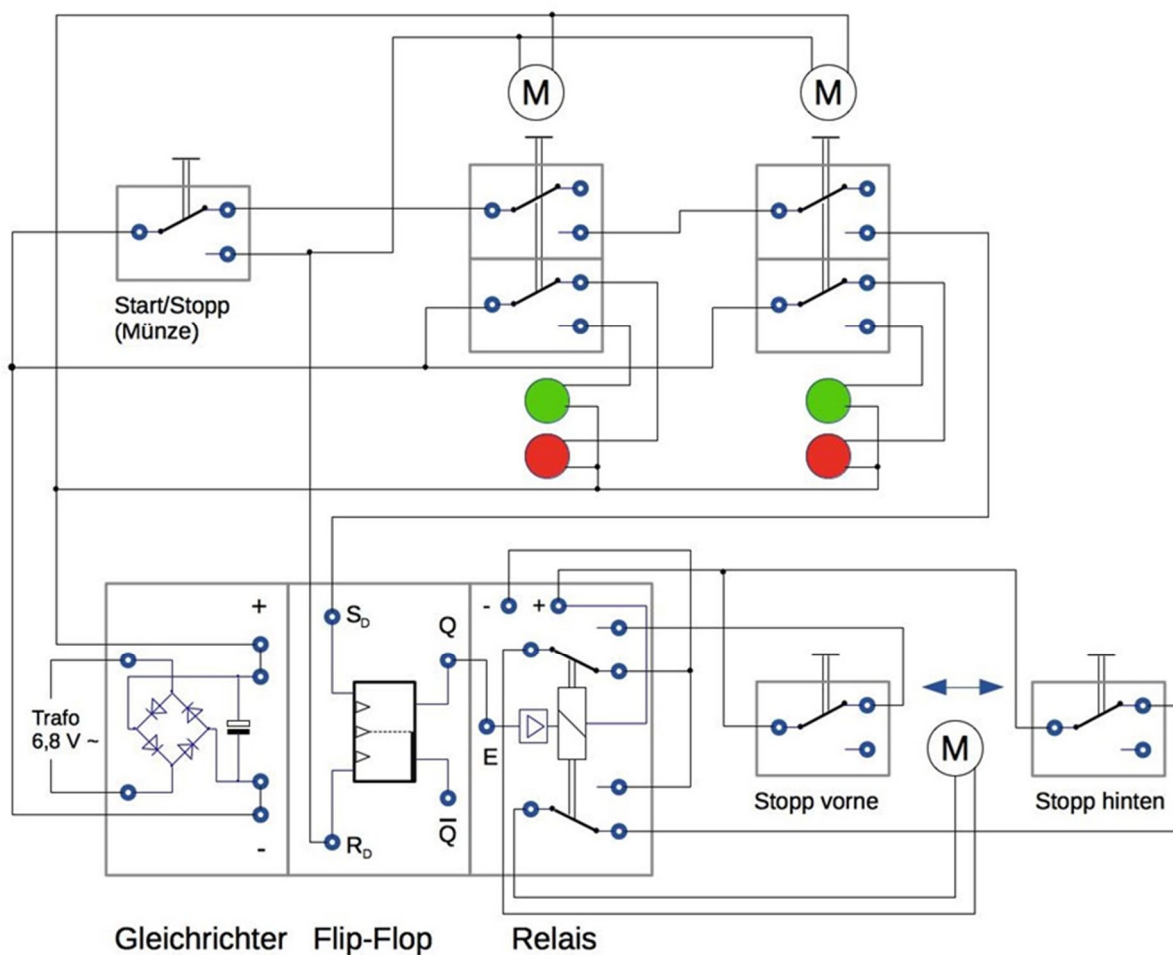


Abb. 14: Schaltplan des Spielautomats



Modell

## Tür zu, bitte!

Stefan Falk

*Wer kennt ihn nicht, diesen Ruf? Der ertönt, wenn jemand durch eine Tür geht, sie aber nicht wieder schließt – und es zieht! Für fischertechniker gibt es hier Abhilfe in Form einer Maschine, die eine Tür automatisch wieder zu schiebt.*

### Die Aufgabenstellung

Unsere Maschine soll folgendes leisten:

1. Sie soll erkennen, wenn die zu überwachende Tür geöffnet wird.
2. Wenn die Tür geöffnet wurde, soll sie einige Sekunden warten. Schließlich soll man ja in Ruhe durch die Tür treten.
3. Nach Ablauf der Wartezeit soll sie die Tür – eine richtige, echte, große und entsprechend schwere Wohnungstür – automatisch zu schieben. Wenn sie vielleicht auch nicht stark genug ist, die Tür wirklich ins Schloss zu drücken, so soll sie die Tür zumindest anlehnen.
4. Wenn die Tür angelehnt ist, soll sich die Maschine wieder stillsetzen.

All das soll natürlich nur mit normalen fischertechnik-Bauteilen gelingen. Vielleicht wollt ihr euch, bevor ihr weiterlest oder die Bilder anschaut, selber überlegen, wie ihr das Problem angehen würdet. Bestimmt gibt es dafür, wie meistens, eine lange Reihe verschiedener Möglichkeiten.

### Der Konstruktionsvorschlag

Die in diesem Beitrag vorgestellte Maschine funktioniert wie folgt: Das blaue Element in der Zeichnung (Abb. 1) steht für die zu konstruierende Maschine. Sie muss einen hinreichend kräftigen Antrieb bekommen, um die Tür tatsächlich schieben zu können. Damit sie nicht aus dem Türbereich

hinaus wandert, wenn jemand die Tür schwingvoll aufstößt bzw. wenn die Maschine sie wieder zu drückt, bekommt sie an beiden Enden Klammern, die etwas um die Tür fassen.

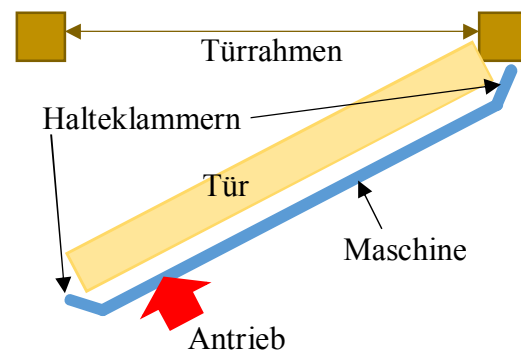


Abb. 1: Funktionsprinzip

Nun bleiben aber die folgenden Probleme zu lösen:

- Wie konstruieren wir einen Antrieb, der eine echte Tür schließen kann?
- Wie erkennt die Maschine, dass die Tür geöffnet wurde?
- Wie realisieren wir die Wartezeit?
- Woran stellt die Maschine fest, dass die Tür nicht weiter zugeschoben werden kann?
- Wie sorgen wir dafür, dass der Antrieb nicht in Stücke gerissen wird, wenn die Tür plötzlich von jemandem aufgestoßen wird? Vielleicht sogar, während die Maschine sie gerade schließen will?

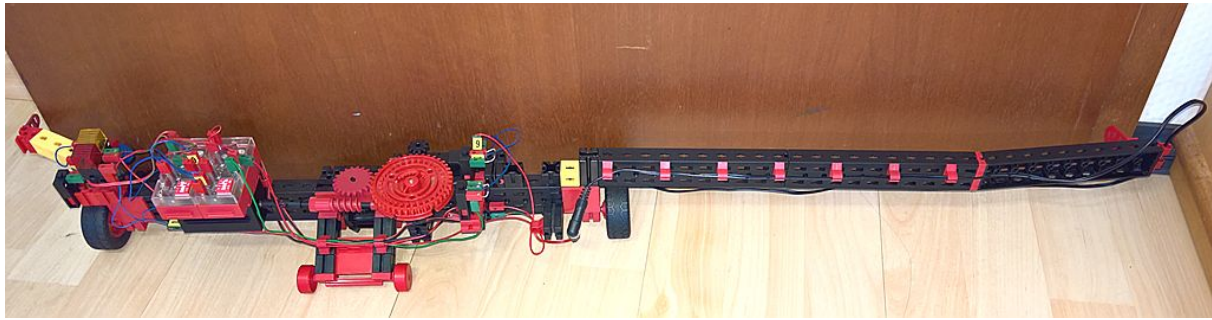


Abb. 2: Die Maschine, zum Einsatz bereit

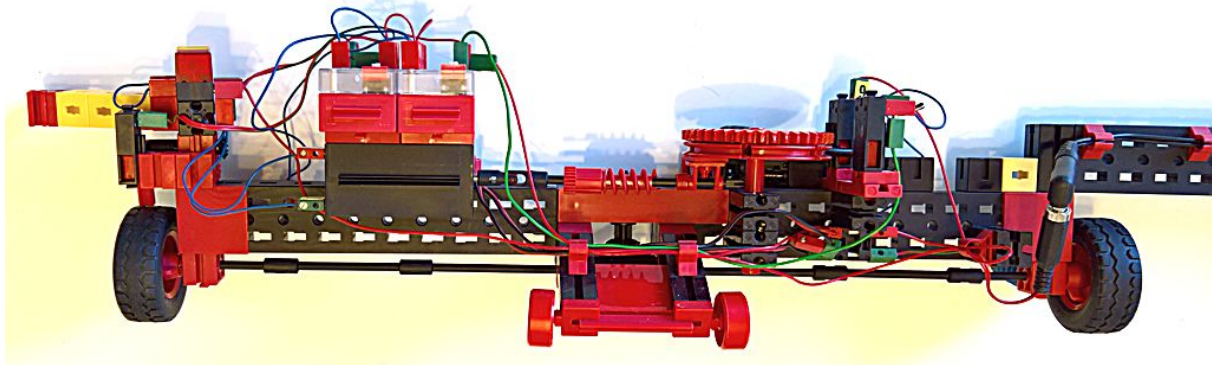


Abb. 3: Die Maschine aus dem Blickwinkel zur Tür

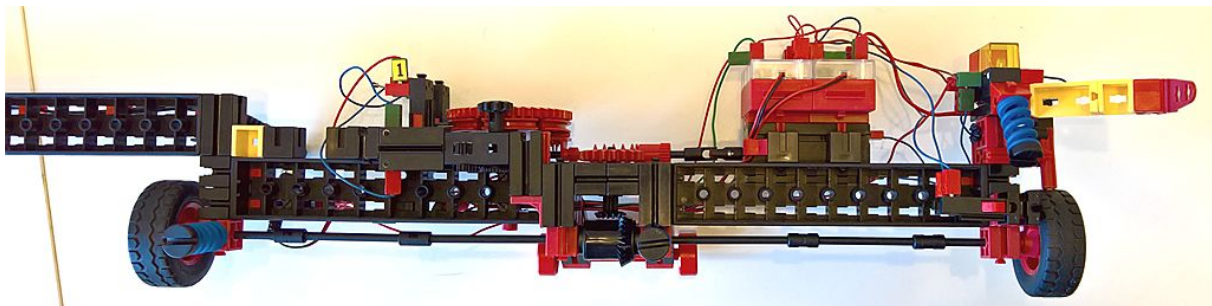


Abb. 4: Die Maschine aus dem Blickwinkel von der Tür

## Die Mechanik der Maschine

Da haben wir uns ja was vorgenommen. Aber es gilt ja „geht nicht – gibt’s nicht!“ Abb. 2 zeigt die Maschine an einer Tür, Abb. 3 und 4 Ansichten beim Blick auf die Tür bzw. von der Tür auf den interessanten Teil der Maschine. Die elektrische und elektronische Steuerung der Maschine schauen wir uns später noch genau an. Zunächst folgt die Beschreibung der Mechanik:

### Der Antrieb

Ein XM-Motor treibt über ein Differenzial und eine später noch zu beschreibende

Kupplung zwei Gummireifen an. Das Eigengewicht der Maschine genügt für hinreichende Reibung am recht glatten Laminat-Fußboden und der in Abb. 2 gezeigten Zimmertür. Auf Teppichboden schiebt die Maschine vermutlich noch etwas kräftiger.

### Die Kupplung

Ein Mensch öffnet eine Tür ja selten zaghaft, sondern mitunter heftig. Damit der Antrieb dabei nicht beschädigt wird, wird er nur für die Dauer des Schiebevorgangs eingekuppelt. Außerhalb dieser Zeit ist der Motor von den Rädern vollständig isoliert, und die Räder mögen so schnell gedreht

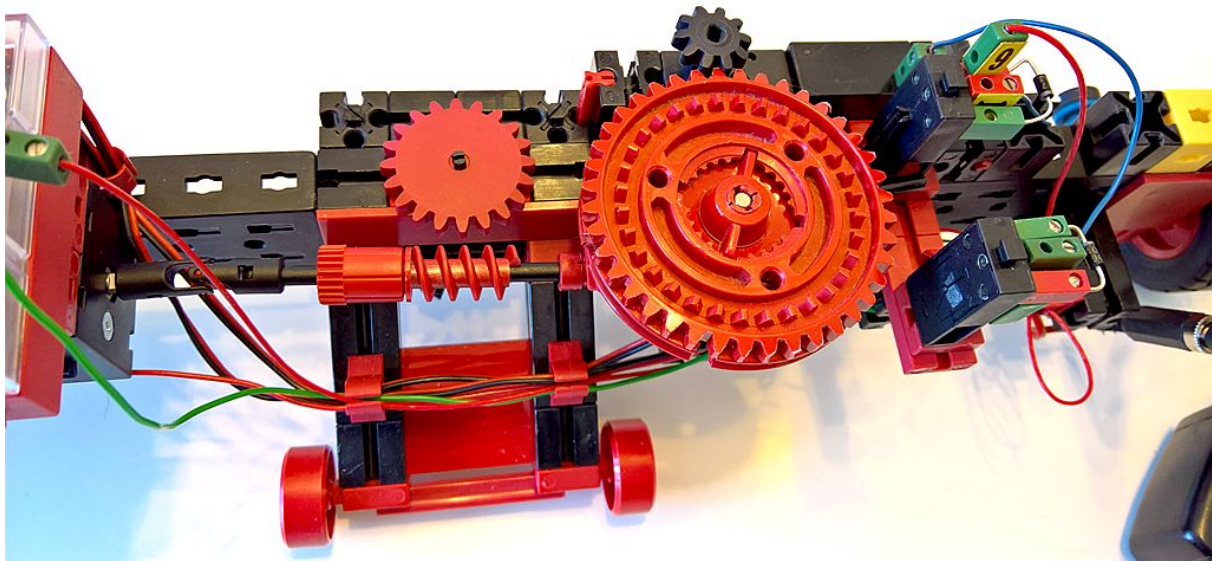


Abb. 5: Überblick über die Kupplungsmechanik

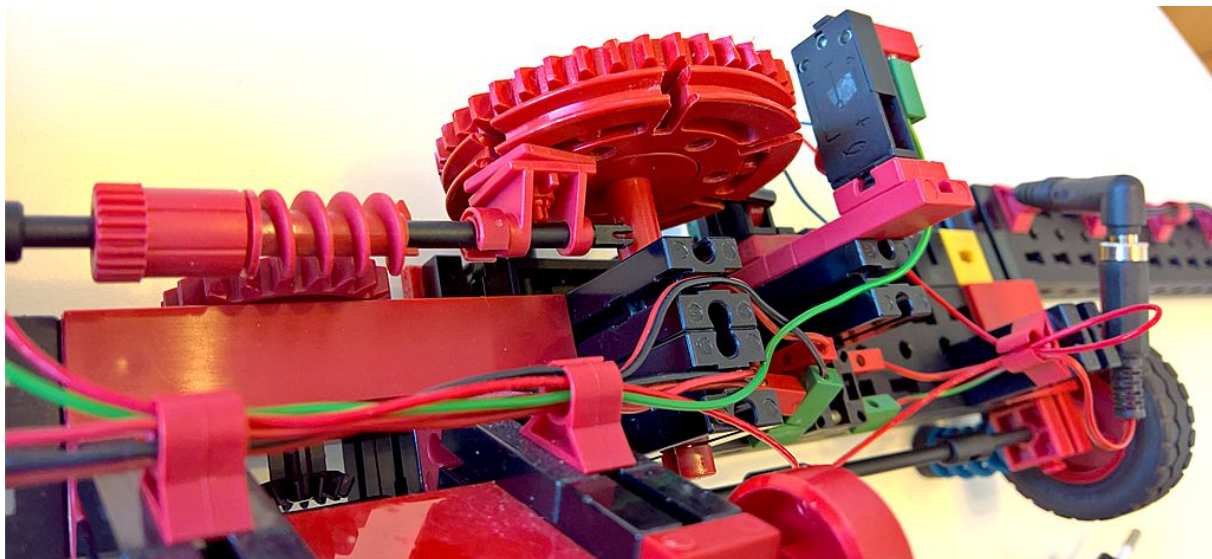


Abb. 6: Blick von unten auf die Kupplung

werden, wie der Mensch die Tür aufstößt – der Mechanik macht das nichts aus. Das Auskuppeln muss nur im Fall des Falles schnell genug erfolgen, sodass auch wirklich nichts beschädigt wird.

Das erledigt die „Kupplung“ (Abb. 5), die die Schnecke ins Rast-Z20 eingreifen lässt oder rasch aus ihr herauszieht. Beachtet das Kardangelenk zwischen XM-Motor und Schnecke. Ein XS-Motor dreht am Z40, mit dem eine Drehscheibe 60 verbunden ist. Die beiden stecken zusammen mit einer [31983 Hülse 15](#) als Abstandshalter auf einer

Metallachse, die in zwei Bausteinen 30 mit Bohrung gelagert ist. Auf der Unterseite der Drehscheibe (Abb. 6) steckt ein [31771 Lagerstück 1](#) mit [31772 Lagerstück 2](#), in dem die Schneckenachse geführt wird. Je nach Drehung des Z40 greift die Schnecke damit ins Rast-Z20 ein oder nicht. Das Z20 sitzt auf einem [31414 Differenzial-Kegelzahnrad Z14](#) und ist für den Antrieb des Differenzialgetriebes zuständig. Den Einbau des XS-Motors zeigt Abb. 7. Woran das Aufstoßen der Tür erkannt wird, besprechen wir gleich noch.

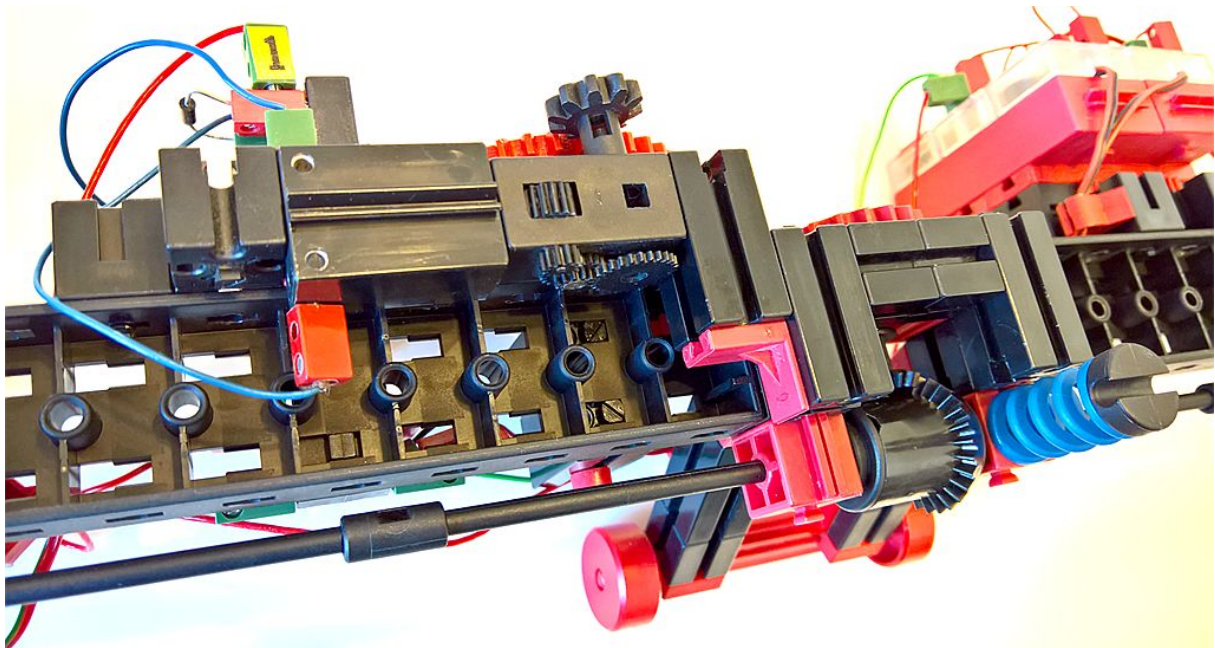


Abb. 7: Antriebsstrang und Kupplungsmotor

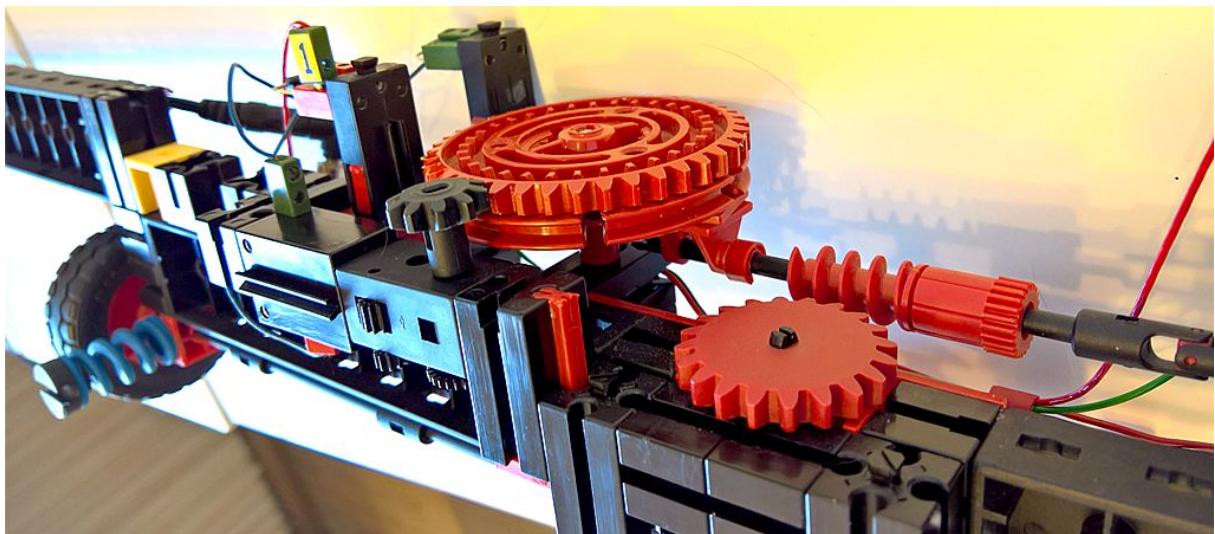


Abb. 8: Querblick auf die Kupplung

An den beiden Endlagen – ganz ein- bzw. ganz ausgekuppelt – wird von einer Achse 30, die in der Drehscheibe steckt, je ein Endlagentaster betätigt (Abb. 9). Die Taster bilden mit einer Diodenschaltung [1] die Endlagenabschaltung des XS-Motors.

Die Kupplung wirkt (insbesondere beim Auskuppeln!) so schnell, dass auch beim plötzlichen Aufstoßen der Tür während eines Zuschiebe-Vorgangs keine Schäden am Getriebe zu befürchten sind.

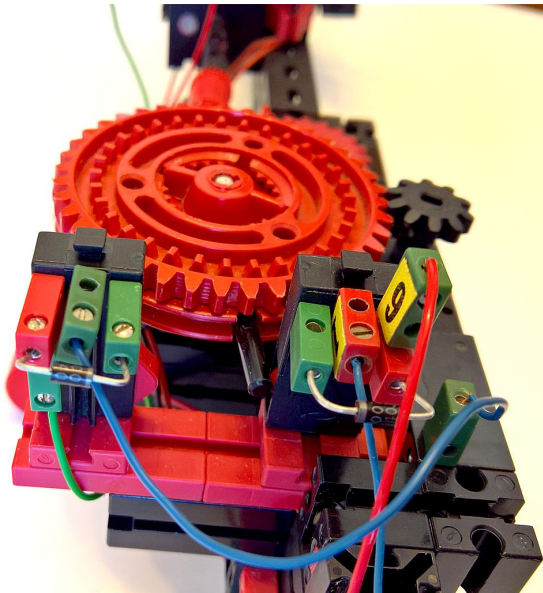
### **Der Antriebsstrang**

Das Differenzial für den Antrieb wird ganz konventionell angetrieben (Abb. 10 und 11). Via Rastachsen geht das Drehmoment auf die beiden außen sitzenden Reifen.

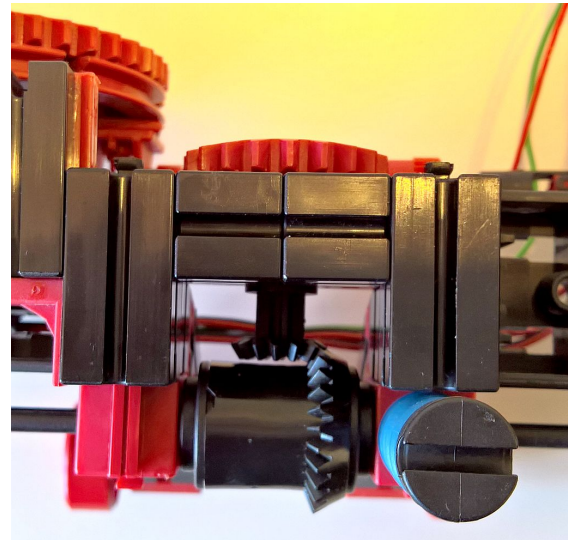
### **Das Öffnen der Tür feststellen**

Die Tür mag langsam oder schnell geöffnet werden: Auf jeden Fall werden dadurch die Gummireifen gezwungen, sich in Öffnungsrichtung der Tür zu drehen. In dieser und nur in dieser Drehrichtung hätten wir

gerne einen elektrischen Impuls für unsere Steuerelektronik. Das leistet die in Abb. 12 bis 14 gezeigte, sehr feinfühlig reagierende Mechanik.

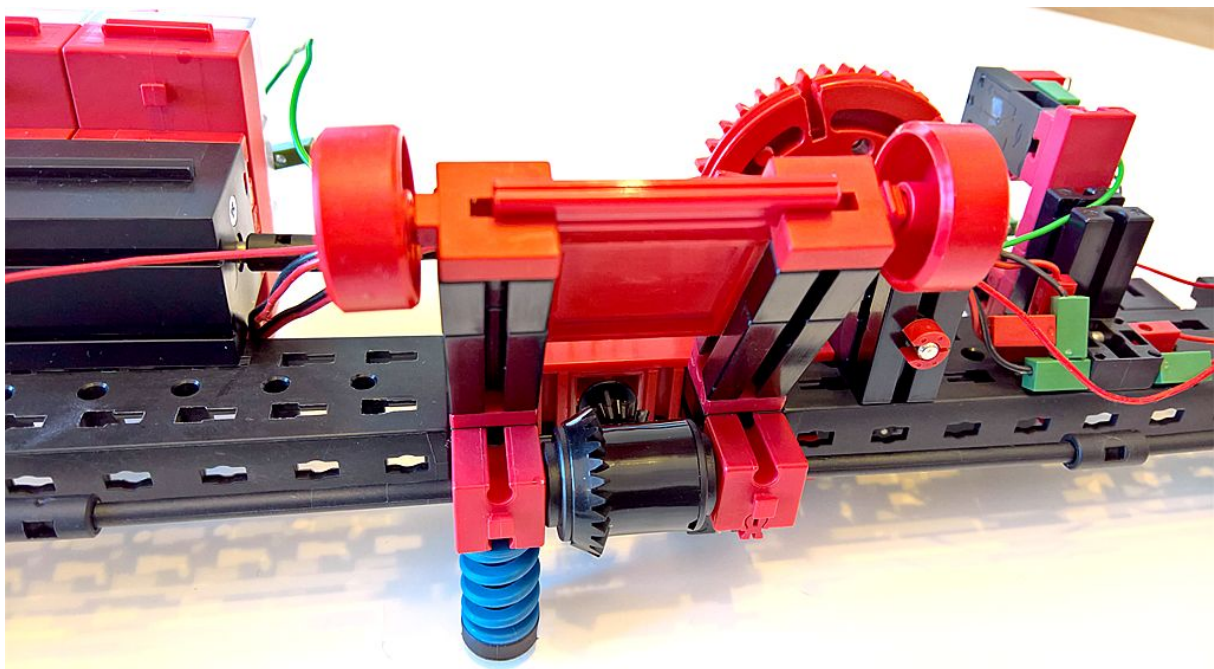


*Abb. 9: Platzierung und Einstellbarkeit der Kupplungs-Endlagentaster mit Dioden*



*Abb. 10: Sitz des Differenzials*

Der bewegliche Verbinder 30 über dem Reifen vor dem Taster ist so justiert, dass er gerade hinreichend auf den Reifen drückt, um von ihm mitgenommen zu werden, und er ist gerade hinreichend weit entfernt vom Reifen angebracht, um nach einem kleinen Auslenkungswinkel fast frei zu sein. Wenn sich der Reifen in Richtung des Öffnens der Tür dreht, wird der Verbinder auf den Taster gedrückt, der uns das gewünschte Signal liefert. In die andere Richtung, also



*Abb. 11: Blick von unten auf den Antriebsstrang*

beim Schließen der Tür, wird er einfach ins Freie gedreht – es passiert weiter nichts. Der zweite, waagrecht angebrachte Verbinder 30 begrenzt diesen Ausschlag in Richtung Tür.



Abb. 12: Feststellen des Öffnens der Tür

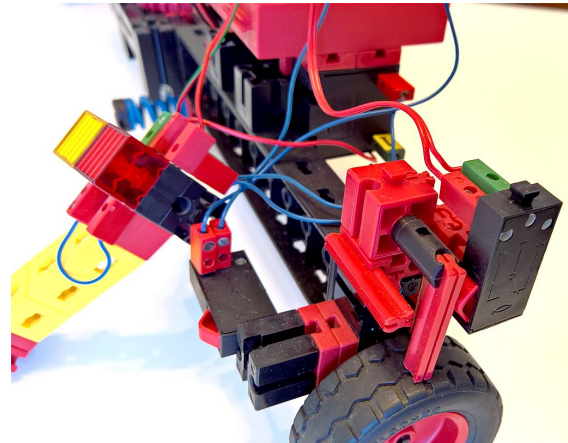


Abb. 13: Blick auf die Erkennung mit abgenommener Türklammer

### **Feststellen, dass die Tür zu ist**

Unser Gerät drückt die Tür nicht direkt, sondern über Federn. Wenn es die Tür zugeschoben hat, wird eine der Federn so stark eingedrückt, dass ein von dieser Feder „beschützter“ Taster betätigt wird. Das ist das Signal zum Ausschalten und Auskuppeln des Antriebs. Es handelt sich um den waagrecht liegende Taster in Abb. 13

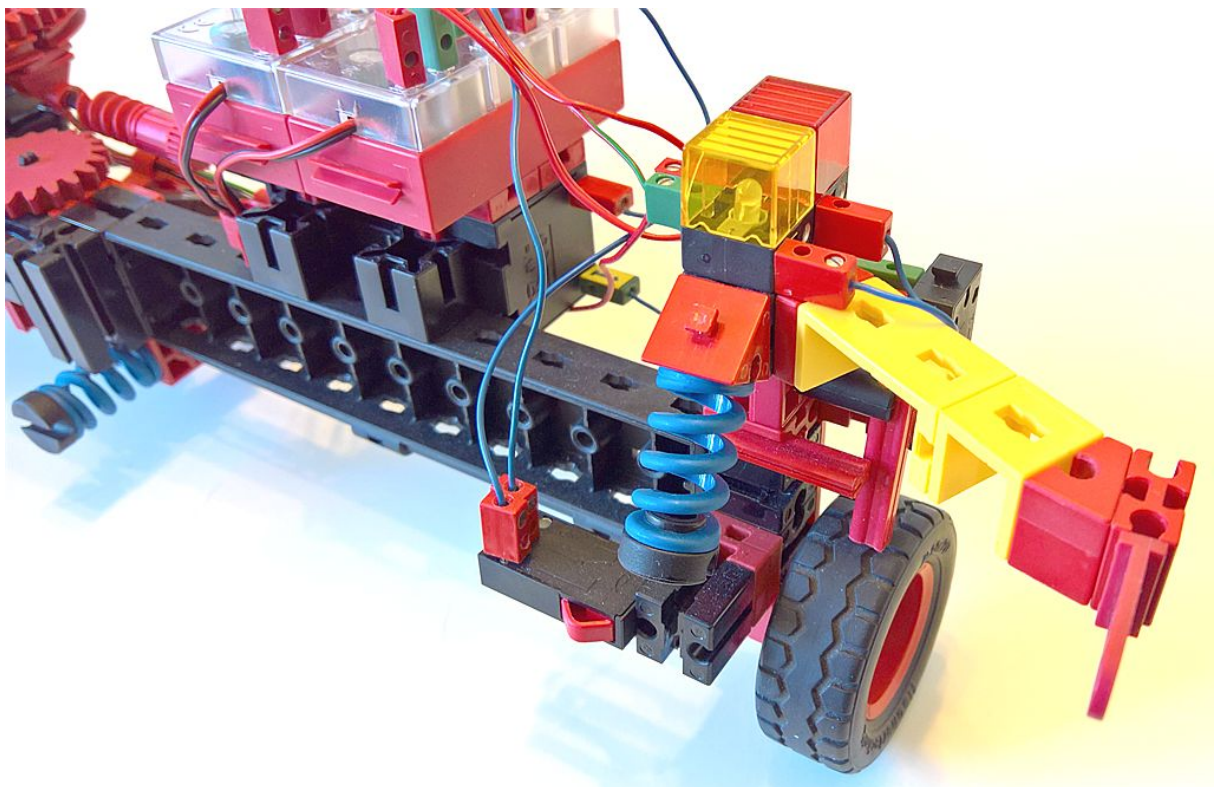


Abb. 14: Erkennung des Öffnens und der Schließung der Tür, Befestigung des XM-Motors

und 14. Die Abb. 15 bis 18 zeigen weitere Details der Konstruktion:

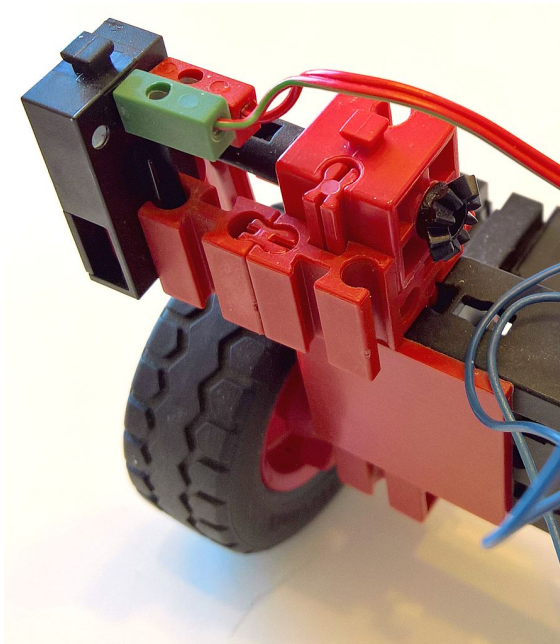


Abb. 15: Zur Befestigung des oberen Tasters

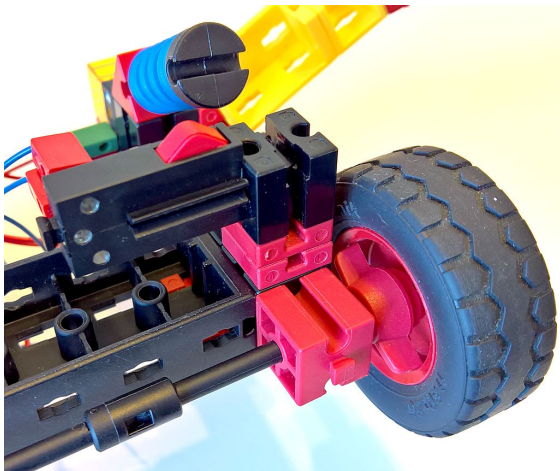


Abb. 16: Aufhängung des äußeren Rades

## Die elektronische Steuerung

Das war sie schon, die ganze Mechanik! Jetzt fehlt noch ein bisschen elektronischer Zauber, um den korrekten Ablauf zu garantieren:

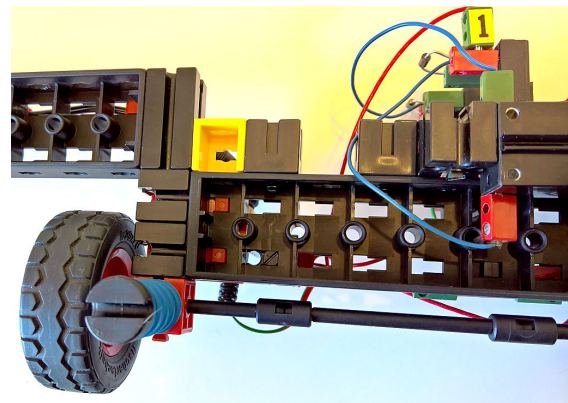


Abb. 17: Aufhängung des Rades beim Türinneren



Abb. 18: Die Tür-innere Halteklammer

## Zeitverzögerung

Wenn das Türöffnen vom zuständigen Taster festgestellt wurde, muss eine Zeitschaltung aktiviert („getriggert“) werden. Das erledigt man in der Elektronik mit einem *Mono-Flop* [2]. Das ist ein Baustein, der auf ein Signal am Eingang für eine bestimmte Zeit einen Ausgang ein- und nach Ablauf der Zeit automatisch ausschaltet. Dazu dient uns ein E-Tec-Modul, welches laut Zusatzanleitung [3] (wie darin im Abschnitt 1.4 auf Seite 3 beschrieben) eben als ein solches Mono-Flop eingestellt wird. Die vier DIP-Schalter müssen auf aus-aus-ein-ein eingestellt werden. Auch die neueren Electronics- und Minibots-Moduln bieten (auf anderen DIP-Stellungen) ein Mono-Flop.<sup>1</sup>

Die Verschaltung laut Schaltbild (Abb. 19) bewirkt eine Zeitverzögerung von etwa zehn Sekunden. Das sollte genügen, um bequem durch die Tür zu kommen. Während dieser Wartezeit lassen wir eine gelbe Warnlampe aufleuchten, die am Mono-Flop

<sup>1</sup> Wer einen PC, ein Notebook, Tablet oder Phone mit Windows ab Version 8.1 als Betriebssystem benutzt, findet unter [4] eine kostenlose kleine

App, die für die DIP-Switch-Einstellung der E-Tec-, Electronics- und MiniBots-Moduln eine handliche Referenz zum Nachschlagen bietet.

angeschlossen ist und ankündigt, dass die Tür demnächst wieder geschlossen wird.

### Aktivieren des Antriebs

Sobald dieses Zeitglied wieder abfällt und die eingestellte Zeit also vorüber ist, wird über den Ausgang  $\bar{Q}$  des Monoflops ein zweites E-Tec-Modul angesteuert. Das ist (wie ebenfalls in [3] im Abschnitt 1.3 ab Seite 2 beschrieben) als Flip-Flop eingestellt, indem die DIP-Switches auf ein-aus-ein-ein geschaltet werden. Das Flip-Flop wird also nach Ablauf der Zeit gesetzt. An ihm sind drei Verbraucher angeschlossen:

1. Der Antriebs-XM-Motor.
2. Eine parallel dazu geschaltete rote Signallampe, die während des Schließens der Tür also aufleuchtet.
3. Der Kupplungs-XS-Motor, allerdings über die Endlagenschaltung mit den beiden Tastern und Dioden (nach [1]).

Sobald der XM-Motor anläuft wird die von ihm angetriebene Schnecke vom XS-Motor ins Z40 eingekuppelt, und sobald der XM-Motor ausgeschaltet wird, wird er auch sofort vom Z40 entkoppelt. Damit das auch dann passiert, wenn die Tür wieder aufgestoßen wird, während unsere Maschine sie gerade zu schiebt, steuert der Taster zur

Öffnungs-Erkennung nicht etwa das Mono-Flop-Zeitglied an, sondern setzt das Flip-Flop sofort zurück. Auf diese Weise wird das Getriebe augenblicklich geschützt, sobald die Tür aufgeschoben wird – egal ob die Maschine gerade im Leerlauf still wartet oder ob sie eben am Schieben war.

### Der Funktionsablauf

Wenn die Maschine richtig funktioniert, ergibt sich damit folgender Ablauf:

1. Die Maschine steht still und wartet darauf, dass jemand die Tür öffnet.
2. Sobald das geschieht, wird der Taster „Tür wird geöffnet“ über den Reifen und Verbinder 30 betätigt. Dadurch startet das Mono-Flop die Zehn-Sekunden-Wartezeit und das gelbe Warnlämpchen leuchtet auf.
3. Nach Ablauf der Zeit geht das gelbe Lämpchen aus, das rote leuchtet auf, der XM-Motor zum Antrieb arbeitet, und er wird vom XS-Motor mit seiner Schnecke ins Z40 eingekuppelt. Die Tür wird zu geschoben.
4. Sobald es nicht mehr weitergeht (weil die Tür zu ist, ein Hindernis im Weg oder eine Person in der Tür steht), wird der Taster „Tür ist geschlossen“ von der Tür

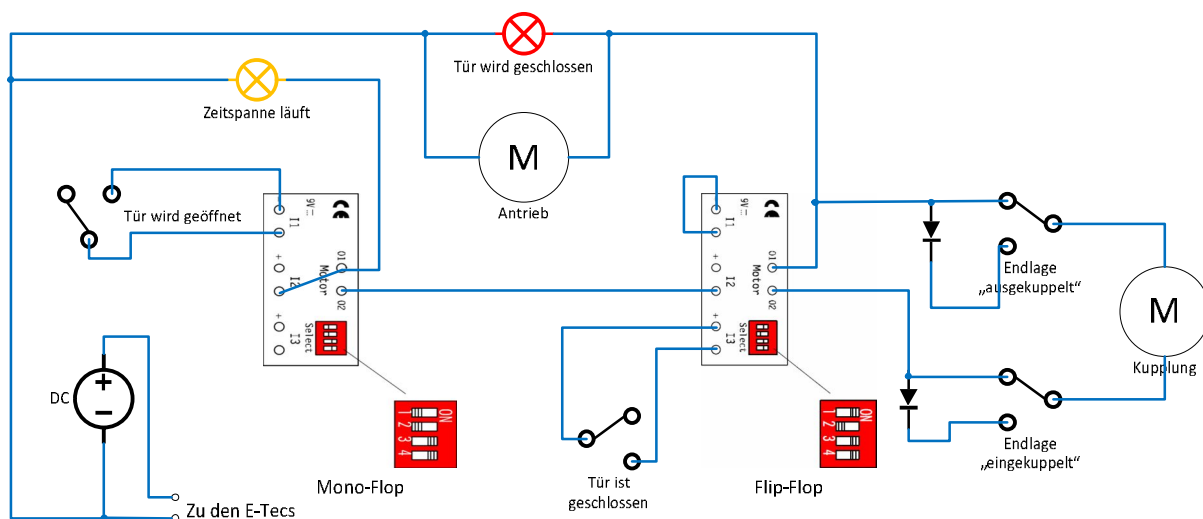


Abb. 19: Schaltbild der „Tür zu!“-Maschine



selbst gedrückt und setzt das Flip-Flop still. Damit gehen die rote Lampe und der XM-Motor aus, und der XS-Motor kuppelt das Getriebe schnell aus. Beim nächsten Öffnen der Tür ist es also im „Leerlauf“ und wird nicht beschädigt, weil sich die Reifen, das Differenzial und das Rast-Z20 frei drehen können.

5. Wenn der XM-Motor ausgeschaltet und ausgekuppelt ist, kommt es vor, dass die Tür etwas zurückschwingt. Eventuell genügt das, um die Maschine erneut ab Schritt 2 in Gang zu setzen. Unser Gerät schiebt also gleich nochmal ein bisschen nach. Das macht aber nichts, denn nach ein oder zwei Mal wird das Zurückschwingen so gering sein, dass es Taster 1 nicht erneut auslöst.
6. Damit ist die Arbeit der Maschine erstmal beendet. Sie geht wieder in den Wartezustand von Schritt 1 über.

## Zur Feinjustage der Maschine

Einige Dinge müssen an der Maschine justiert werden, damit sie im rauen Alltagsbetrieb wie gewünscht funktioniert:

### **Der Taster für die Öffnungs-erkennung**

Dieser Taster kann nach oben verschoben werden, wenn die Erkennung noch zu empfindlich ist. Das merkt man daran, dass selbst ein geringes Zurückschwingen der Tür die Maschine wieder in Gang setzt. Ebenso kann der Verbinder 30 durch Verschieben mehr oder weniger tief in den Reifen drücken, um die Erkennung gerade zuverlässig zu machen.

### **Die Endlagentaster der Kupplung**

Diese beiden Taster sind leicht zu justieren: Je weiter außen sie sitzen, desto später wird der XS-Motor abschalten. Stellt die Taster so ein, dass die Schnecke auf der einen Seite zuverlässig und satt ins Z40 eingreift und in der anderen Endlage sicher außerhalb des Z40 positioniert wird.

### **Der Taster für die Erkennung der geschlossenen Tür**

Dieser Taster kann in der Nut des Bausteins, der ihn trägt, fein weiter nach vorne oder hinten positioniert werden. Ihr müsst ihn so weit nach hinten (weg von der Tür) schieben, dass er während des Zuschiebens der Tür nicht schon vorzeitig auslöst, aber auch so weit nach vorne (zur Tür hin), dass er in der Endlage der Tür zuverlässig betätigt wird und die Maschine abschaltet.

## Quellen

- [1] Falk, Stefan: *Motorsteuerungen (Teil 3)*. ft:pedia 3/2011, S. 7-11.
- [2] Wikipedia: [Monostabile Kippstufe](#).
- [3] fischertechnik: [E-Tec Spezialprogramme für Digitaltechnik](#).
- [4] Falk, Stefan: [ft electronics](#). App für Windows 8.1/10, Windows Phone 8.1, Windows 10 Mobile. SF Software-Beratung GmbH, Ettlingen, 2016.
- [5] Falk, Stefan: *Tür zu, bitte!* [YouTube-Video](#), 2016.

Modell

## Gottwald MK500

Peter Krijnen

*Mitten in den 1970er Jahren gab es in Boskoop (NL) einen Unfall an einer Hebebrücke. Während des Anhebens brach in einem der beiden Türme eine Achse, was zur Folge hatte, dass die Brücke durch ihre Schiefelage nicht mehr zu bewegen war. Für eine Gemeinde, die von dieser Brücke abhing, ergab das große Probleme: Schifffahrt war nicht mehr möglich; LKWs und PKWs mussten Umwege fahren. Nur für Fußgänger und Radler gab es eine kleine Fähre. Zur Reparatur haben die Behörden der Provinz Zuid-Holland die Firma Biglift um Hilfe gebeten. Biglift hat dazu ihren damals schwersten Kran eingesetzt: den Gottwald MK500-88.*

Als ich das erfuhr, bin ich jeden Tag auf meinem Fahrrad nach Boskoop gefahren, um den Aufbau zu beobachten. Leider hatte ich damals keine Kamera und durfte auch die meines Vaters nicht mitnehmen.

Damals versuchte ich natürlich, diesen Kran mit fischertechnik nachzubauen. Ich erinnere mich noch, dass ich die 32 Stück 45er Räder in ein 90 mm breites Fahrgestell eingebaut hatte. Ob ich auch die Abstützarme angebaut hatte, kann ich nicht mehr sagen. Ich weiß nur, dass ich die 180 x 90-Grundplatten als Basis für Fahrgestell und Oberbau benutzt habe – ohne Drehkranz.

Während der letzten 25 Jahre war ich mehrfach Teilnehmer der „Modelshow Europe“. Während einer dieser Shows kaufte ich das Buch „Gottwald: 100 Jahre Bagger – Krane – Rammen...“ [1]. Auf Seite 254 findet sich eine kleine Zeichnung des MK500-88, auf Seite 259 eine des MK600-89. Da von letzterem mehr Details beschrieben wurden, verwendete ich den MK600 als Basis für meine Modelle.

In den 1970er Jahren gab es fischertechnik nur in Grau und Rot, daher wollte ich das erste Modell auch in Grau/Rot bauen. Allerdings wollte die Motorisierung nicht recht gelingen. Wegen des Maßstabs von 1:27 (an dem der 45er Reifen schuld ist) gab es keinen Platz, um drei graue M-Motoren

einzubauen. Auch der Versuch mit drei Mini-Motoren schlug fehl. Im selben Zeitraum fand ich auf der Website von Conrad endlich die 12-Volt-Versionen der „Micro Metal Gear Motoren“, die dort mit fünf verschiedenen Übersetzungen angeboten werden (Abb. 1). Damals bestellte ich von jeder Variante einen.



Abb. 1: Conrad „Micro Metal Gear“-Motor und ein aufgebohrter Baustein 15 mit Bohrung und zwei Federnocken

Später kaufte ich neun Stück 1:50 für die Seilwinden und vier Stück 1:1000 zum Drehen der Kräne. Für die Abstützarme meines Demag CC4800 habe ich auch noch vier Stück 1:100 bestellt und eingebaut. Außerdem stehen die Varianten 1:150 und 1:298 zur Verfügung. Die Motoren haben einen Durchmesser von 12 mm; die Achse ist 3 mm dick.

Im Demag-Modell schraubte ich die Motoren auf je einen Flachstein 30 x 30. Das war beim Gottwald aber nicht möglich. Als nächstes versuchte ich, die Motoren mit Hilfe der Schneckenmutter m1,5 ([37925](#)), einzubauen (Abb. 1). Mit der Fräsmaschine bohrte ich das Loch auf 12 mm auf. Da aber die Abschlussplatte an der Anschlussseite der Motoren um einige Zehntel Millimeter größer ist, wird der Motor nie wirklich fest gehalten. Mit einem aufgebohrten Baustein



Abb. 2: Schwerlastkran

15 mit Bohrung (32064) ergab sich eine bessere Lösung (Abb. 3). Mit der Fräsmaschine wurde wieder das Loch aufgebohrt, aber jetzt so weit, dass der Motor hineinpasste. Die Getriebe haben ein Querschnitt von  $12 \cdot 10$  mm, sodass sie gegen Verdrehen geschützt sind. Zwei Federnocken 31982 befestigen die Motoren zusätzlich.

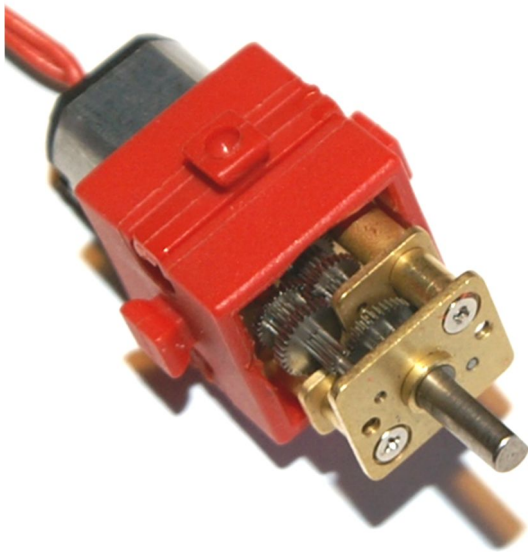


Abb. 3: Der Motor, eingebaut in einen aufgebohrten Baustein 15 mit Bohrung

Zum Antrieb der Seilwinden brauchte ich nur das Loch des Rastritzels mit der Drehmaschine auf 2,9 mm aufzubohren. Für die 1:1000-Motoren bohrte ich aber erst eine Achse auf 2,9 mm auf und presste sie auf die Motorachse. Das Rastritzel wurde auf 3,9 mm aufgebohrt.

Das 550 mm lange Fahrgestell baute ich zwischen den Anbolzungen der Abstützarme wie ein umgedrehtes 60 mm breites U auf. Es wurde mit vier doppelt bereiften Achsen bestückt. Am einen Ende ist eine einzelne Achse angebaut, am anderen sind es drei. Insgesamt benötigte das Fahrwerk damit 32 Reifen. Zum Schleppen befindet sich am langen Ende eine Deichsel. Am anderen Ende wurde ein Warnstreifen aufgeklebt. Wegen des Maßstabs von nur 1:27 ist das Fahrgestell ganz einfach gehalten: keine Steuerung, keine Motorisierung. Die

Breite des MK500-88 beträgt 115 mm, die des MK600-89 130 mm.

Der Kranoberwagen ist 105 mm breit und besteht aus zwei Längsträgern, die mehrfach miteinander verbunden sind. An drei dieser Verbindungen sind die Motoren für die Seilwinden montiert (Abb. 4), davor liegen die Seilrollen. Natürlich ist zwischen den beiden Trägern auch der Drehkranz eingebaut. Das Maschinenhaus sitzt weit vorne und trägt an seiner Frontseite die Kabine (Abb. 2). Die Länge vom Drehkranz bis zum Gegengewicht beträgt 330 mm. Der Auslegerfuß ist 95 mm vom Drehkranz entfernt gelagert. Um ein Durchbiegen zu verhindern, steckt in jedem Träger eine Metallachse.

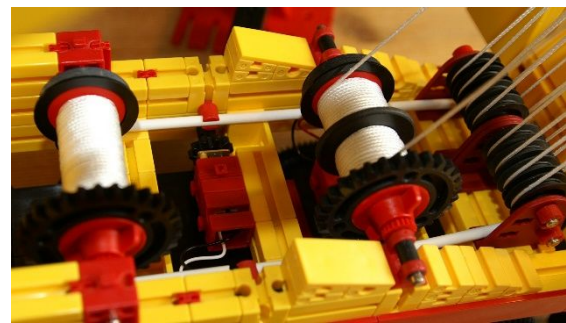


Abb. 4: Seilwinden

Die drei Seilwinden bestehen aus je einer Achse 110, einem Zahnrad Z30, einem Spurkranz 30 und fünf Klemmbuchsen 10. Für die doppelte Seilwinde fertigte ich mit der Drehmaschine eine Scheibe mit 34 mm an. Statt der fünf Klemmbuchsen 10 kamen dann vier 10er und zwei 5er zum Einsatz.

Das weiße Seil stammt aus dem Baumarkt und ist 1 mm dick. Auf der doppelt breiten Seilwinde sind davon 16 m aufgewickelt, die beiden anderen fassen je 9 m.

Da der Bau des 630 mm langen Auslegers doch ein wenig problematischer war als zunächst angenommen, habe ich in AutoCAD eine Zeichnung davon erstellt. Das war sehr hilfreich, gibt es in AutoCAD doch mehrere Messfunktionen, z. B. für die erforderliche Länge der Streben. Das ergab aber Längen, die mit einer Kombination



Abb. 5: Kran-Parade

von Streben nur ungefähr realisierbar sind. Deswegen musste ich die Anzahl Verstrebungen verringern. Es gelang mir aber, die Form des Auslegers beizubehalten. Später baute ich noch Zwischenstücke zu 12 m und 6 m ein. Der Querschnitt des Auslegers beläuft sich auf 105 x 90 mm.

Erst nachdem die grau/rote Version fertig war, fragte ich mich, wie der Kran wohl in den Originalfarben der damaligen Betreiber aussehen mag. In Abb. 2 und 5 ist das Ergebnis zu sehen: zwei MK500-88 und zwei MK600-89. Später baute ich auch noch einen geschleppten MK600 und drei MK650 (siehe Bilderpool [2]).

Bei einem solchen Projekt bemerkt man natürlich gleich, dass man sein Modell nicht ganz in gelb, rot oder schwarz aufbauen

kann: Bestimmte Teile gibt es eben nicht in allen Farben. Das Sarens-Modell habe ich deswegen einfach in originalen fischertechnik-Farben gehalten: ohne blaue Teile. Auch das Fehlen roter Streben ist ärgerlich.

Obwohl ich ein einfaches Modell bauen wollte, sollte es doch über eine Fernsteuerung verfügen. Ich besitze zwei RC-Geräte; mein Zweikanal-Accoms-Gerät hat zwei Kanäle zu wenig, und mein auf 22 Kanäle ausgebautes Robbe/Futaba F14 ist fest im Demag CC4800 eingebaut. Da blieb also nur die fischertechnik-IR-Fernsteuerung. Der Empfänger des IR-Control-Sets verfügt über drei Motoranschlüsse und einen Servoanschluss – ich brauchte aber vier Motoranschlüsse. Bei der Bestellung der Motoren

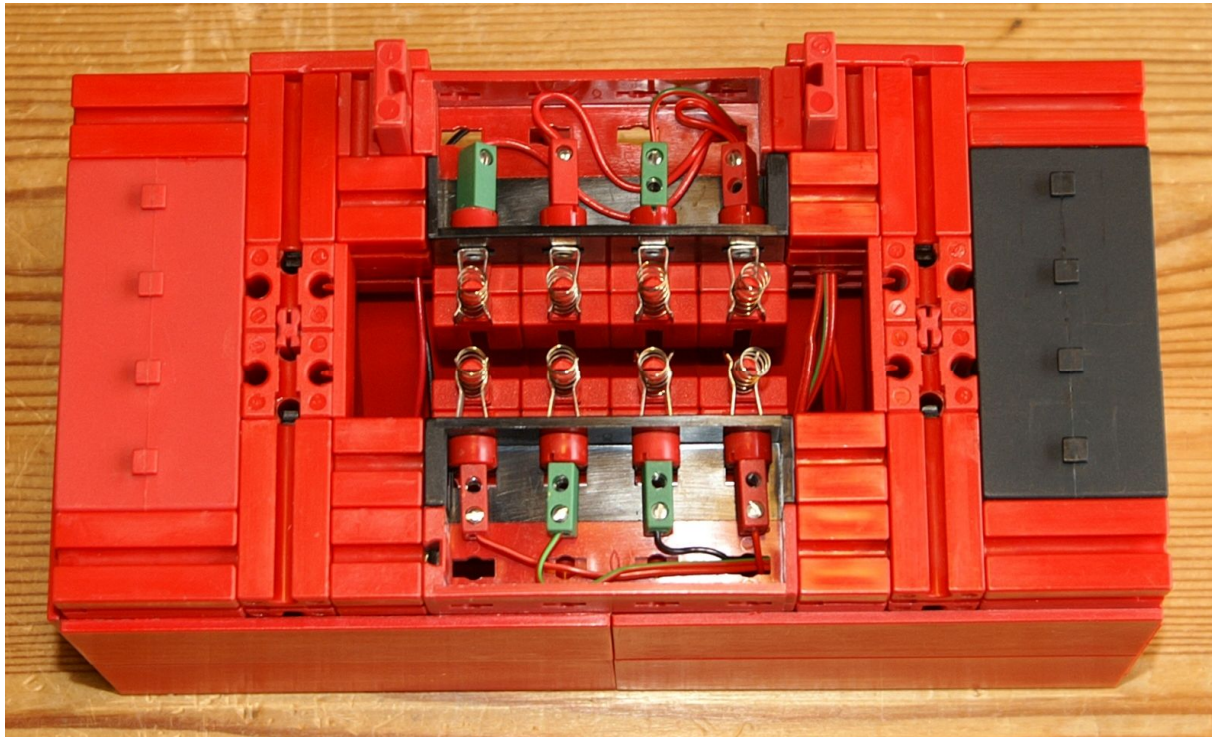


Abb. 6: Anschlüsse

kaufte ich auch gleich vier Micro-Fahrtregler. Obwohl die kleiner sind als eine Bauplatte 15 x 15 halten sie doch Ströme bis zu 1,8 A aus. Anstelle eines Servos schloss ich einfach einen Fahrtregler am Servoausgang an. Mit ein wenig Knetmasse auf einer V-Bauplatte 15 x 15 war der Regler einfach zu befestigen (Abb. 7).

Empfänger, Regler und eine 9 V-Batterie konnte ich in eins der Gegengewichte einbauen (Abb. 6). Auf jedem Kranmodell sitzt eine Platine mit acht Kontaktflächen, an denen die acht Leitungen für die Motoren angelötet sind. Mit Hilfe von vier halbierten Federn unterhalb des Gegengewichts wird der Kontakt mit der Steuerung auf einfache Weise hergestellt.

## Quellen

- [1] Weinbach, Wolfgang: *Gottwald: 100 Jahre Bagger – Krane – Rammen...* Podszun Motorbücher GmbH, 2006, ISBN 3-86133-421-6.
- [2] Gottwald-Kranmodelle im [ft Community Bilderpool](#).

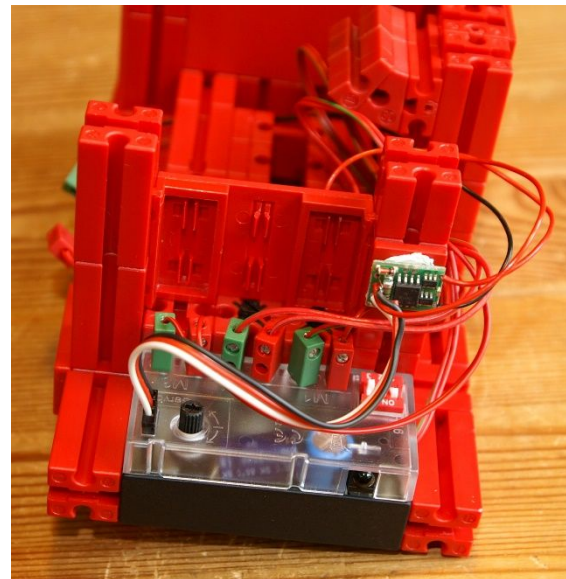


Abb. 7: Befestigung des Fahrtreglers und Anschluss an den IR-Empfänger

Modell

## Anwendungen für Magneten (3): Magnetkupplungen und Magnetrührer

Andreas Gail

Im vorliegenden Beitrag geht es im Grunde um die Anwendung von Magnetkupplungen. Hierbei wird eine Antriebskraft mithilfe der Haltekraft von Magneten übertragen. In den meisten Fällen erfolgt die Übertragung der Drehbewegung durch eine oder mehrere Wandungen hindurch. Als spezielles Beispiel soll hier ein gängiges Gerät eines Chemielabors gezeigt werden.

### Der typische Aufbau eines Magnetrührers

In Abb. 1 ist ein voll funktionsfähiges fischertechnik-Modell eines Magnetrührers gezeigt. Es besteht äußerlich nur aus einem Gehäuse mit zwei Bedienelementen und einer Geschwindigkeitsanzeige.

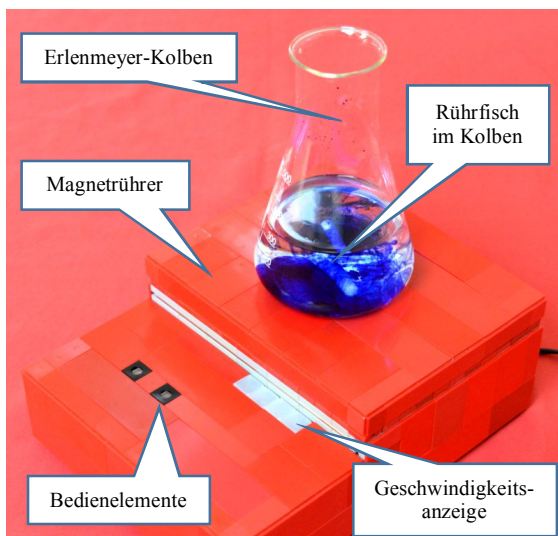


Abb. 1: fischertechnik-Modell eines Magnetrührers

Auf den Magnetrührer wurde ein Erlenmeyer-Kolben [1] gestellt, in dem sich eine inhomogene blau angefärbte Lösung befindet. Weiterhin wurde in den Kolben ein sog. Rührfisch [1] gegeben. Ein solcher Rührfisch ist in Abb. 2 nochmal gesondert gezeigt. Es handelt sich dabei um einen

vollummantelten Stabmagneten. Derartige Rührfische werden in chemischen Labors häufig verwendet.



Abb. 2: Rührfisch: vollummantelter Stabmagnet

### Anforderungen an einen Magnetrührer

Bevor mit dem Aufbau begonnen wird, sollte vorher stets klar sein, welche Eigenschaften bzw. Funktionalitäten ein Modell haben soll. Für einen funktionsfähigen Magnetrührer müssen mindestens folgende Anforderungen erfüllt sein.

- Eine gute Durchmischung im Kolben muss sichergestellt sein, deshalb ist eine ausreichende Rührgeschwindigkeit erforderlich.

- b) Zumindest eine grobe Anzeige der Rührgeschwindigkeit ist erforderlich, um diese bei den chemischen Versuchen protokollieren zu können.
- c) Die Rührdrehzahl soll langsam erhöht oder vermindert werden können. Bei zu starker Beschleunigung muss ansonsten damit gerechnet werden, dass der Rührfisch im Kolben anfängt zu springen, statt einer Drehbewegung zu folgen.
- d) Bei Arbeiten im Labor ist immer damit zu rechnen, dass Teile der Lösung verspritzen oder auch ein Kolben kaputtgeht. Deshalb sollte eine möglichst vollständige Abdichtung des Magnetrührer-Innenraums vorgesehen werden.

### Integration von Neodym-Magneten

Richtige Magneten gehören derzeit offenbar nicht zum Standard-Programm von fischertechnik. Das ist aber kein Problem, denn über das Internet lassen sich schnell die passenden Magneten [3] beschaffen. Diese müssen dann nur noch passend verklebt und richtig beschriftet werden.



Abb. 3: Quadermagneten 15x15x3 mm, Neodym NdFeB, vernickelt, verklebt und beschriftet

Damit die Verklebung möglichst genau erfolgen kann, ist eine eigene Vorrichtung hilfreich. Zum Schutz der Bauteile vor Klebstoffresten wurden diese mit Tesafilm beklebt. Abb. 4 und 5 zeigen den Aufbau:

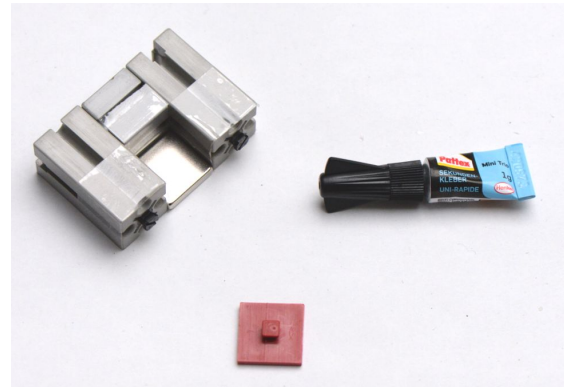


Abb. 4: Vorrichtung zur Verklebung, vor der Durchführung

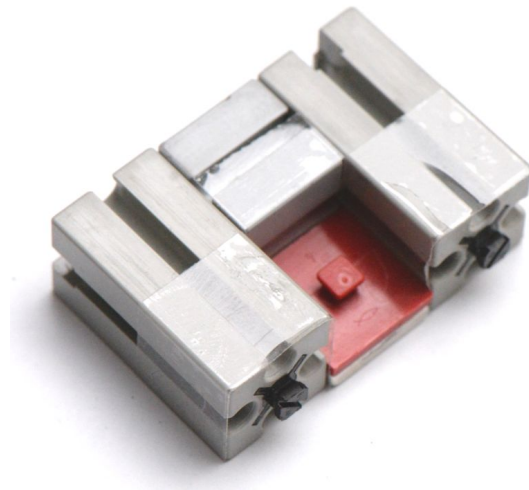


Abb. 5: Vorrichtung zur Verklebung, nach der Durchführung

Nach der Verklebung kann das neue Bauteil aus der Vorrichtung entnommen werden. Anschließend ist noch die Beschriftung vorzunehmen. Hierzu werden ein Kompass und eine weitere Vorrichtung benötigt, wie sie in Abb. 6 und 7 gezeigt sind.

Mit dem Aufbau gemäß Abb. 6 wird herausgefunden, dass im vorliegenden Beispiel der Kompass zum Magneten gerichtet die Nordausrichtung anzeigt. Somit muss die glänzende Magnetenoberfläche mit „N“ für Norden beklebt werden. Wird der Aufbau laut Abb. 6 zur Gegenprobe leicht verändert, ergibt sich Abb. 7.



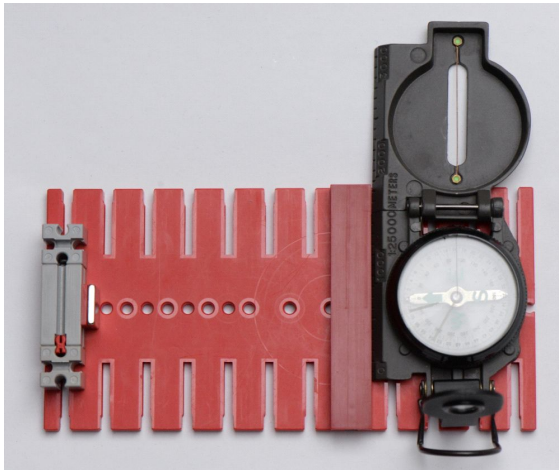


Abb. 6: Vorrichtung zur Bestimmung der magnetischen Ausrichtung: Der Kompass zeigt an, dass links Norden ist.

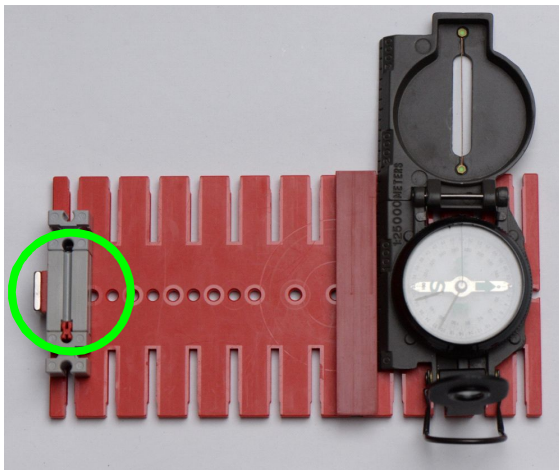


Abb. 7: Vorrichtung zur Bestimmung der magnetischen Ausrichtung: Der Kompass zeigt an, dass links Süden ist.

## Details des Magnetrührers

### Antrieb

Der verbaute Encodermotor hat eine Nenn-drehzahl von  $175 \text{ min}^{-1}$  (175 Umdrehungen pro Minute). Bei kommerziellen Laborgeräten sind  $1500$  bis  $2000 \text{ min}^{-1}$  üblich. Deshalb wurde im Modell unter Berücksichtigung der Leistungsfähigkeit des Encodermotors ein Getriebe eingebaut. Es hat ein Übersetzungsverhältnis von 1:5, d. h. ausgehend von der Nenndrehzahl beträgt die maximale Drehzahl der Drehscheibe  $875 \text{ min}^{-1}$ .

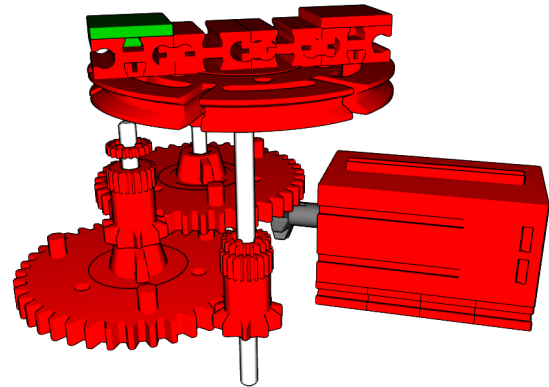


Abb. 8: Antriebsmotor und Getriebe

### Sicherung der Magnete

Wie oben beschrieben beträgt die maximale Drehzahl des Magnetrührers  $875 \text{ min}^{-1}$ . Um zu vermeiden, dass die Magneten aus den Nuten der Drehscheibe herausgeschleudert werden, werden sie wie in Abb. 9 befestigt:

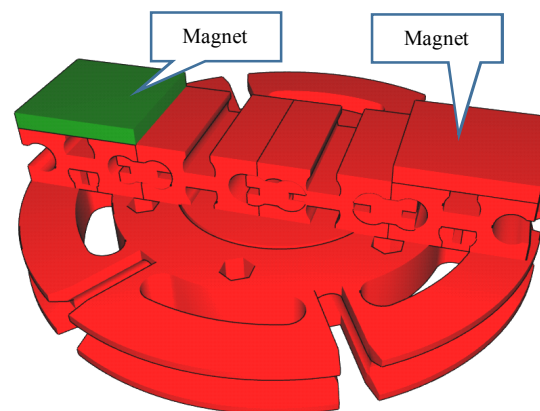


Abb. 9: Befestigung der Magnete an der Drehscheibe

### Sicherheitshinweis



Bei Inbetriebnahme des Magnetrührers ohne das Gehäuse von Abb. 10 ist die Verwendung einer Schutzbrille erforderlich, um Augenverletzungen durch eventuell umherfliegende Teile zu vermeiden. Das gilt auch für Personen im Umfeld.

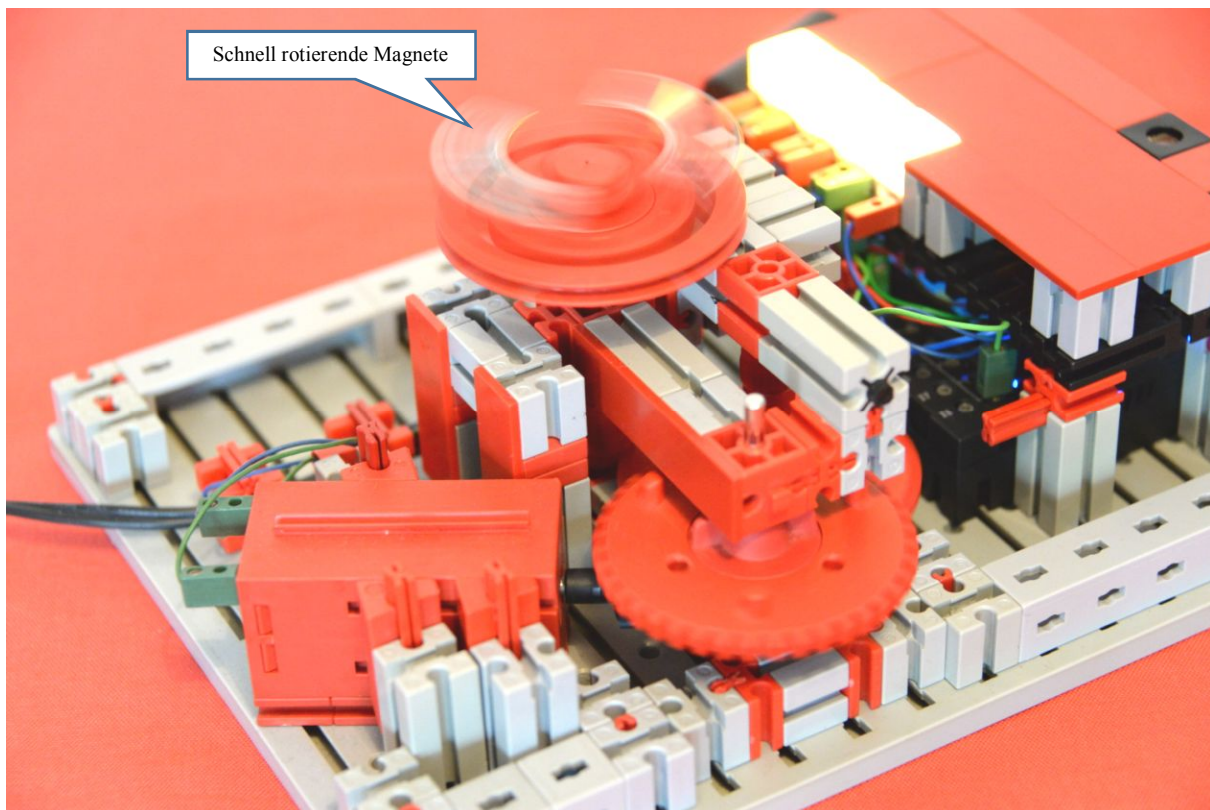


Abb. 10: Betrieb des Magnetrührers ohne Gehäuse zu Testzwecken

### Gehäusekonstruktion

Die Gehäusekonstruktion hat zwei Hauptmerkmale. Zum einen ist sie leicht demon- tierbar, wie Abb. 11 zeigt.

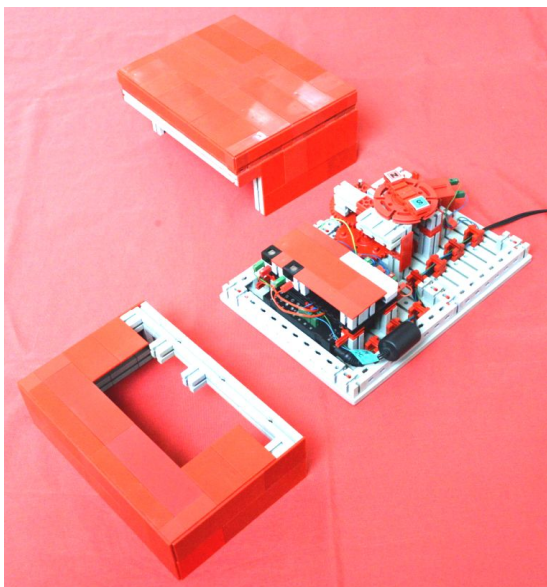


Abb. 11: Magnetrührer mit abgenommenen Gehäuseteilen

Als weiteres Hauptmerkmal zeigt Abb. 12, dass das Gehäuse leicht rundherum wasser- dicht verschlossen werden könnte. Hierzu wäre eine durchsichtige selbstklebende Kunststoffolie vollflächig auf die glatten Gehäuseoberflächen aufzubringen. Die Bedienbarkeit wäre aufgrund der Ver- wendung optischer Taster weiterhin unver- mindert gegeben.

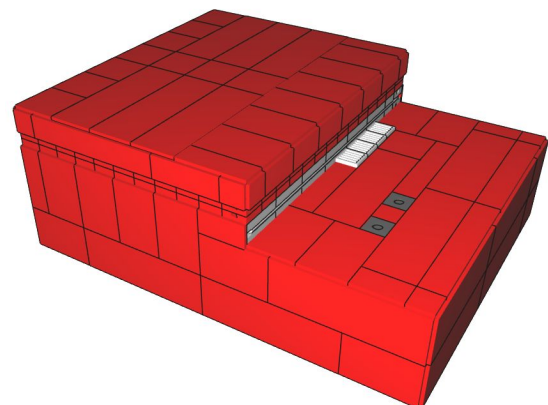


Abb. 12: Gesamtansicht des Magnetrührers mit geschlossenen Gehäuseteilen

**Bedienelemente**

Als Bedienelemente verfügt der Magnet-  
rührer über zwei optische Taster. Wie oben  
erwähnt, ist der Vorteil der optischen Taster  
die einfache Gehäuseabdichtung. Damit sie  
funktionieren, muss im Raum eine Mindest-  
helligkeit vorherrschen, was in der Praxis  
kein Problem sein sollte (wer arbeitet schon  
in einem dunklen Labor).

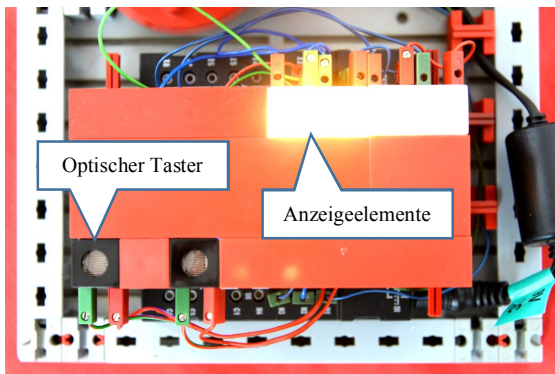


Abb. 13: Bedienelemente und Anzeigeelemente  
des Magnetrührers bei geöffnetem Gehäuse

Mit dem rechten optischen Taster wird eine  
schnellere Zieldrehzahl eingestellt, mit dem  
linken eine geringere. Beim Modellaufbau  
wurden als optische Taster die üblichen  
fischertechnik-Fotowiderstände verwendet.  
Diese sind in der Lage, analoge Helligkeits-  
messwerte für den Robotics TXT Controller  
(RTXTC) bereitzustellen (Einstellung  
A5k). Das Grundprinzip der Erkennung der  
Bedienung ist der schnell eintretende  
Helligkeitsunterschied.

Innerhalb von 500 ms wird der aktuelle  
Helligkeitswert der Fotowiderstände zwei  
Mal gemessen. Eine gewünschte Bedienung  
wird erkannt, wenn zwischen den beiden  
Messungen eines Fotowiderstandes inner-  
halb dieser Zeit eine hinreichend starke  
Abdunklung erfolgte (ROBO Pro: -800%).  
Weiterhin wird eine gewünschte Bedienung  
immer nur dann als solche gewertet, wenn  
die Helligkeitsverminderung bei nur einem  
der beiden Fotowiderstände auftrat – so hat  
das plötzliche Ausschalten der Zimmer-  
beleuchtung keinen Einfluss.

Abb. 14 zeigt das verwendete Unterpro-  
gramm. Es wird in einem eigenen Haupt-  
programm (Task) ständig durchlaufen. Je  
nach erkannter Bedienung wird die Soll-  
Drehzahl um 25 % erhöht oder erniedrigt.  
Nicht sinnvolle Bedienungen werden igno-  
riert, z. B. Drehzahlverminderungen bei  
stehendem Magnetrührer.

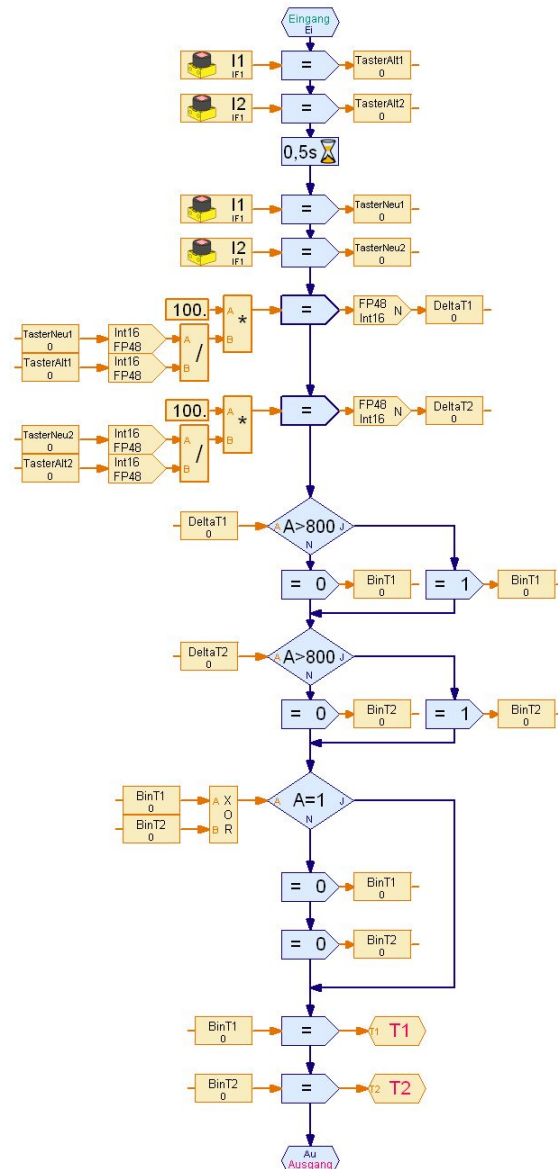


Abb. 14: ROBO Pro Unterprogramm der  
optischen Taster, Ergebnisausgabe unter  
T1 und T2

## Anzeigeelemente

Die Anzeigeelemente dienen als Geschwindigkeitsanzeige und haben gleichzeitig zwei Funktionen. Einerseits wird die Soll-Drehzahl angezeigt und andererseits die aktuelle Ist-Drehzahl. Zur Anzeige werden vier einzelne Lampen L1, L2, L3 und L4 verwendet. L1 steht für die geringste Drehzahl und L4 für die höchste.

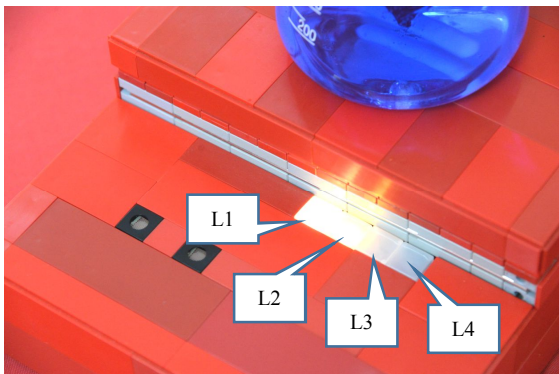


Abb. 15: Anzeigeelemente des Magnetrührers, mit Lampe L1, L2, L3 und L4

Die aktuelle Ist-Drehzahl wird durch kontinuierliches Leuchten dargestellt. In Abb. 15 z. B. leuchten die Lampen L1 und L2. Das bedeutet, dass der Magnetrührer demnach mit 50 % Geschwindigkeit betrieben wird. Ist die Soll-Drehzahl nicht erreicht, blinkt die entsprechende Lampe so lange, bis Ist-Drehzahl die Soll-Drehzahl erreicht hat. Wenn alle Lampen abgeschaltet sind, steht der Magnetrührer.

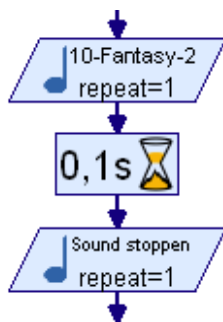


Abb. 16: ROBO Pro-Programmabschnitt für eine nach 100 ms abgebrochene Geräuschabspielung

## Signaltöne

Immer, wenn über die optischen Taster eine 25 %ige Geschwindigkeitsänderung vor-

gewählt wurde, ertönt ein Signalton. Damit bekommt der Bediener eine akustische Rückmeldung zu seinem Bedienvorgang.

Da sich eigene Geräusche derzeit nur sehr umständlich in den RTXTC einspielen lassen, wird ein vorhandenes Geräusch verwendet und nach 100 ms abgebrochen.

## Rampenfahrweise

Immer, wenn die Ist-Drehzahl ungleich der Soll-Drehzahl ist, muss die Ist-Drehzahl so lange angepasst werden, bis beide gleich sind. Das erfolgt durch ein eigenes Hauptprogramm. In ROBO Pro kann die Drehzahl im Bereich von 0 bis 512 eingestellt werden. Alle 100 ms erfolgt ein Drehzahlvergleich. Im Fall der Ungleichheit wird eine Anpassung um 1 durchgeführt. Es wird somit stets eine lineare Geschwindigkeitsanpassung vorgenommen.

## Rührtrombe



Abb. 17: Im Erlenmeyerkolben gerührte Lösung bei 75 % Drehzahl

Abb. 17 zeigt die im Kolben gerührte Lösung und damit die Funktionsfähigkeit des Gesamtaufbaus:

## Einschränkungen

Der RTXTC wurde verwendet, weil er Geräusche abspielen kann. Er hat jedoch bei der verwendeten Firmware 4.2.3 den gravierenden Nachteil, dass er nach dem Einstecken des Steckernetzteils nochmal extra über den eingebauten Taster unter der Anzeige eingeschaltet werden muss. Das kann das Vorgängermodell, der Robo TX Controller (RTXC), besser. Wenn das Gehäuse also wirklich mit Klebefolie flüssigkeitsdicht verklebt werden sollte, muss der RTXC zur Automatisierung verwendet werden.

## Sicherheit beim Umgang mit Neodym-Magneten



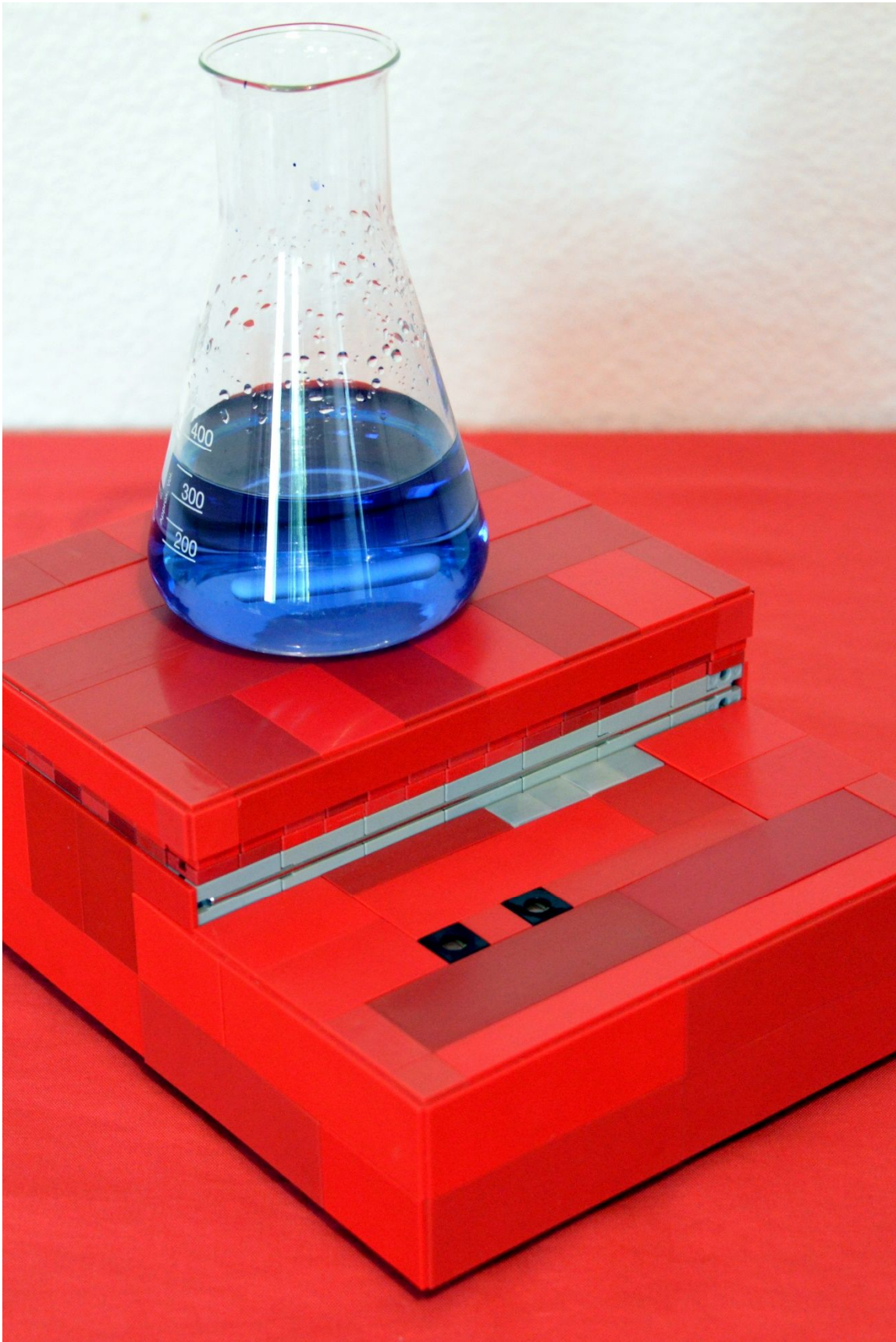
Bei aller Experimentierfreude ist stets auf die eigene Sicherheit zu achten sowie auf Personen im Umfeld. Aus diesem Grund sind unbedingt die Gefahrenhinweise des Lieferanten zu lesen und zu

beachten. Übliche Gefahren sind beispielsweise Verschlucken, Zersplittern beim Herunterfallen und der Einfluss auf Herzschrittmacher oder andere empfindliche Geräte wie Armbanduhren. Von mechanischer Bearbeitung (bohren, schleifen) ist dringend abzuraten.

Unter [4] findet ihr ein Video des Modells. Unter den Referenzen [5, 6] findet ihr professionelle Magnetrührer.

## Quellen

- [1] Laborshop 24 Laborbedarf: <https://www.laborshop24.de>
- [2] Wikipedia: [Magnetrührer](#), *Rührfisch*
- [3] Quelle für Neodym-Magnete: <https://www.supermagnete.de>
- [4] Gail, Andreas: *fischertechnik: Magnetrührer*. [YouTube-Video](#), 2016.
- [5] IKA-Werke GmbH & Co. KG: [Magnetrührer](#).
- [6] Heidolph: [Magnetrührer](#).



*Abb. 18: Magnetrührer: Gesamtansicht*

Modell

## Anwendungen für Magneten (4): Lautsprecher

Andreas Gail

*Von der hier gezeigten Anwendung für Magneten hat sicher jeder schon mal gehört. Es geht um den Lautsprecher, der im klassischen Kofferradio oder den Boxen der Stereo-Anlage verbaut ist. Das Grundprinzip kann mit fischertechnik anschaulich nachgebaut werden.*

### Was wird gebraucht?

Neben fischertechnik-Teilen werden ein paar Dinge benötigt, die nicht zum Standard-Baukastensystem gehören:

- a) Zwei Neodym-Magneten 1,5 x 1,5 cm [1],
- b) PWM-Tongenerator für Robo(tics) TX(T) Controller [2] und 3,5 mm mono Klinkenstecker, *oder* ein einfaches Radio mit Kopfhöreranschluss und passendem Stecker,
- c) leichte Bastler-Pappe (max. 300 g/m<sup>2</sup>) im Format DIN A3.

### Die Lautsprechermembran

Zunächst wird die konusförmige Lautsprechermembran aus der Bastler-Pappe hergestellt. Abb. 14 bildet eine geeignete Vorlage dafür. Es ist wichtig, dass ein Ausdruck dieser Zeichnung so erfolgt, dass die Linien unten links jeweils 20 cm lang sind. Dann geht es ans Ausschneiden. Das Ergebnis sollte dann etwa aussehen wie in Abb. 1 gezeigt. Die späteren Knickstellen sollten mit einem stumpfen Messer eingeritzt werden. Nicht eingeritzt werden darf die Klebelasche zum Zusammenkleben der Membran. Im nächsten Schritt gemäß Abb. 2 wird die Pappe zu einem Konus zusammengeklebt. Die zuvor eingeritzten Laschen werden wie gezeigt umgeknickt. Die Löcher in den 12 Sicken werden einfach mit einem normalen Locher hergestellt.



Abb. 1: Herstellung Lautsprechermembran, Schritt 1: Ausschneiden und Knickstellen einritzen

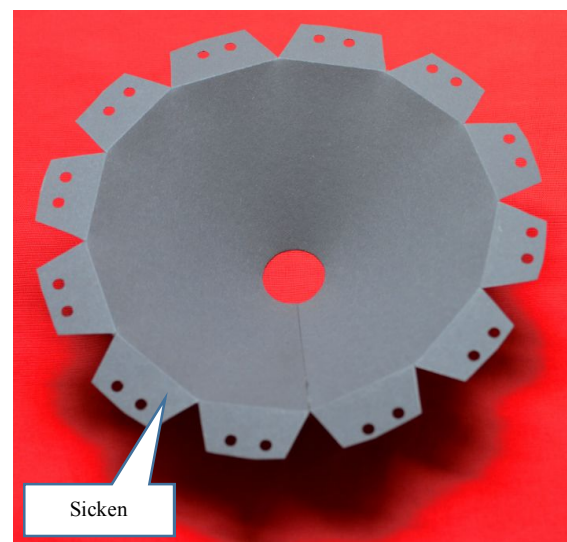


Abb. 2: Herstellung Lautsprechermembran, Schritt 2: Kleben und Lochen

Die Lautsprechermembran gemäß Abb. 2 wird nun in die 60°-Bogenteile eingebaut, wie in Abb. 3 und 4 gezeigt:

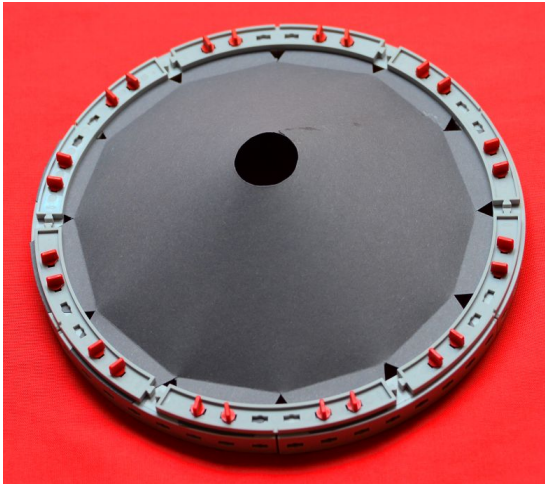


Abb. 3: Herstellung Lautsprechermembran, Schritt 3: Einbau in fischertechnik-60°-Bogenteile, Rückansicht

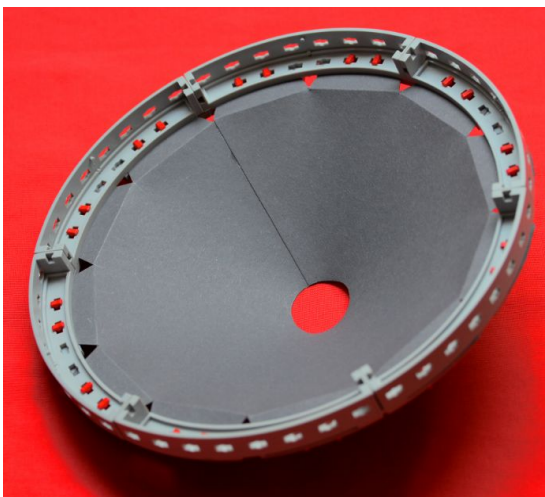


Abb. 4: Herstellung Lautsprechermembran, Schritt 3: Einbau in fischertechnik-60°-Bogenteile, Vorderansicht

Der letzte Schritt zur Herstellung der Lautsprechermembran ist das Einkleben der Magnetplatte. Hierzu wird zunächst das zweite Teilstück laut Abb. 14 ausgeschnitten und die Löcher mit einem Locheisen ausgeschlagen (vgl. Beitrag in ft:pedia 2/2014 [4]). Anschließend werden die fischertechnik-Teile eingebaut, was in Abb. 5 und 6 gezeigt wird. Dieses Innenteil wird nun von der Rückseite (Abb. 3) an den Klebelaschen aufgeklebt. Damit ist die

Lautsprechermembran fertiggestellt und der handwerklich schwierigste Teil ist geschafft.



Abb. 5: Herstellung Lautsprechermembran, Schritt 4: Vorbereitung Magnetplatte, Rückansicht



Abb. 6: Herstellung Lautsprechermembran, Schritt 4: Vorbereitung Magnetplatte, Vorderansicht

## Gesamtaufbau des Lautsprechers

Der Gesamtaufbau ist in Abb. 15 gezeigt. Entscheidend dabei ist, dass der Korb zur Halterung der Lautsprechermembran möglichst stabil und verwindungssteif aufgebaut ist. Die rückseitige Konstruktion zeigen die Abb. 7 und 13. Ein wichtiges Detail wird in Abb. 8 und 9 gezeigt. Hier ist die Spule (Elektromagnet) gezeigt, die sich im Magnetfeld der Dauermagneten befindet. Das ist der sogenannte *Antrieb* des Lautsprechers.



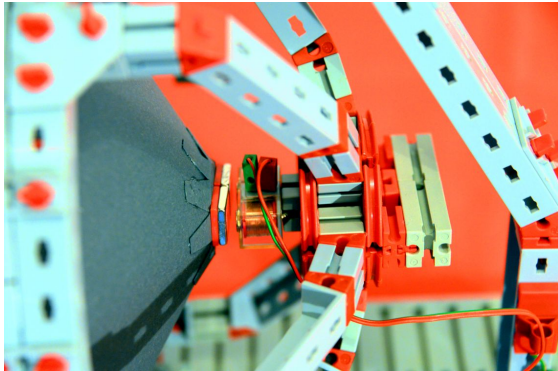


Abb. 8: Seitenansicht des Lautsprechers mit Korb und Antrieb

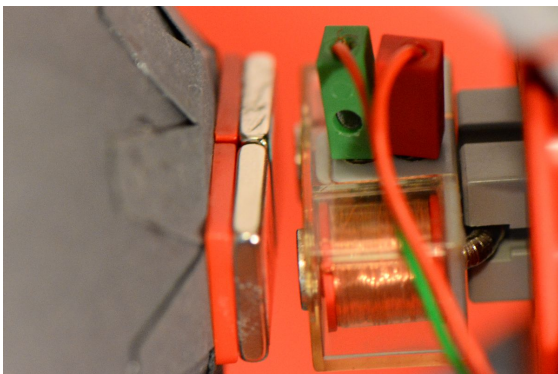


Abb. 9: Spule und Elektromagnet als Antrieb des Lautsprechers

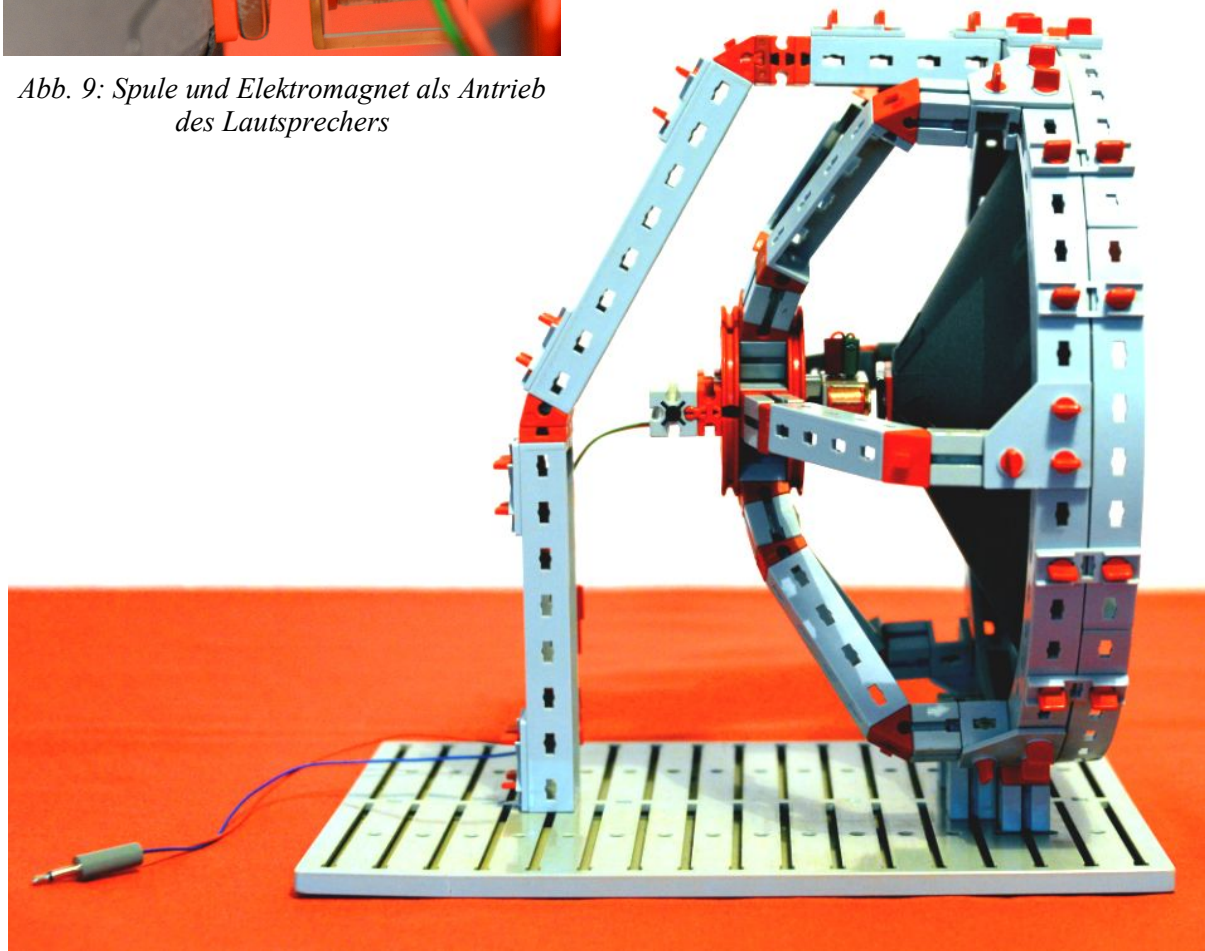


Abb. 7: Lautsprecher Gesamtaufbau, Seitenansicht

## Anschluss des Lautsprechers

Was nützt ein Lautsprecher, wenn er nicht mit dem passenden Eingangssignal versorgt wird. Der hier gezeigte Lautsprecher wird beispielhaft alternativ an zwei Signalquellen angeschlossen. Einmal wird der PWM-Tongenerator verwendet (siehe ft:pedia Ausgabe 4/2015 [2] sowie Abb. 10), es funktioniert aber auch, wie Abb. 11 zeigt, ein einfaches Radio. Hierbei sind die Anschlusspezifikationen des Radios zu beachten. Im vorliegenden Fall sind für den Ohrhöreranschluss 3 bis 8  $\Omega$  angegeben. Das ist der Gleichstrom-Widerstand des anzuschließenden Lautsprechers. Das bedeutet, dass der Widerstand von 3  $\Omega$  nicht unterschritten werden darf. Größere Widerstandswerte sind für das Radio ungefährlich.

Der fischertechnik-Magnet aus Abb. 9 zeigt einen Widerstandsmesswert von  $46 \Omega$  und kann also problemlos angeschlossen werden.

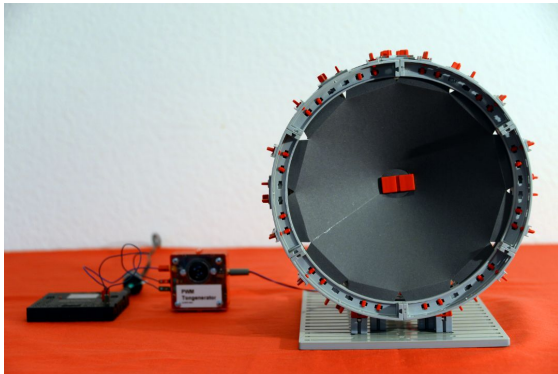


Abb. 10: Betrieb des Lautsprechers mit dem PWM-Tongenerator



Abb. 11: Betrieb des Lautsprechers mit einem einfachen handelsüblichen Radio

## Verbesserungsmöglichkeiten

Der hier vorgestellte Lautsprecher ermöglicht z. B. in der Konfiguration gemäß Abb. 11 problemlos das Verfolgen von Radiosendungen. Trotzdem kann auch dieser Aufbau weiter verbessert werden.

1. Die Verwendung leichter beweglicher Sicken würde die Lautsprechermembran leichter schwingen lassen. Gemäß Abb. 12 könnte für die Sicken ein anderes Material benutzt werden. Auch

möglich wäre ein paralleles Einritzen der Sicken zur Knickkante.

2. Auch der Antrieb als Kombination von Dauermagnet und Elektromagnet könnte bestimmt effektiver aufgebaut werden, insbesondere, wenn weniger Masse bewegt werden muss, um die Lautsprechermembran zum Schwingen zu bringen.

Ein Video des fischertechnik-Lautsprechers mitsamt Hörproben gibt es unter [5]. Viel Spaß beim Experimentieren!

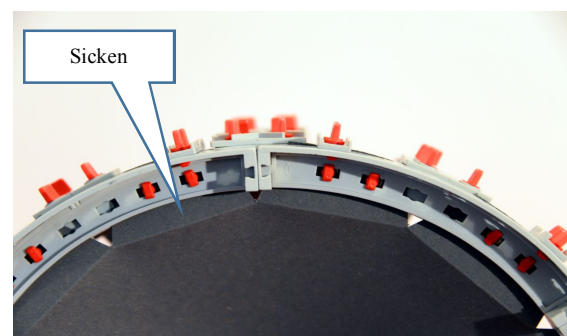
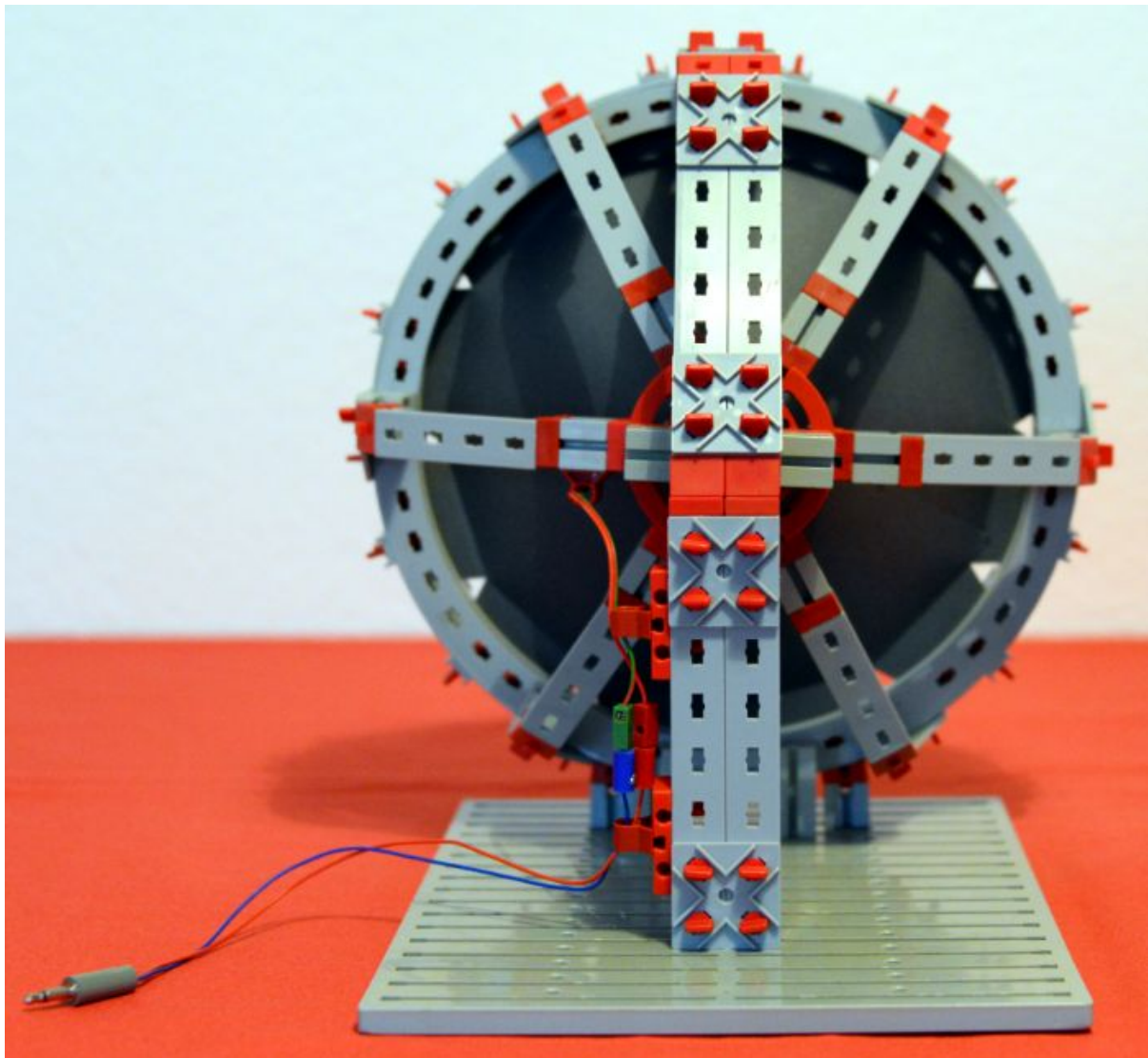


Abb. 12: Sicken zur Halterung der konusförmigen Lautsprechermembran

## Referenzen

- [1] Gail, Andreas: *Magnetkupplungen und Magnetrührer*. [ft:pedia](#) 4/2016 (in dieser Ausgabe).
- [2] Gail Andreas: *PWM-Tongenerator für Robo TX(T) Controller*. [ft:pedia](#) 4/2015, S. 31-39.
- [3] Wikipedia: [Lautsprecher](#).
- [4] Gail, Andreas: *Einstieg in Experimente mit Lasern*. [ft:pedia](#) 2/2014, S. 17.
- [5] Gail, Andreas: *fischertechnik: Lautsprecher*. [YouTube-Video](#) zum Modell.



*Abb. 13: Gesamtaufbau des Lautsprechers von hinten mit 3,5 mm mono Klinkenstecker*

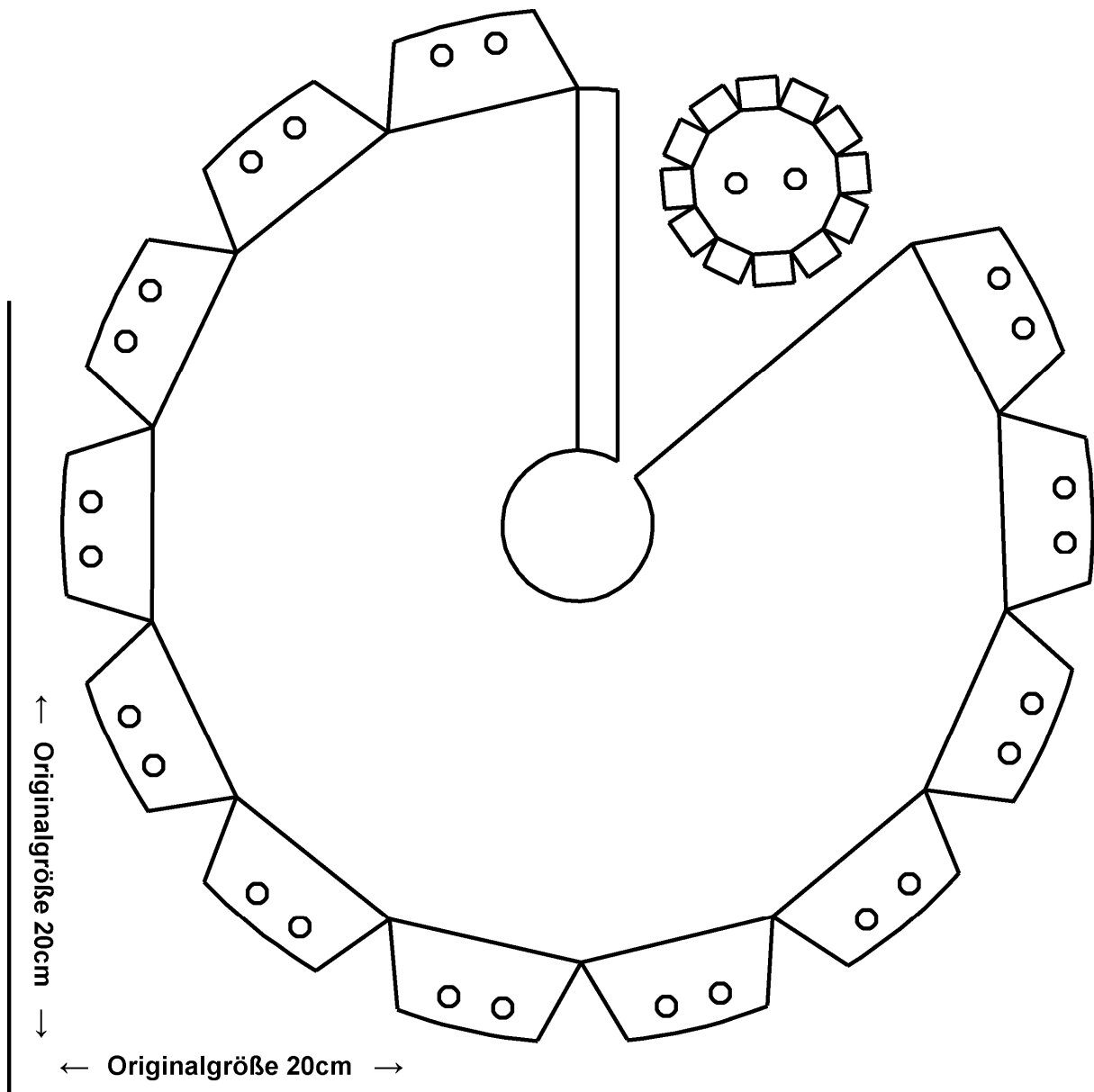
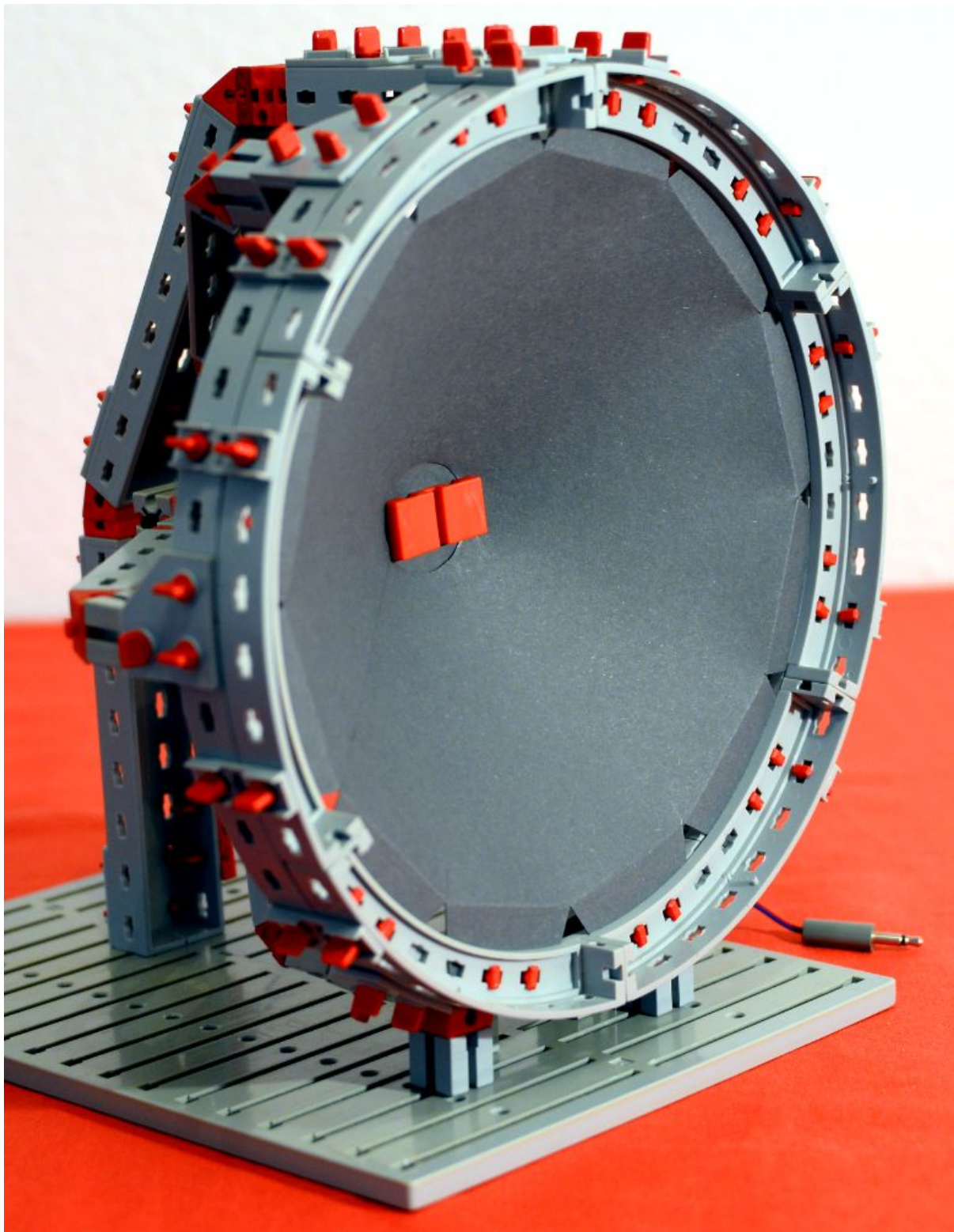


Abb. 14: Lautsprechermembran, Vorlage zur Herstellung aus Pappe (max. 300 g/m<sup>2</sup>).  
Der Ausdruck dieser Abbildung sollte so erfolgen, dass die waagerechte und die senkrechte  
Linie unten links jeweils 20 cm lang ist (ggf. beim Ausdrucken den Zoom-Faktor anpassen).



*Abb. 15: Gesamtaufbau des Lautsprechers mit 3,5 mm mono Klinkenstecker*

Elektromechanik

## Der Elektromagnet: Was kann er (vertragen)?

Rüdiger Riedel

Der Elektromagnet von fischertechnik [1] ist zusammen mit den starken Neodym-Magneten im Synchronmotor wieder aktuell geworden [2, 3]. Ansonsten führt er ein eher bescheidenes Dasein; in der aktuellen Einzelteilliste sind sowohl der ältere Typ Nr. 31324 als auch der neuere Typ Nr. 32363 nicht einmal mehr enthalten. Was ich besonders bei meinen experimentellen Motoren [4] vermisste, sind technische Daten wie Anziehungskraft und Belastbarkeit. Das will ich hier beleuchten.

Beim experimentellen Aufbau von historischen Motormodellen und ungewöhnlichen Anordnungen ist es oft hilfreich, eine möglichst große Magnetkraft nutzen zu können. Ein erster Schritt ist die Nutzung des fischertechnik-Elektromagneten neueren Typs: Er ist deutlich stärker als der ältere. Außerdem kann man die Spannung und damit den Strom durch den Magneten vergrößern. Das hilft vielfach – aber was ist zulässig, und wie weit kann ich gehen, ohne den Magneten zu beschädigen?

Ein Hinweis vorneweg: Da die Datenlage so unklar ist wie im Weiteren beschrieben, bin ich wahrscheinlich mit meinen Versuchen an die Grenzen des Geräts (oder sogar darüber hinaus) gegangen. Wer das nachmacht, sollte sich dessen bewusst sein. Und bitte keine zu strengen Maßstäbe an diese Versuche anlegen: Mir ging es um praktische Erkenntnisse, nicht um exakte Wissenschaft.

Abb. 1 zeigt Bilder der beiden fischertechnik-Elektromagneten (31324, 32363) aus der ft-Datenbank. Dazu passend gibt es eine rechteckige (31325) und eine runde Rückschlussplatte (31326).

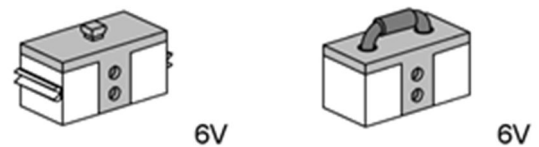


Abb. 1: fischertechnik-Elektromagnete neueren (32363, links) und älteren Typs (31324, rechts)

### Was leistet der Elektromagnet?

Als Kraftmessapparat verwende ich eine angepasste Balkenwaage aus Hobby 1-1, Seite 27 [5] (Abb. 2). Als Gewichte nutze ich Münzen, die ich mit einer Küchenwaage im Zehnerpack abgewogen habe (Tab. 1).

Münze	Gewicht
1 ¢	2,3 g
2 ¢	3,1 g
5 ¢	3,9 g
10 ¢	4,1 g
20 ¢	5,7 g

Tab. 1: Münzgewichte

Die Werte stimmen recht gut mit den offiziellen überein [6].

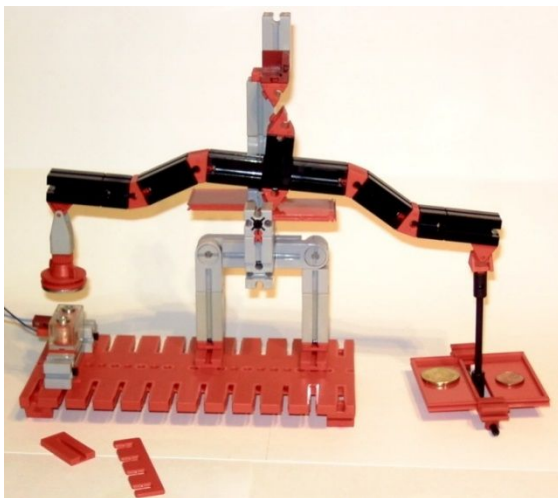


Abb. 2: Kraftmessapparatur

Ich habe so lange Gewichte aufgelegt, bis der Magnet sie gerade noch halten konnte. Aus den Gewichten lassen sich die Haltekräfte des Magneten wie folgt errechnen: Die Kraft  $F$  entspricht der Masse  $m$  mal (Erd-) Beschleunigung  $g$  ( $9,81 \text{ m/s}^2$ ):

$$F = m \cdot g$$

*Beispiel:* Ein Gewicht von je einer 5 ¢- und 10 ¢-Münze ergibt 8 g; das entspricht einer Kraft von 0,078 N.

Die Anziehungskraft ist stark abhängig vom Abstand zwischen Magnet und Rückschlussplatte. Ich habe den Abstand mit Hilfe von Platten der Stärken 2 / 3,75 / 4 / 5 / 6 und 7,5 mm eingestellt.

Die Ergebnisse lassen sich sauber und übersichtlich in einer Excel-Tabelle darstellen; die Auswertung in Diagrammen zeigen die folgenden Abbildungen.

### Elektromagnet älteren Typs (31324)

Die maximale Anziehungskraft wird bei direktem Aufliegen der Rückschlussplatte erreicht. Um ein „Kleben“ durch den Restmagnetismus (*Remanenz*) im Eisen des abgeschalteten Elektromagneten zu verhindern, habe ich ein Stück Papier dazwischen gelegt (ca. 0,1 mm dick).

Bei 6 V hält der Magnet bereits 1,5 N; mehr kann ich mit meiner Messapparatur nicht einstellen (Abb. 3).

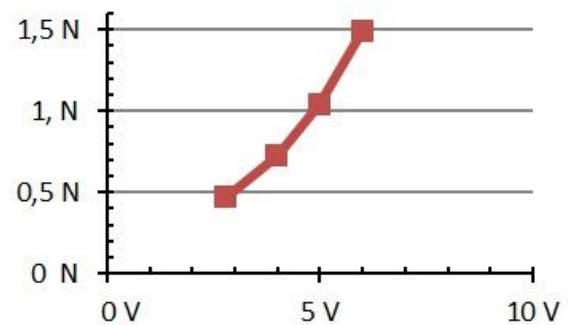


Abb. 3: Maximale Anziehungskraft

In den Abb. 4 und 5 wird die Haltekraft in Abhängigkeit von der Spannung mit dem Abstand als Parameter gezeigt.

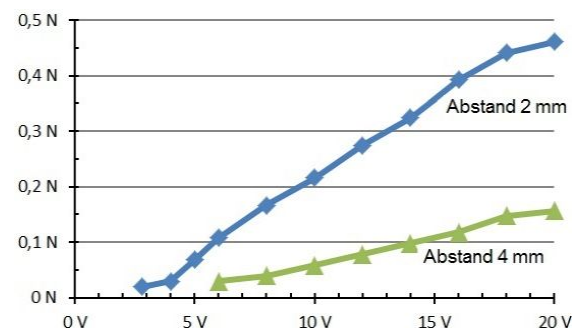


Abb. 4: Abstand von 2 und 4 mm

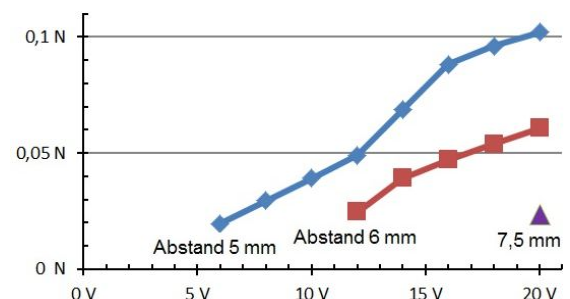


Abb. 5: Abstand von 5, 6 und 7,5 mm

Die Abb. 6-8 zeigen die Anziehungskraft in Abhängigkeit vom Abstand mit der Spannung als Parameter.

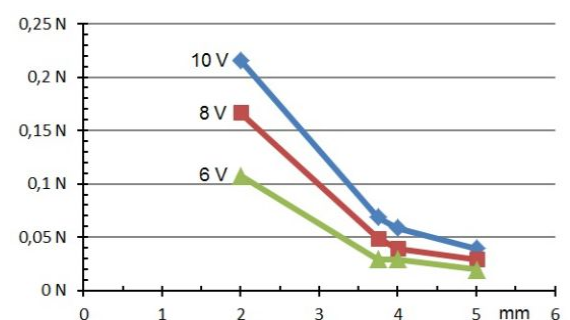


Abb. 6

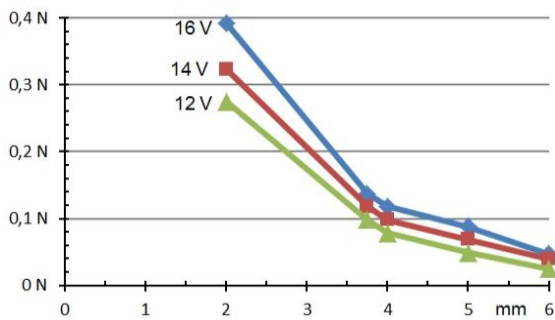


Abb. 7

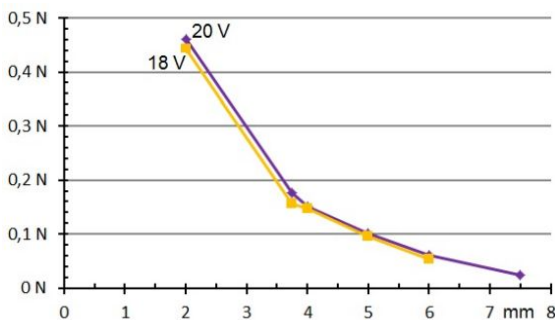


Abb. 8: Die Kurven für 18 V und 20 V liegen sehr dicht beieinander

Vermutlich wird bei 20 V (Abb. 8) und einem Strom von mehr als 450 mA die Eisensättigung erreicht; eine weitere Steigerung der Spannung bringt dann nur noch eine geringe Steigerung der Anziehungskraft.

**Elektromagnet neueren Typs (32363)**

Die Anziehungskräfte des Elektromagneten neueren Typs sind mehr als doppelt so groß wie die des älteren Typs:

Beim Abstand von 0,1 mm (Papier) wird bereits mit 3 V eine Kraft von 1,5 N erreicht; höhere Kräfte kann ich nicht darstellen, deshalb gibt es hierzu kein Diagramm.

Spannung	Kraft
2 V	0,98 N
3 V	1,5 N
4 V	> 1,5 N

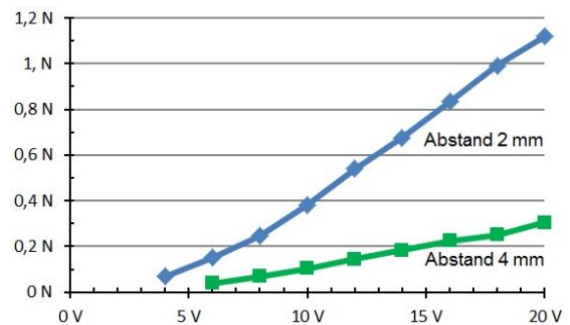


Abb. 9: Abstand von 2 und 4 mm

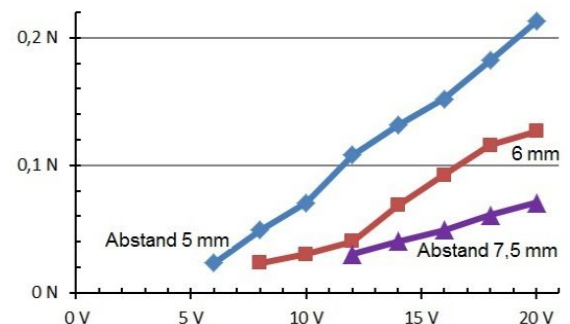


Abb. 10: Abstand von 5, 6 und 7,5 mm

Sogar bei einem Abstand von 10 mm und einer Spannung von 18 V oder 20 V wird noch eine Kraft von 0,03 N erreicht.

In den Abb. 11 bis 13 ist die Anziehungskraft in Abhängigkeit vom Abstand mit der Spannung als Parameter dargestellt.

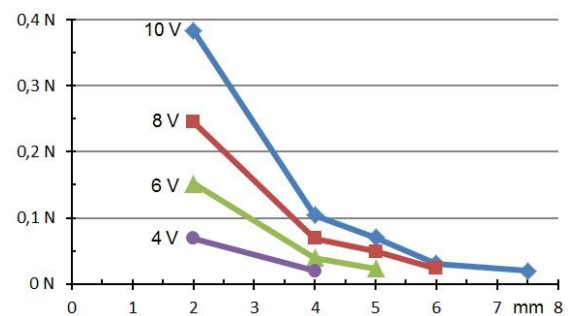


Abb. 11

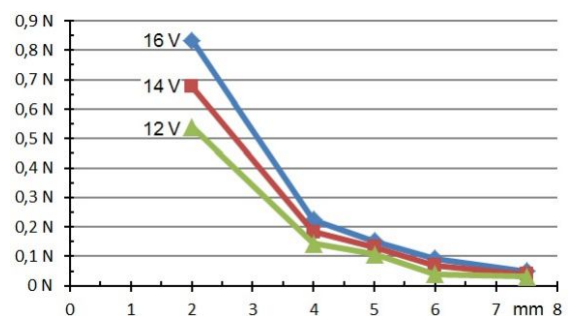


Abb. 12



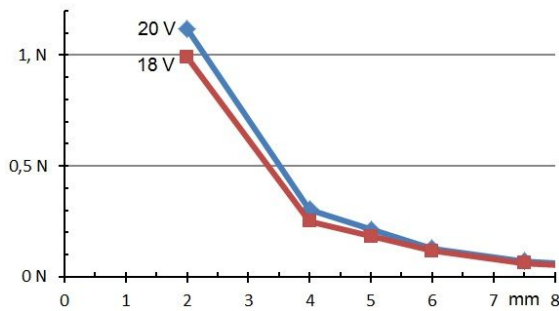


Abb. 13

## Ergebnis

Es lohnt sich offensichtlich, die Spannung auf 18 bis 20 V hochzutreiben: Die Anziehungskraft ist mindestens doppelt so groß wie bei 10 V. Allerdings bleibt die Frage: Verträgt das der Magnet?

### Was verträgt der Elektromagnet?

Natürlich geht der Elektromagnet kaputt, wenn man dem Gehäuse Gewalt antut.

Beim älteren Typ kann man den Eisenkern abschrauben. Bei meinem waren Gehäuse und Spulen nicht festgeklebt, und als eine Spule herausrutschte, ist prompt ein Anschlussdrähtchen abgerissen – also Vorsicht!

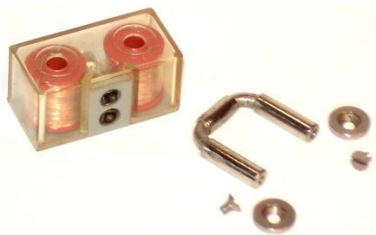


Abb. 14: Demontierter Elektromagnet

Warum macht man so etwas? Weil es Versuche gibt, bei denen eine eisenlose Spule benötigt wird.

Schließlich kann man das Magnetchen durch Überstrom den Hitzetod sterben lassen. Damit das nicht passiert, habe ich die nachfolgenden Versuche unternommen.

Die fischertechnik-Datenbank nennt sowohl für den älteren als auch für den neueren Typ Anschlussdaten von 6 V und 0,2 A; danach hätte der Magnet einen Innenwiderstand von 30  $\Omega$ .

Das kann ich mit meinen gebraucht gekauften nicht nachvollziehen: Sie haben Innenwiderstände von 43,5  $\Omega$  (Typ 31324) und 41,4  $\Omega$  (Typ 32363) – was eher mit den Angaben in Hobby 3 Band 1, Seite 41 übereinstimmt (50  $\Omega$ ) [7].

Nun ist der elektrische Widerstand von Metallen von der Temperatur abhängig. Ausgehend von 20°C erhöht sich der Widerstand von Kupferdraht um den Faktor 0,004 pro Kelvin (K) [8]; umgangssprachlich kann man auch Grad Celsius sagen (°C), der Zahlenwert ist der gleiche.<sup>2</sup> Umgekehrt gilt: Wenn die Messung des Widerstandes  $R$  einen größeren Wert als z. B. 43,5  $\Omega$  ergibt, dann hat sich die Temperatur des Magneten erhöht.

Damit kann ich also durch Messung der Versorgungsspannung  $U$  und des Stromes  $I$  den neuen Widerstand  $R_2$  berechnen und daraus die Temperatur  $T_W$  ermitteln.

Temperaturbeiwert von Kupfer [8]:

$$\alpha = \frac{1}{250} \text{ K}^{-1}$$

$$R_2 = R_1 \cdot (1 + \alpha \cdot \Delta T)$$

Mit  $R = \frac{U}{I}$  beträgt die Temperaturänderung also

$$\Delta T = \frac{U}{\alpha \cdot R_1 \cdot I} - \frac{1}{\alpha}$$

und die Wicklungstemperatur beträgt dann

$$T_w = \Delta T + T_R$$

mit  $T_R =$  Raumtemperatur.

<sup>2</sup> In Technik und Wissenschaft wird unterschieden zwischen der Temperatur  $T$ , gemessen z. B. in

Grad Celsius, und der Temperaturänderung  $\Delta T$ , gemessen in Kelvin.

## Temperaturbeständigkeit

Die Temperaturbeständigkeit von Kupferlackdraht wird in [11] mit 158°C angegeben. Das Kunststoffmaterial der Bausteine ist Polyamid (= PA, Nylon), genauer PA6 [9]. Gemäß Datenblatt „nylatron-6“ der Fa. Polytron Kunststofftechnik [10] sind dies die Dauergebrauchstemperaturen von PA6:

- kurzzeitig (wenige Stunden): bis 160 °C
- dauernd während 5.000 h: bis 85 °C

In der internationalen Normung (IEC 34-1 bzw. IEC 85) werden die Isolierstoffe in Klassen eingeteilt. Für die Wärmeklasse B gilt:

- Grenztemperatur des Isolierstoffes: 130°C

Ich gehe davon aus, dass für die Elektromagnete eine Temperaturgrenze von 130°C gilt. Unter dieser Annahme habe ich die nachfolgenden Versuche durchgeführt.

In den Diagrammen wird als Starttemperatur immer 20°C angegeben, was ungefähr der Raumtemperatur entspricht.

Auf ein oder zwei Grad darf man mich nicht festlegen, das gibt die Genauigkeit der Messungen nicht her. Auch hier gilt wieder: Es kam mir nicht auf exakte Einzelwerte an, sondern auf Trends, die Vergleichbarkeit und die Auslotung der Grenzen, möglichst ohne diese zu überschreiten.

Aus meiner Sicht sind sowohl der ältere als auch der neuere Magnet dauerhaft mit 12 V belastbar. In beiden Fällen ist die Temperaturänderung nach 25 Minuten zum Stillstand gekommen, ohne die Grenztemperatur von 130 °C zu erreichen. Deshalb gibt es keine Diagramme mit kleineren Spannungen.

### Temperaturbelastbarkeit des älteren Typs

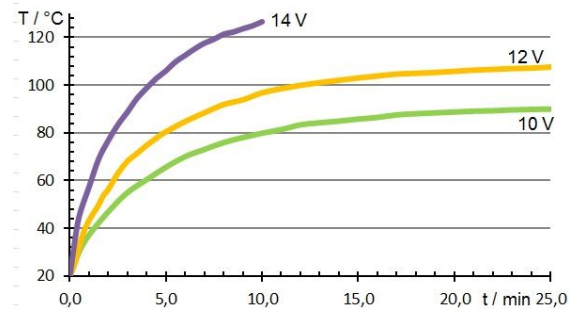


Abb. 15

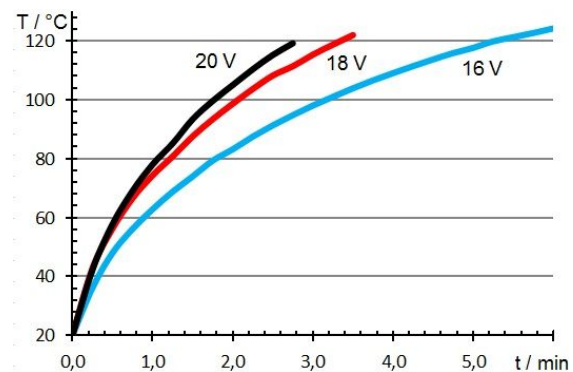


Abb. 16

### Temperaturbelastbarkeit des neueren Typs

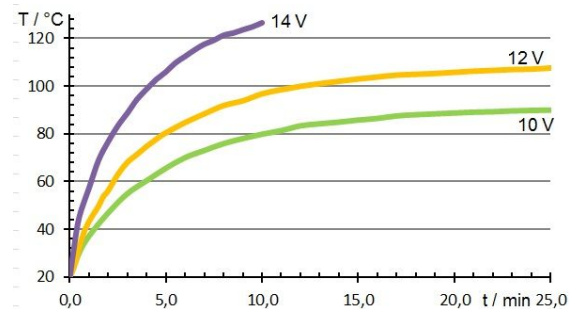


Abb. 17

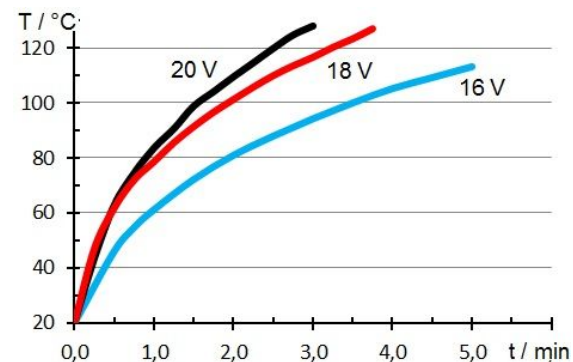


Abb. 18

## Ergebnis der Temperaturbelastungen

Das Temperaturverhalten beider Magnete ist – im Gegensatz zur Kraftentwicklung – sehr ähnlich. Das deutet darauf hin, dass die Spulen praktisch gleich sind, aber der neuere Typ den besseren Eisenkern besitzt. Er hat eine wesentlich höhere Permeabilitätszahl ( $\mu_r$ ). Zusammenfassend können wir festhalten:

- Beide Magnete sind für Dauerbelastung mit 12 V geeignet.
- Bei Kurzzeitbelastung kann die Spannung für zwei oder drei Minuten auf 18 V erhöht werden, um die größere Anziehungskraft nutzen zu können.

Meine Magnete haben alle genannten und noch ein paar zusätzliche Versuche klaglos überstanden; optisch ist keinerlei Veränderung feststellbar.



Abb. 19: So sehen die Magnete nach den Versuchen aus

Das Gehäuse wird zwar heiß, aber verbrennen kann man sich daran nicht. Ein Gefahrhinweis ist also nicht nötig.

Im Modell ist der Eisenkern nicht zugänglich. Wird er dennoch berührt, passiert nichts außer einem „Autsch“. Die Wärmekapazität ist zu gering um Schäden anzurichten.

## Referenzen

- [1] Stefan Falk: *Perlentauchen (Teil 3)*. [ft:pedia 1/2013](#), S. 22-31.
- [2] Dirk Fox, Thomas Püttmann: *Technikgeschichte mit fischertechnik*. dpunkt.verlag, Heidelberg, 2015.
- [3] Matthias Dettmer: *Synchronmotoren*. [ft:pedia 2/2016](#), S. 48-52.
- [4] Rüdiger Riedel: *Der etwas andere Motor*. [ft:pedia 3/2016](#), S. 22-25.
- [5] Fischer-Werke: *Hobby 1 Band 1*. 1972.
- [6] Muenzen.eu: [Euro-Umlaufmünzen](#).
- [7] Fischer-Werke: *Hobby 3 Band 1*. 1971.
- [8] BBC: *Taschenbuch für Schaltanlagen*. 7. Auflage, 1979.
- [9] fischertechnik GmbH: [Qualität](#).
- [10] Polytron Kunststofftechnik: [Polyamid 6](#).
- [11] Elektrisola: [Kupferlackdraht nach IEC-Europa](#).

Hinweis zur Quelle [8]: BBC ist die Abkürzung der Schweizer Firma Brown, Boveri und Cie., die es inzwischen nicht mehr gibt. Das Buch wurde von der damals großen Mannheimer Niederlassung herausgegeben.

Elektromechanik

## Funktionsmodelle von Gleich- und Wechselstrommotoren

Rüdiger Riedel

*Wegen der Eleganz und Einfachheit wollte ich einen Elektromotor mit einem Reedschalter steuern. Die ersten Versuche waren schnell beendet, weil die Kontakte miteinander verschweißten – der Schalter schaltete nicht mehr. Üblicherweise wird dieser Schaltertyp nur als Sensor und nicht als „Lastschalter“ verwendet [4]. Doch mit ein paar Überlegungen und einer einfachen Maßnahme ist der Weg frei für einen einfachen Gleichstrommotor – als Anlaufhilfe für einen Synchronmotor.*

Das „Anwerfen“ eines Synchronmotors ist etwas trickreich: Man muss manuell die Laufgeschwindigkeit einigermaßen treffen, damit er weiterläuft. Wie praktisch wäre da doch ein Gleichstrommotor als Anlaufhilfe...

Beginnen wir zunächst mit einem Bild des Funktionsmodells:

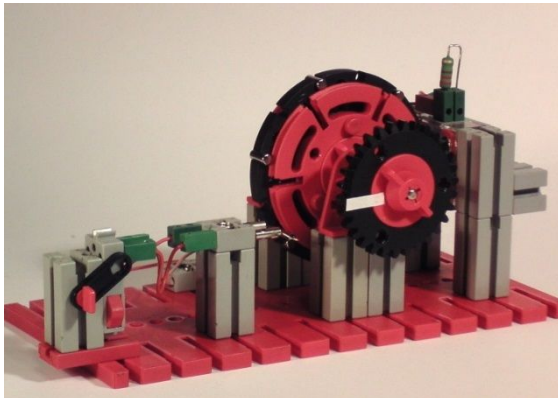


Abb. 1: Reedkontakt als Kommutator

Links im Bild erkennt man den Einschalter, daneben folgen der Reedkontakt und dann der Rotor mit Zahnrad Z30 einschließlich Markierung (darauf komme ich später zurück). Dahinter erkennt man oben den Antriebsmagnet, ausgerüstet mit einem Widerstand. Die Verdrahtung ist noch unvollständig und es fehlt der Synchron-(Wechselstrom-) Anteil.

### Der Reedkontakt

Ich verwende Miniatur-Reedkontakte mit den folgenden Eigenschaften:

<b>Schaltleistung</b>	10 VA/50 V
<b>Schaltstrom</b>	max. 0,5 A
<b>Maße (L x Ø)</b>	15 x 2,2 mm

Tab. 1: Werte des Reedkontakts

Das sieht eigentlich gut aus, reicht aber nicht, wie meine fehlgeschlagenen Versuche gezeigt haben. Also muss die Physik her.

Was passiert, wenn ein stromdurchflossener Elektromagnet abgeschaltet wird? Die Lenzsche Regel besagt, dass der Strom einer Induktivität (und das ist unser Magnetchen auch) nicht plötzlich geändert werden kann. Werden also die Schalterkontakte geöffnet, dann sucht sich der Strom einen Weg durch die Luft, er bildet einen (hier sehr kleinen) Funken. Um diesen Strom durch die Luft zu treiben, muss die Spannung stark ansteigen.

Die Spannungsspitze über dem Schalter erreicht sehr hohe Werte, wie man mit einer Glimmlampe aus einem so genannten Spannungsprüfer nachweisen kann (meine hat eine Ansprechspannung von ca. 70 V).

Oder man greift die entstehende Spannung an der Spule mit den Händen ab, siehe die „Elektrisiermaschine“ in [5].

Solche Spannungsspitzen von mehreren 100 V machen den üblichen Schaltern, Tastern und Relais nichts aus; sie sind ausreichend robust und die Kontakte öffnen weit genug. Der Reedkontakt öffnet nur einige Zehntel mm, ist also viel empfindlicher. Was ist zu tun? Der Funke muss weg.

### Funkenlöschung

Der dahinter stehende Gedanke: Der von der Induktivität getriebene Strom muss beim Abschalten weiter fließen können, bis er durch die Dämpfung des Innenwiderstands abgeklungen ist.

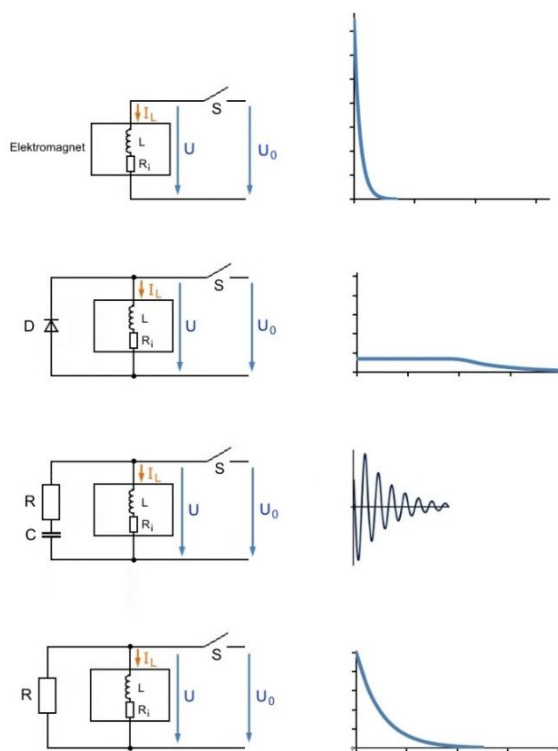


Abb. 2: Abschaltvorgänge mit und ohne Beschaltung

Abbildung 2 zeigt links den Elektromagneten ohne und mit Beschaltung und rechts den zugehörigen Verlauf der Spannung  $U$  nach Abschaltung der Versorgungsspannung  $U_0$ . Der Elektromagnet ist mit seiner Induktivität  $L$  und seinem Widerstand  $R_i$  dargestellt.

*Fall 1:* Ohne äußere Beschaltung erreicht die Abschaltspannung  $U$  mehrere 100 V, klingt aber sehr schnell ab.

*Fall 2:* Die Diode  $D$  ist in Bezug zur Versorgungsspannung  $U_0$  in Sperrrichtung geschaltet. Beim Abschalten übernimmt sie den induktiven Strom. Dieser klingt relativ langsam ab mit der Zeitkonstanten  $\frac{L}{R_i}$ .

*Fall 3:* Der Induktionstrom  $I_L$  fließt in den Kondensator  $C$ , worauf sich zusammen mit der Induktivität  $L$  eine gedämpfte Schwingung ergibt. Der Widerstand  $R$  bewirkt eine zusätzliche Dämpfung.  $C$  muss etwa  $15 \mu\text{F}$  groß sein (dies bitte nicht verwechseln mit einer Beschaltung zur Hochfrequenzdämpfung, da ist der Kondensator viel kleiner).

*Fall 4:* Der Widerstand  $R$  beträgt etwa das zwei- bis sechsfache von  $R_i$ , dadurch wird die Abschaltspannung über dem Schalter  $S$  auf weniger als  $U_0$  mal  $\frac{R-R_i}{R_i}$  reduziert.

Die Diode gefiel mir am besten.



Abb. 3: Löschiode am Magnet

Der Motor läuft damit einwandfrei, der Reedkontakt zeigt keine Schwächen. Die Leerlaufdrehzahl habe ich mit 26 V und 0,12 A auf 1.000 U/min gebracht. Der Nachteil: Bei falscher Polung der Diode gibt es einen Kurzschluss!

Deswegen habe ich auch den Fall 4 mit einem  $220 \Omega$ -Widerstand ausprobiert. Und siehe da, der Motor läuft noch besser, mit 12,8 V und 0,085 A habe ich ebenfalls 1.000 U/min erreicht und mit 23 V und 0,13 A sogar 1.500 U/min. Die geringen Verluste durch den Widerstand, nämlich ca. 20 % des Gesamtstroms, kann man hinnehmen.



Abb. 4: Der von mir bevorzugte Löschwiderstand

In dieser Konfiguration ist der Gleichstrommotor schon mehr als fünf Stunden gelaufen und der Reedkontakt hat wohl eine halbe Million Schaltspiele hinter sich.

Sollte im Übereifer mal zu große Spannung auf den Reedkontakt gekommen sein und die Kontakte sind verschweißt, kann man diese meistens wieder trennen, indem ein Magnet darüber gestrichen wird: Sind beide Zungen vom gleichen Pol magnetisiert, stoßen sie sich ab. Bei mir hat es geklappt.

## Der Rotor, auch Läufer genannt

Der Rotor soll mit Wechselstrom auch synchron laufen – warum hat er dann sechs (und nicht wie sonst üblich 12 [6]) Magnete?

Der „Stromwender“ oder „Kommutator“ besteht aus den Magneten des Rotors und dem Reedkontakt. Der schaltet aber nur ein und aus, eine „Stromwende“, also Umpolung findet nicht statt. Somit kann der antreibende Elektromagnet auch nur eine Polarität annehmen. Der zweite Grund ist die hohe Ansprechempfindlichkeit des Reedkontakts: Bei 12 Rotormagneten gelingt es kaum, ein sauberes Ein- und Ausschalten zu erzielen.

In [3] hat Matthias Dettmer sehr schön dargelegt, wie und mit welchen Drehzahlen ein Synchronmotor läuft. Bei 12 Rotormagneten mit abwechselnder Polarität (die Polpaarzahl ist dann sechs) und 50 Hz Netzfrequenz wird eine Drehzahl von 500 U/min erzielt.

Statt die volle Netzschnwingung auszunützen, wird mit den einseitig polarisierten Rotormagneten nur eine Halbwelle genutzt. Frequenz und Drehzahl bleiben gleich, das Drehmoment und damit die Ausnutzung werden etwa halbiert (Abb. 5).

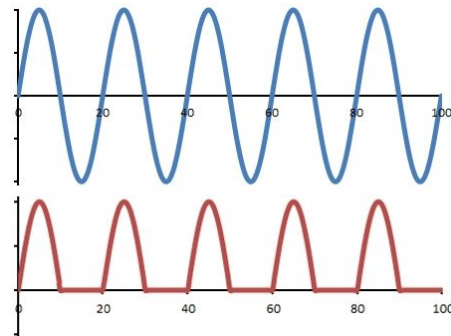


Abb. 5: Oben die volle Netzfrequenz, darunter die positive Halbwelle

Somit kann ich in Tabelle 1 von Matthias Dettmer die Spalte „Polpaare“ ersetzen durch „einseitig ausgerichtete Magnete“.

## Drehzahl

Natürlich möchte man auch die Drehzahl des Motors wissen, wenn möglich mit wenig Aufwand und mit „Bordmitteln“. Die Messung der Impulse vom Reedkontakt brachte sprunghafte Änderungen der Anzeige, hat also nicht funktioniert.

Zur Verfügung hatte ich ein billiges Digital-Multimeter mit Frequenzmessbereich, eine kleine Spule aus einer (mechanisch) defekten Zeitschaltuhr und einen „diametral“ magnetisierten Stabmagneten mit 4 mm Durchmesser und 10 mm Länge.

Den Stabmagnet habe ich mit einer Klemmbuchse 10 am hinteren Wellenende befestigt (Abb. 6). Dann habe ich die Spule auf einen Sockel gesetzt – und die Anzeige fing an zu spinnen: Viel zu hohe Frequenzen von über 170 Hz.

Ursache waren die Vibrationen der Maschine, die sich auf die Spule übertrugen, wodurch das Magnetfeld moduliert wurde.

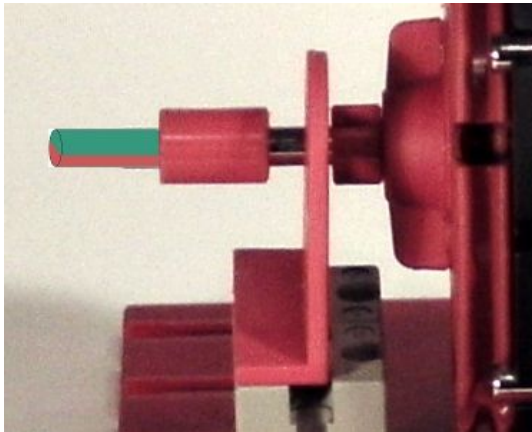


Abb. 6: „Diametral“ magnetisierter Stabmagnet

Abhilfe schaffte eine weiche, gedämpfte Befestigung der Spule wie in Abb. 7.

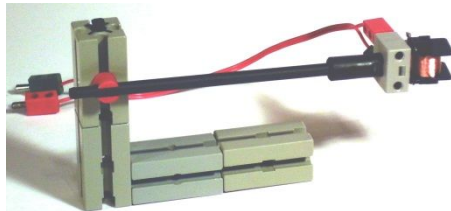


Abb. 7: Halterung der Spule zur Drehzahlmessung

Das Messgerät zeigt nun sauber die Drehzahl in Hz an. Mit 60 multipliziert ergibt das die Drehzahl in U/min.

Die Umrechnung kann man sich zwar mit einer Tabelle erleichtern, aber irgendwie ist das nicht wirklich „elegant“. Wie wäre es daher mit einem ...

## Stroboskop

Die Herstellung ist ganz einfach: Eine LED an Wechselstrom anschließen – fertig. Eine LED ist ja eine Diode und lässt den Strom nur in einer Richtung durch, so dass das Licht der Halbwelle in Abb. 5 folgt. Ein Licht-„Blitz“ ist 10 ms lang, gefolgt von einer 10 ms langen Pause. Mit einem Fotoblitzen kann man das zwar nicht vergleichen, der leuchtet kürzer als 1 ms, aber für unsere Zwecke reicht das aus. Beleuchtet man mit dieser LED das Zahnrad Z30 mit der weißen Markierung aus Abb. 1 und erhöht langsam die Versorgungsspannung, dann erscheinen bei bestimmten Drehzahlen feststehende

Muster. Hat man die Anzahl  $A$  der Magnete gezählt, dann ergibt sich die Drehzahl zu

$$\text{Drehzahl} = (3.000 \text{ U/min}) / A$$

Die Farbunterschiede in Abb. 8 sind durch zwei verschiedene LEDs entstanden; eine andere Beleuchtung war nicht vorhanden. Belichtungszeit lag bei ca.  $\frac{1}{2}$  s.

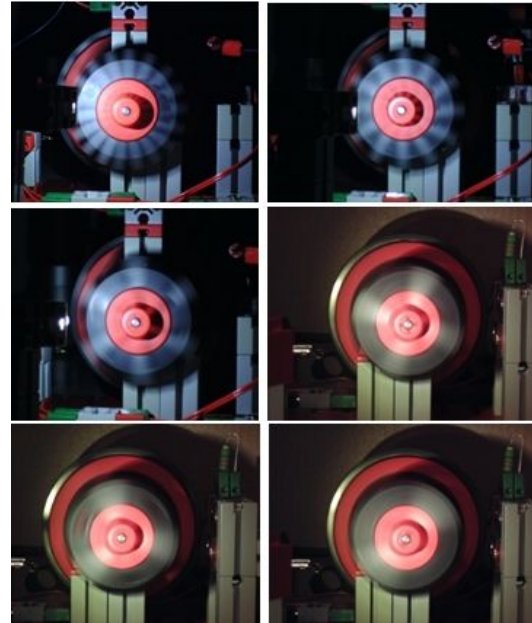


Abb. 8: Stroboskopbilder für die Drehzahlen 167, 250, 500, 750, 1.000 und 1.500 U/min

Die Sperrspannung einer LED ist nicht besonders hoch, deshalb habe ich zum Schutz noch eine Standarddiode (z. B. 1N4001) in Flussrichtung vorgeschaltet.

## Wechselstrombetrieb

Die Erweiterung um den Synchronantrieb leistet der „Powerblock“ rechts in Abb. 9. Der obere Magnet treibt den Gleichstrommotor, der unter den Wechselstrommotor (dass es verschiedene Typen sind, hat keine Bedeutung – die waren einfach gerade verfügbar).

Mit dem linken Schalter in Abb. 10 wird der Gleichstrommotor eingeschaltet, mit dem rechten der Wechselstrom- (Synchron-) motor. Die Stromkreise sind vollständig getrennt.

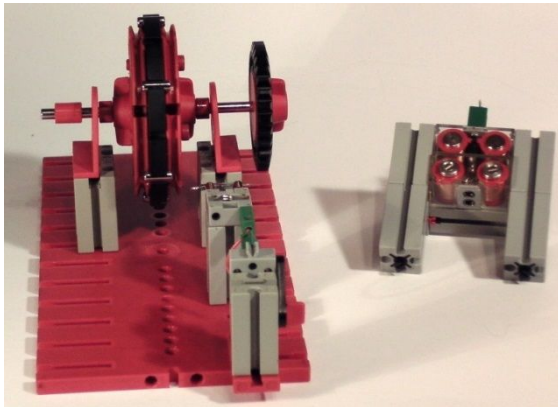


Abb. 9: Rechts der abgenommene Powerblock

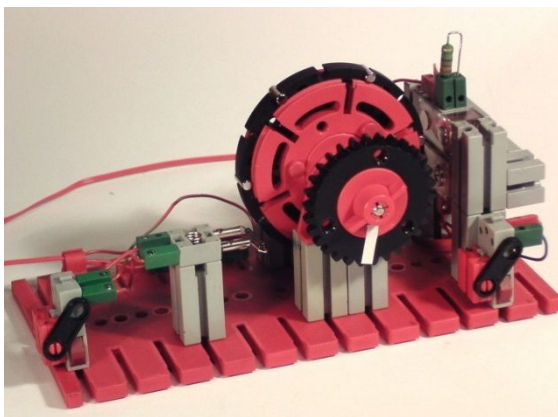


Abb. 10: Die vollständige Kombimaschine

Der Betrieb der „Kombimaschine“ beginnt mit der Zuschaltung des Gleichstroms und leichtem Anstoßen des Rotors. Zum Anlaufen reicht eine Spannung von 5 V aus. Mit der Stroboskop-LED wird das Zahnrad Z30 mit der weißen Markierung beleuchtet.

Dann erhöht man langsam die Versorgungsspannung, bis der sechsarmige Stern in Abb. 8 erscheint (bei mir sind das etwa 6 V). Der Stern sollte still stehen oder langsam im Sinn der Rotordrehung rotieren. Dann wird der Wechselstrom (bei mir ebenfalls 6 V) zugeschaltet. Wenn der Stern nur leicht vor und zurück pendelt, kann der Gleichstrom abgestellt werden, und der Motor läuft synchron weiter.

Der „runde“ Lauf des Gleichstrommotors begeistert mich immer wieder. Dieser wird begünstigt durch das relativ große Schwungmoment und die leichtgängige Lagerung im Lagerbock 38252.

Bei kleinen Drehzahlen kann man das Klicken des Reedkontaktes hören. Der Synchronmotor läuft dagegen eher ruppig.

## Varianten

Mit unserer „Kombimaschine“ lassen sich nun auch Synchronmotorvarianten mit unterschiedlicher Zahl an Permanentmagneten realisieren.

### Läufer mit drei Magneten

Die Hälfte der Magnete führt zur Synchrondrehzahl von 1.000 U/min. Diese habe ich mit dem Gleichstromteil mit 10,3 V und 0,07 A erreicht, die Synchronisierung erfolgte mit 6 V Wechselstrom.

### Läufer mit zwei Magneten

Jetzt braucht der Motor schon ordentlich hohe Spannungen. Mit 22,4 V habe ich 1.500 U/min erreicht bei einem Strom von 0,06 A. Die Synchronisierung erfolgte mit einer Wechselspannung von 14 V, der Synchronmotor lief einwandfrei.



Abb. 11: Läufer mit nur zwei Magneten

### Läufer mit vier Magneten

Mit zwei Drehschalter-Unterteilen 31311 lässt sich die „virtuelle“ Polpaarzahl vier erreichen. Zur Fixierung befindet sich in der Mitte ein Klemmzahnrad 15, je eine Flachnabe vorn und hinten drücken das Ganze zusammen, siehe Abb. 12. Sollte der Läufer klappern, kann man ein Stück Pneumatik-Schlauch um das Zahnrad legen.

Die Synchrondrehzahl von 750 U/min wurde bei 9,7 V Gleichstrom und 0,1 A erreicht, der Synchronmotor lief anschließend mit 6 V Wechselstrom.



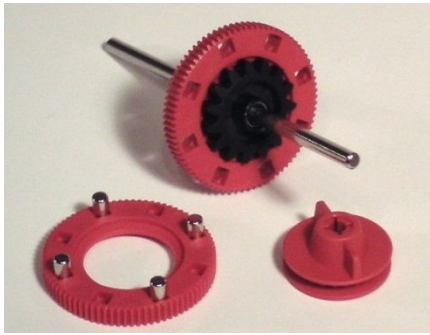


Abb. 12: Die Magnete halten ausreichend gut in den beiden Drehschalter-Unterteilen

### Läufer mit fünf Magneten

Meine Kreation eines Läufers mit fünf Magneten wirkt etwas gezwungen. Die Stabmagnete sind hier 25 mm lang. Zwei Gelenklaschen 36330 halten drei Magnete in den Langlöchern der Drehscheibe 60 auf Abstand. Zwei Gummis vom Spurkranz, einer vorne und einer hinten, halten die beiden anderen Magnete zusammen.

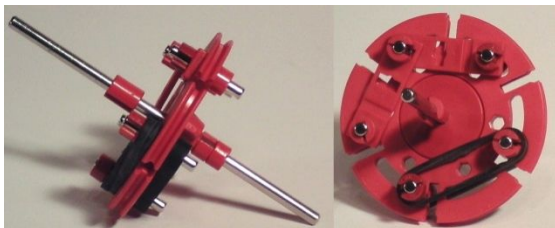


Abb. 13: Der Läufer für 600 U/min

Das ist nicht ganz symmetrisch, aber es läuft recht ordentlich. Um 600 U/min zu erreichen brauchte ich 11,8 V und 0,09 A Gleichstrom. Der Synchronlauf funktioniert gut mit 6 V.



Abb. 14: Synchronlauf mit 600 U/min

### Drehzahl?

Bisher sind wir davon ausgegangen, dass es einen eindeutigen Zusammenhang zwischen der Anzahl der Läufermagnete und der synchronen Drehzahl gibt. Das ist in [3]

schön dargelegt. Nun stelle man sich vor, der Läufer aus Abb. 10 drehe sich mit halber Drehzahl, also 250 U/min. Dann erscheint nur bei jeder zweiten Halbwelle der Netzspannung ein Läufermagnet vor dem Elektromagneten. Der würde aber wie bisher antreiben und wenn das Drehmoment ausreicht, sollte der Motor mit dieser Drehzahl synchron laufen.

Mit dem Läufer mit nur zwei Magneten funktioniert das sehr gut. Ich habe die Spannung langsam erhöht und dabei folgende Drehzahlen mit anschließendem Synchronbetrieb erreicht:

Spannung	Drehzahl
7,0 V	300 U/min
8,6 V	375 U/min
9,2 V	750 U/min
22,4 V	1.500 U/min

Tab. 2: Drehzahlen

Wie oben beschrieben sind 1.500 U/min die primäre Synchrondrehzahl. Weiter habe ich folgende Synchrondrehzahlen unterhalb der primären Synchrondrehzahl erreicht: 167, 250, 333 U/min. Weiß jemand, mit welchen Läufern das möglich ist?

Man könnte diese Betriebsart „Lückbetrieb“ nennen.

### Nochmal: Die Löschiode

Nach all den o. g. Versuchen habe ich die Löschiode nochmal am DC-Motor mit zwei Magneten im Drehschalter ausprobiert. Diesmal zeigte sich kein Unterschied zum Löschwiderstand.

Am empfindlichsten reagiert der Motor auf die Position des Reedkontaktes; sorgfältiges Einstellen kann die Eigenschaften erheblich verbessern. Ich habe den Motor mit der Löschiode hochgejubelt bis 52 Hz, das entspricht einer Drehzahl von mehr als 3.100 U/min – allerdings mit 31,8 V und 0,13 A. Das hat dem Reedkontakt auch nach vielen Minuten Betrieb nichts ausgemacht. Um unliebsamen Überraschungen

vorzubeugen, sollte man zusätzlich zur Löschdiode jedoch eine Schutzdiode  $D_S$  nach Abb. 15 vorsehen.

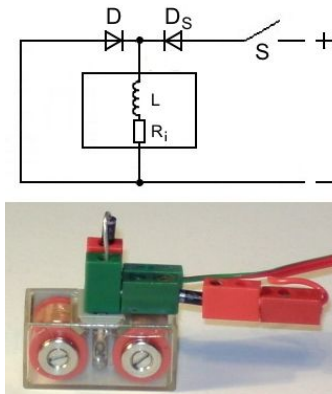


Abb. 15: Lösch- und Schutzdiode; oben das Schaltbild, darunter ein Realisierungsvorschlag

In Abb. 16 ist der Motorlauf mit 3.000 U/min mit der Stroboskop-LED festgehalten.

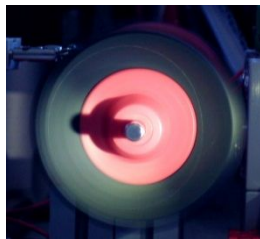


Abb. 16: Motor dreht mit 3.000 U/min

Wäre die Spannung an der LED rechteckförmig mit 10 ms Dauer, dann müsste die Markierung auf Bild 16 über den halben Umfang verschmiert sein. Die Drehzahl von 3.000 U/min entspricht ja 50 U/s, also dauert eine Umdrehung 20 ms. Wegen des Sinusverlaufs der Spannung kann man aber Anfang und Ende nicht erkennen, deshalb erscheint der helle Sektor nur über etwa einem Drittel des Umfangs.

### „Forschungsbedarf“

Wie bekommt man den kombinierten Gleichstrom- und Wechselstrommotor auf 3.000 U/min mit anschließendem Synchronlauf?

Der Betrieb des Drehschalter-Läufers mit nur einem Magnet ist möglich, indem der Kommutierungsmagnet separat mitläuft.

Die Motoreigenschaften sind aber ziemlich schlecht, ein sauberer Rundlauf ist kaum zu erreichen. Drehzahlen von 750 und 1.000 U/min habe ich so mit anschließendem Synchronlauf erreicht. Die primäre Synchrondrehzahl von 3.000 U/min ist damit aber wohl nicht erreichbar.

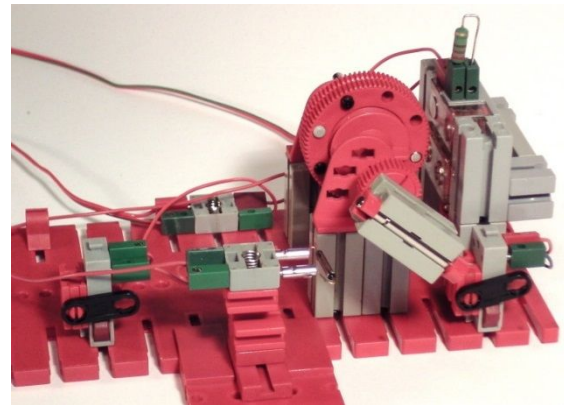


Abb. 17: Nur ein Magnet im Läufer

Die drei Stifte im Läufer links von Abb. 17 dienen dem Massenausgleich; im Baustein 30 vorne befindet sich der Kommutierungsmagnet und eine Achse 30.

### Ein Dankeschön

Mein ausdrücklicher Dank gilt Stefan Roth (ffm) und Andreas Schallmaier (fitec), ohne die ich nicht das Material hätte, meinem ft-Hobby zu frönen.

### Referenzen

- [1] Thomas Habig: *LEDs mit Vorwiderstand*. [ft:pedia 2/2011](#), S. 14-18.
- [2] Dirk Fox: *Der Elektromotor*. [ft:pedia 3/2013](#), S. 4-8.
- [3] Matthias Dettmer: *Synchronmotoren*. [ft:pedia 2/2016](#), S. 48-52.
- [4] Stefan Falk: *Perlentauchen (Teil 3)*. [ft:pedia 1/2013](#), S. 22-31.
- [5] fischertechnik: *Clubnachrichten 13*. März 1972, S. 22-23.
- [6] Dirk Fox, Thomas Püttmann: *Technikgeschichte mit fischertechnik*. dpunkt.verlag, Heidelberg, 2015.

Computing

## Auf zu neuen Ufern: Die Geschichte der „Community-Firmware“ für den TXT

Till Harbaum

*Mit dem TXT hat fischertechnik vor gut zwei Jahren seinen bisher komplexesten Robotics-Controller auf den Markt gebracht und auch den Wettbewerb aus Billund in die Schranken verwiesen. Mit seiner Hardwareausstattung entspricht er ungefähr einem Arbeitsplatz-PC aus dem Jahre 2000 bzw. dem ersten Raspberry-Pi-Modell von 2011. Damit ist der TXT zwar 2016 technisch auch im Vergleich zu anderen Embedded-Plattformen nicht mehr ganz aktuell. Aber mehr als das, was die mitgelieferte Software aus ihm herausholt, ist auf jeden Fall machbar. Die fischertechnik-Community verspricht daher mit der Bereitstellung einer unabhängigen Community-Firmware eine frische Brise in den Segeln des inzwischen leicht in die Jahre gekommenen TXTs.*

### Doppelherz

Der TXT besteht tief in seinem Innersten eigentlich aus zwei getrennten Computern, die, wie man auch ohne den eigenen TXT zu öffnen z. B. unter [1] sehen kann, sogar jeweils auf einer eigenen Platine untergebracht sind. Einer dieser Computer, der Linux-Controller, entspricht mit seiner 600-MHz-CPU nebst 256 MB RAM der ersten Generation des bereits erwähnten Raspberry-Pi und läuft wie jener unter Linux. Der zweite, im folgenden Motor-Controller genannte Computer ist sehr viel einfacher gestrickt und verfügt nur über gut 10 % der CPU-Leistung sowie wenige Kilobyte RAM. Die Arbeitsteilung zwischen beiden Teilen entspricht der eines PCs und eines früheren fischertechnik-Robo-Interfaces im Online-Betrieb. Während sich der Linux-Controller wie damals der PC um die Kommunikation mit dem Anwender kümmert, ist es die Aufgabe des Motor-Controllers, die fischertechnik-spezifischen Ein- und Ausgänge zeitnah zu bedienen, genau wie es das Robo-Interface damals auch tat.



Abb. 1: Die gestartete Community-Firmware

Beide Controller des TXT verfügen über eine eigene fest verbaute Software, die sogenannte Firmware. Im Folgenden wird es ausschließlich um die Firmware des Linux-Controllers gehen. Den Motor-Controller lässt die Community-Firmware zurzeit genauso unangetastet, wie es auch die regulären ROBO Pro-Updates machen, die ebenfalls nur den Code des Linux-Controllers aktualisieren.

## Es war einmal ...

Es ist kein Geheimnis, dass der TXT wie schon sein Vorgänger der TX etwas unfertig auf den Markt gebracht wurde. Trotz einer deutlichen verspäteten Markteinführung sind nach wie vor nicht alle beworbenen Funktionen in der offiziellen Firmware umgesetzt. Während die Kunden bei den früheren Controllern wenig mehr tun konnten, als ihren Unmut zu äußern, so sieht die Situation beim Linux-Controller des TXT fundamental anders aus. Mit dem Einsatz des bekannten Open-Source-Betriebssystems erhält fischertechnik zwar auf der einen Seite kostenlosen Zugriff auf Millionen von Codezeilen, sie gehen als Gegenleistung aber die Verpflichtung ein, den TXT-Kunden ihrerseits vollen Zugriff auf alles zu geben, was sie aus dieser Code-Spende abgeleitet haben. Dieser Vorgang wird oft als der „virale Aspekt“ der Gnu Public License [2] angesehen, also der Lizenz, unter der u. a. der Kern des Linux-Betriebssystems weitergegeben wird. Nur wenige Firmen sind sich dieser Konsequenzen sofort bewusst, und offensichtlich war auch fischertechnik zunächst nicht ganz klar, was ihre Verpflichtungen im Rahmen der Linux-Nutzung anging, denn es dauerte ein gutes Jahr, bis die Kollegen von fischertechnik der Lizenz entsprechend den Code des TXTs ihrerseits offenlegten.

Damit hatten sie der Lizenz weitgehend genüge getan. Sehr viel gewonnen hatte die Community aber noch nicht, denn diese mehrere hundert Megabyte große Code-Sammlung zurück in etwas zu verwandeln, das man auf dem TXT nutzen kann, erfordert die Überwindung einer ganzen Anzahl von Hürden.

## Backe, backe Kuchen

Zunächst muss diese riesige Quellcode-Menge in sogenannte Binärform gebracht werden. Den veröffentlichten Code kann man sich wie ein viele Millionen Zeilen langes Kuchenrezept vorstellen. Erst wenn

man alle Werkzeuge zusammen hat, sie bedienen kann und ausreichend Erfahrung als Bäcker mitbringt, kann man daraus letztlich wieder genau den richtigen Kuchen in Form der Firmware backen. Natürlich könnte man weiterhin auf den fertigen Kuchen, also die von fischertechnik als fertige Updates verteilte Firmware zurückgreifen. Aber ohne das Rezept und ohne das nötige Handwerkszeug und Know-How muss man mit dem fertig verkauften Kuchen leben und kann keine Veränderungen an der Rezeptur vornehmen. Genau darum ging es der Community aber. Wie bereits gesagt, ist der TXT technisch zu viel mehr in der Lage, als zu dem, was die Original-Firmware nutzt. Erst die Möglichkeit, eigene Rezeptideen umzusetzen, eröffnet die Möglichkeit, aus der mitgelieferten Napfkuchen-Firmware des TXT eine schmackhafte Sahnetorte zu machen.

fischertechnik selbst bzw. ihre Zulieferer haben sich bei diversen öffentlich zugänglichen Rezeptquellen bedient. Und selbst wenn fischertechnik selbst nicht viel dazu sagen mag (und es auch nicht muss), wie man seine Rezepte zu lesen hat, so gibt es ausreichend Dokumentation bei der ursprünglichen Quelle, dem sogenannten Buildroot-Projekt [3]. Tatsächlich hat sich gar nicht fischertechnik selbst beim Buildroot-Projekt bedient, sondern es war der Mikroprozessor-Hersteller Texas Instruments. Dieser hat für seinen TI-Sitara-Prozessor ein Linux-Paket auf Basis des Buildroot-Projektes geschnürt. Das hat sich wiederum der Zulieferer von fischertechnik genommen, um den TXT auf Basis des Sitara zu bauen und mit Software zu versehen. So ging der Code durch viele Hände, bis er schließlich über diverse Umwege auf der Download-Seite von fischertechnik und in den Händen der Community landete. Vieles war auf dem langen Weg an Dokumentation und Wissen verloren gegangen, manche Dinge waren etwas unbeholfen verändert worden, vor allem war das alles aber deutlich veraltet

seit der Abzweigung aus dem Original-Buildroot-Projekt Mitte 2014. Das Buildroot-Projekt hat sich seitdem weiterentwickelt, und die gesamte Software war inzwischen auf einem Stand weit jenseits dessen, was fischertechnik ausliefert. Mit viel Linux-Know-How wurden die Fragmente von der Community wieder verbunden, aktualisiert und in geordneter Form unter [4] abgelegt. Damit war der Grundstein für eine eigene Firmware gelegt. Allerdings galt es, eine zweite große Hürde auf diesem Weg zu nehmen.

Diese zweite große Hürde besteht darin, den TXT mit dem neuen, eigenen Firmware-Code zu befüllen. Die Community-Firmware wie ein Originalupdate durch eine eigens modifizierte ROBO Pro-Variante aufzuspielen, wäre ohne Know-How der ROBO Pro-Interna nicht ganz einfach gewesen. Vor allem aber hätte man die Funktion der Originalfirmware verloren, was gerade in der Anfangszeit ein Problem gewesen wäre, da die Community-Firmware zunächst u. a. mit den fischertechnik-Anschlüssen des TXT nichts anfangen konnte. Aber auch heute noch würden die meisten Benutzer vor der Community-Firmware verständlicherweise zurückschrecken, wenn sie dafür die gewohnte und erprobte Funktion der Originalfirmware auch nur zu einem kleinen Teil verlieren würden. Zum Glück verfügt der TXT über einen ungenutzten SD-Karten-Slot. Leider ignoriert der TXT den Slot im Auslieferungszustand, so dass auch hier zunächst kein Zugang für eigene Entwicklungen war.

Das änderte sich schlagartig Anfang 2016, als im Community-Forum der Tipp auftauchte, dass auf den Pins des EXT-Erweiterungssteckers die sogenannte „serielle Konsole“ verfügbar sei. Eine solche Konsole bildet einen Wartungs- und Kommandozugang über eine übliche serielle Schnittstelle. In den Anfangstagen der IT-Entwicklung verfügten Computer generell über keinen eigenen Bildschirm

und keine Tastatur, so dass Benutzer-eingaben standardmäßig über ein an die Konsole angeschlossenes sogenanntes Terminal erfolgte, dass seinerseits nichts großartig Anderes war als eben ein Bildschirm mit Tastatur und einer seriellen Schnittstelle. Viele eingebettete Geräte verfügen nach wie vor über diesen Konsolenanschluss, wenn sie über keinen eigenen Bildschirm und keine Tastatur verfügen. So auch der TXT, wie sich nun also herausstellte. Die am Erweiterungsstecker anliegenden Spannungen sind nicht zu den üblichen RS232-Schnittstellen kompatibel, so dass man einen Pegelwandler zwischen PC und TXT schalten muss, wenn man den TXT nicht beschädigen will. Oder man verwendet gleich einen passenden USB-Adapter zum Anschluss an den PC.

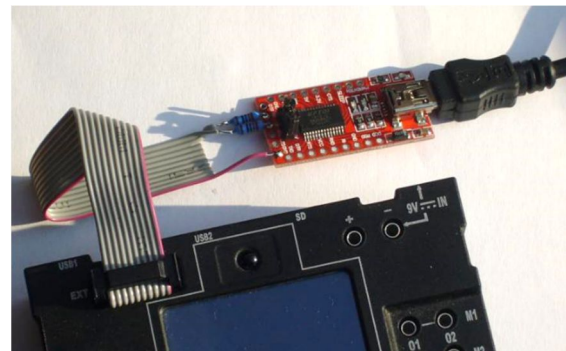


Abb. 2: Ein USB-nach-Seriell-Adapter

Die sogenannten „USB-TTL“-Konverter sind für wenig Geld online erhältlich; man sollte aber darauf achten, eine 3,3 Volt-Version zu erstehen. Die ebenfalls gängigen 5 Volt-Adapter können dem TXT schaden.

Nachdem dieser Adapter angeschlossen war erfüllte sich nicht nur die Hoffnung auf einen direkten Kommando-Zugang zum Linux des TXT, sondern es meldete sich dort direkt nach dem Einschalten des TXT der sogenannte Bootloader, also das Stück Software, das ähnlich dem BIOS des PCs den eigentlichen Startvorgang steuert. Beim Bootloader des TXT handelt es sich um ein Programm namens UBoot [5]. Dieser Bootloader ist weit verbreitet und kommt oft bei vergleichbaren Systemen zum

Einsatz. Vor allem besitzt er die Fähigkeit, das System von allen möglichen Medien zu laden und zu starten, seien es USB-Sticks, interner Flash-Speicher, Netzwerkadapter oder eben SD-Karten. Schnell zeigte sich, dass der im TXT verbaute Bootloader nach wie vor auch von SD-Karte booten konnte, wenn man direkt nach dem Einschalten des TXT über den Adapter vom PC die richtigen Kommandos sendete.

Damit waren alle Grundlagen für die Community-Firmware vorhanden. Die Community hatte den gesamten Code in der Hand und eine Möglichkeit, den TXT ein neues Betriebssystem von SD-Karte booten zu lassen. Es dauerte nur wenige Tage und die ersten Erfolgsmeldungen schlugen im Forum auf. An diesem Punkt war klar, dass es keine fundamentalen technischen Hürden mehr gab, sondern dass es nur noch an Fleiß und Arbeitseinsatz fehlte, um dem TXT ein frisches Innenleben zu verpassen.

Die Tatsache, dass nun eine SD-Karte zum Einsatz kam, hatte zwei sehr angenehme Nebeneffekte: Zum einen kann man seinen TXT jederzeit in den Ursprungszustand zurückversetzen, einfach indem man die SD-Karte mit der Community-Firmware darauf aus dem TXT entfernt. Zum anderen ist der Speicherplatz auf der SD-Karte wesentlich größer als der interne 128MB-Flash-Speicher des TXT, auf dem sich die Original-Firmware nach wie vor befindet.

Kleine Anekdote am Rande: Dieser interne Flashspeicher ist 128 MB groß und nicht, wie von fischertechnik beworben, nur 64 MB. Den gleichen Fehler machen die offiziellen Unterlagen auch beim RAM-Speicher des TXT, der 256 MB umfasst und nicht nur die beworbenen 128 MB.

## Puzzlesteinchen

Ein Linux macht aber noch lange keinen TXT und die Community hatte noch einigen Weg vor sich, um von den ersten Bootmeldungen eines eigenen Linux-Kernels zu einer Firmware zu kommen, die auch für

nicht-Hacker eine erkennbare Funktion hat und sich eben nicht nur über die serielle Kommando-Konsole bedienen lässt. Schnell teilten sich die Rollen unter den beteiligten Community-Mitgliedern auf, und so gab es Leute, die in der Folge die diversen Standard-Linux-Komponenten an den TXT anpassten und die Hardware des TXT Schritt für Schritt in Betrieb nahmen. Parallel entwickelten andere Mitglieder eine neue Benutzeroberfläche für den Touchscreen des TXT.

Dass dabei überhaupt wieder relativ viel Arbeit anfiel, hat zwei Gründe. Zum einen war die Originalfirmware und ihr zugrundeliegender Code wie erwähnt deutlich in die Jahre gekommen. Da die Community-Firmware sofort mit aktuellen Komponenten startete, musste an vielen Stellen jede inzwischen weiter entwickelte Komponente erneut an den TXT angepasst werden. Vor allem der Linux-Kernel, der u. a. die gesamte Hardware des Systems verwaltet, hatte einige Versionsprünge seit der Veröffentlichung des TXT hinter sich, und viele Dinge funktionierten inzwischen einfach etwas anders. Der zweite Grund ist die Tatsache, dass fischertechnik nur einen Teil des Codes des TXT freigegeben hatte, und zwar nur genau den Teil, zu dem sie per Lizenz verpflichtet waren. Die Eigenentwicklungen, zu denen u. a. die gesamte Benutzeroberfläche sowie die Anbindung des Motor-Controllers gehören, hielt fischertechnik zunächst weiter unter Verschluss. Im Laufe der Zeit wurden auch hier ein paar Unterlagen von fischertechnik nachgeliefert, ganz komplett und vor allem aktuell waren diese Dokumente aber bisher nicht, so dass die Community sich nach wie vor hier und da mit Tricks behelfen musste oder gleich selbst mit einer Neuentwicklung aktiv wurde.

## Trick der Motor-Controller-Anbindung

Die größte Lücke bildete lange Zeit die Anbindung des Motor-Controllers im TXT.

Ohne die nötigen Unterlagen war es zunächst nicht möglich, aus der Community-Firmware auf die fischertechnik-Anschlüsse des TXT zuzugreifen. Der TXT war damit seiner eigentlichen Hauptfunktion beraubt, was zu der kuriosen Situation führte, dass der TXT unter der Community-Firmware über einen per USB angeschlossenen Lego-WeDo-Hub Lego-Motoren steuerte, noch bevor er einen fischertechnik-Motor ansprechen konnte (Abb. 3).



Abb. 3: Lego-Motor am TXT

Der Trick, der diese Situation etwas entspannte, ergab sich aus der Erkenntnis, dass der gesamte Code der Originalfirmware ja noch im internen Flash-Speicher des TXT vorhanden war. Hier lag und liegt nach wie vor das Programm, das in einem TXT im Originalzustand die gesamte Kommunikation mit dem Motor-Controller übernimmt und z. B. über WLAN mit dem ROBO Pro-Programm auf dem PC kommuniziert. Ein paar Kniffe waren nötig, um dieses eigentlich veraltete Programm unter der massiv aktualisierten Community-Firmware zum Laufen zu bringen und es zu überzeugen, sich z. B. beim Zugriff auf den Bildschirm des TXT mit der Bedienoberfläche der

Community-Firmware zu arrangieren. Aber letztlich gelang es, diesen Teil der Original-Firmware wieder so zum Laufen zu bringen, dass auch Programme unter der Community-Firmware Zugriff auf die fischertechnik-Anschlüsse des TXT erhalten.

## Aufgehübscht

Parallel zu diesen sehr technischen Aspekten wurde an der Benutzeroberfläche gefeilt. Die Originalfirmware setzt an dieser Stelle auf das „Simple DirectMedia Layer“-Framework, kurz SDL [6]. Dieses Framework hat eigentlich nur genau einen Vorteil: Es ist klein und simpel. Als Basis für eine moderne Benutzerschnittstelle eignet es sich aber nur sehr eingeschränkt. Vor allem muss sich der Entwickler um sehr viele Dinge selbst kümmern, die ihm moderne, sogenannte GUI-Frameworks abnehmen. Ein Grund für die Verwendung von SDL in der Originalfirmware wird sicher auch der begrenzte Speicherplatz im TXT-internen Flash-Speicher gewesen sein. Da die Community-Firmware von vornherein darauf ausgelegt war, von SD-Karten zu laufen, existiert diese Beschränkung dort nicht, so dass komplexere und speicherintensivere Frameworks wie GTK [7] und Qt [8] zum Einsatz kommen können. Linux-Benutzer kennen diese beiden Frameworks in der Regel, da sie die Basis der populären Gnome- und KDE-Desktopumgebungen darstellen.

Die Wahl fiel auf Qt, da es bereits im Buildroot-System vorhanden war und sich leicht aktivieren ließ. Entscheidend war aber die Tatsache, dass QT keinen so genannten Window-Manager benötigt, sondern auf kleinen Systemen wie dem TXT ohne komplexe Software-Infrastruktur direkt auf den eingebauten TFT-Bildschirm zugreifen kann. Damit war Qt auf dem TXT mit wenigen Handgriffen einsatzbereit. Bei der verwendeten Programmiersprache orientierten sich die Entwickler der Community-Firmware an dem, was sich

bereits im Vorfeld etabliert hatte. Es gab bereits diverse in Python geschriebene Projekte für den TXT und mit fropopy [9] auch bereits Python-Unterstützung für die fischertechnik-Ein- und Ausgänge des TXT, so dass die Wahl auf Python fiel.

Als Beispiel für die möglichen Benutzeroberflächen sei die Gegenüberstellung der Flipper-Steuerung im Original und in der Community-Firmware-Fassung gezeigt (Abb. 4 und 5).

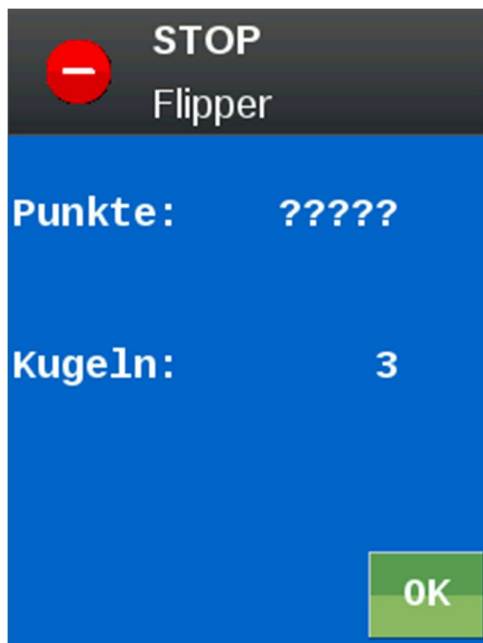


Abb. 4: Die originale TXT-Flipper-App

Das QT-Framework bietet die Möglichkeit, den Stil der grafischen Darstellung in weiten Grenzen zu verändern. Die Farben, Größen und Schriften die z. B. für „Buttons“ Verwendung finden, lassen sich sehr einfach in Konfigurationsdateien festlegen und werden automatisch immer dann angewendet, wenn eine Anwendung einen Button auf dem Bildschirm darstellen will. Ähnliches gilt auch für alle anderen Grafikelemente, aus denen eine moderne Benutzeroberfläche besteht. Der Anwendungsprogrammierer muss sich am Ende um diese Details überhaupt nicht mehr kümmern. Anders als z. B. unter SDL übernimmt das Qt-Framework die komplette Darstellung der Oberfläche. Der

Programmierer kann sich auf das „was“ konzentrieren und kann das „wie“ dem Framework überlassen.

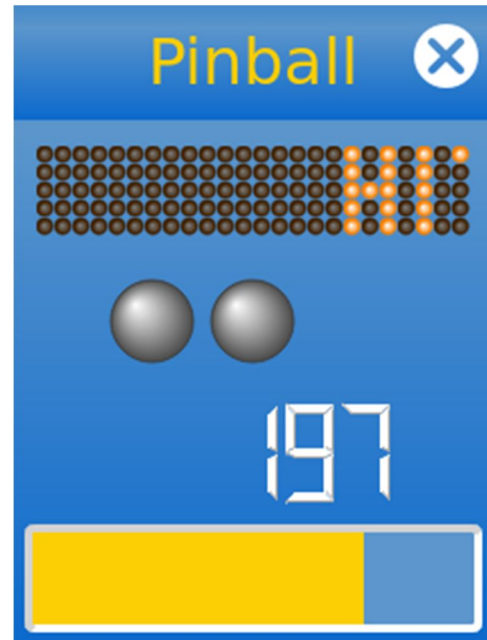


Abb. 5: Die Flipper-App der Community

Als weiteres Beispiel zeigen Abb. 6 und 7 einen Taschenrechner, wie er auf einem PC aussieht, und wie derselbe Programmcode mit nur minimalen Änderungen auf einem TXT gestartet wirkt.

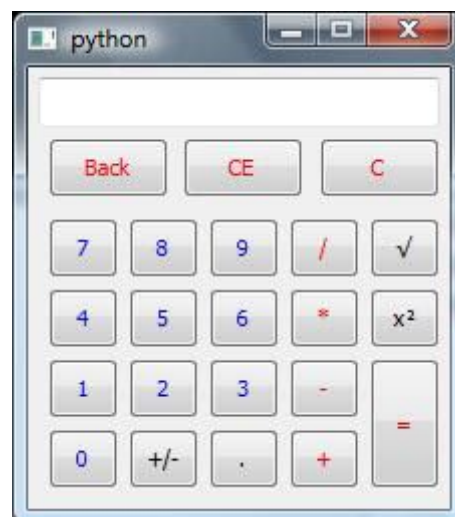


Abb. 6: Calculator-App auf einem PC





Abb. 7: Calculator-App auf dem TXT

## Wie das Handy

Mit dem leistungsfähigen Qt-Framework im Rücken stellte sich die Frage, wie denn die neue Benutzeroberfläche generell aufgebaut sein soll. Hier haben es sich die Community-Entwickler einfach gemacht und sich großzügig bei den Konzepten der üblichen Smartphone-Oberflächen bedient. Große Icons symbolisieren auf dem Startbildschirm die einzelnen Anwendungen (Apps). Durch einen Klick auf das zugehörige Icon startet man eine App. Ebenfalls bekannt vom Smartphone ist der „App-Store“, der es der Community-Firmware erlaubt, weitere Programme aus dem Internet mit wenigen Klicks direkt auf den TXT zu übertragen. Es ist kein PC und kein ROBO Pro nötig, so dass der TXT mit der Community-Firmware in Zeiten aussterbender Heim-PCs lernt, auf eigenen Beinen zu stehen. Die ROBO Pro-Funktion ist natürlich nach wie vor vorhanden, und selbst die Original-Benutzeroberfläche versteckt sich hinter einem der Icons der Community-Firmware. Man kann jederzeit zur alten Oberfläche zurückkehren.



Abb. 8: Der „Launcher“ der Community-Firmware

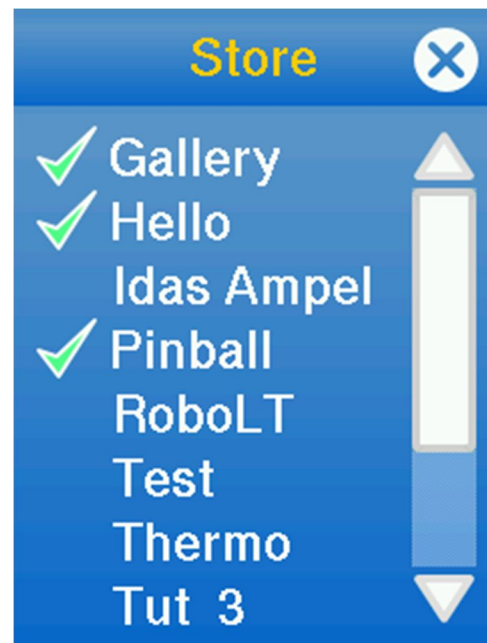


Abb. 9: Community-Firmware-„App Store“

Ziel der Community-Firmware ist es aber, die Original-Firmware und deren Benutzeroberfläche mittelfristig komplett abzulösen und sämtliche Funktionen der Original-firmware inkl. der ROBO Pro-Anbindung besser, schneller und moderner selbst zur Verfügung zu stellen.

## Online-Controller

Generell mutiert der TXT mit der Community-Firmware zu einem „online“-Gerät. Aktuelle Entwicklungen konzentrieren sich auf Umgebungen wie Blockly [10] und Jupyter [11]. Beide Umgebungen laufen auf dem TXT und erlauben die Programmierung des TXT von jedem Web-Browser aus. Damit muss auf dem PC nichts installiert werden, und es eignen sich neben PCs auch Tablets und Smartphones zur Programmierung, sowie nicht-Windows-PCs. Blockly erlaubt dabei eine mit ROBO Pro vergleichbare grafische Programmierung und eignet sich bereits für Grundschulkinder, während IPython-Notebook eher an Matlab erinnert und eine fortgeschrittene Text-basierte Programmierung in Dokumentform ermöglicht.

## Risikominimierung

Um das Benutzen der Community-Firmware auch ohne den eingangs erwähnten UART-Adapter zu ermöglichen, war bis Oktober 2016 eine nicht ganz risikolose Software-Modifikation am TXT nötig. Das permanente Ändern des Bootloaders, um den SD-Karten-Boot dauerhaft zu aktivieren, erforderte zwar nur wenige Kommandos, ein Tippfehler an der falschen Stelle konnte aber aus dem TXT einen Servicefall machen. fischertechnik hat dankenswerter Weise die wenigen bekannt gewordenen Fälle dieser Art kulant repariert.

Viel wichtiger ist aber, dass fischertechnik mit dem ROBO Pro-Update auf Version 4.2.4 und der damit einhergehenden Aktualisierung der Original-TXT-Firmware im Oktober 2016 einen risikolosen Weg geschaffen hat, die für den Betrieb der Community-Firmware nötigen Änderungen am TXT ohne Risiko vorzunehmen. Ein von fischertechnik mit dem Update ausgeliefertes PDF-Dokument [12] beschreibt die nötigen Handgriffe. Damit ist mit der Benutzung der Community-Firmware keinerlei Risiko für den TXT mehr

verbunden. Zusammen mit den von der Community bereit gestellten sog. Image-Dateien ist es damit in wenigen Minuten möglich, den TXT auf die Community-Firmware umzustellen [13]. Vor allem kann man durch Entfernen der SD-Karte aus dem TXT den TXT mit einem einzigen Handgriff in seinen Originalzustand zurück zu versetzen. Die Community-Firmware ist damit zu einer Option für jedermann geworden.

## Wunschliste

Nach wie vor gibt es in der Community-Firmware Stellen, die nicht ganz ausgereift sind. Dies sind genau die Stellen, an denen der Community die Dokumentation fehlt. Konkret betrifft dies primär sämtliche Aspekte der Kommunikation mit dem Motorcontroller sowie die PC/ROBO Pro-Anbindung. An vielen Stellen ist durch Tricks und Hacks bereits eine Grundfunktion hergestellt. Es ist aber der ausdrückliche Wunsch der Community-Entwickler, auf Basis passender Dokumentation technisch befriedigende und für den Endanwender leicht verwendbare Lösungen anzubieten.

## Wiedervereinigung

Die offizielle Firmware und die Community-Firmware gehen nach wie vor getrennte Wege. Das bedeutet an vielen Stellen, dass Arbeit doppelt gemacht wird. Die Community stellt ihre Arbeit auf Github [4] jedem zur freien Verfügung und jeder ist eingeladen, sich an der Weiterentwicklung zu beteiligen. Dies schließt explizit auch die Firma fischertechnik, ihre Mitarbeiter und ihre Zulieferer ein.

Die Hoffnung von uns Entwicklern ist, dass auf lange Sicht nur noch eine Firmware für den TXT entwickelt wird, die die Bedürfnisse eines Einsteigers genauso erfüllt wie die eines IT-Profis. Das ist aber nur durch eine Zusammenarbeit aller Beteiligten möglich.

## The day after TXT

Die Zeiten, in denen sich ein IT-Produkt allein aus seinen Hardware-Spezifikationen definiert hat, sind vorbei. Vielmehr verkaufen sich Geräte inzwischen über Software-Funktionen. Damit stellt die Community-Firmware bereits einen möglichen ersten großen Schritt zu einem TXT-Nachfolger dar.

Dass die Benutzeroberfläche der Community-Firmware mit ihren Apps problemlos den Sprung auf z. B. den Raspberry-Pi geschafft hat [14], zeigt, wie sehr die konkrete Hardware immer weiter in den Hintergrund rückt. Solange die Hardware der nächsten fischertechnik-Controller-Generation weiterhin unter Linux läuft, wird auch die Community-Firmware ihren Weg auf das neue Gerät finden.



Abb. 10: Community-Firmware auf einem Raspberry Pi

## Quellen

- [1] Boge, Dr.-Ing. Christian: [fischertechnik "Robo TXT Controller"-Hardware](#), 2015.
- [2] Free Software Foundation: [GNU General Public License](#), 2007.
- [3] The Buildroot developers: [Buildroot](#)
- [4] fischertechnik Community: [fischertechnik ROBOTICS TXT firmware, community edition](#)
- [5] DENX Software Engineering: [U-Boot](#)
- [6] [Simple DirectMedia Layer](#)
- [7] The GTK+ Team: [The GTK+ Project](#)
- [8] The Qt Company: [Qt](#)
- [9] [fischertechnik TXT phyton \(ftrobopy\)](#)
- [10] Google Developers: [Blockly](#)
- [11] Project Jupyter: [jupyter](#)
- [12] fischertechnik: [Freischaltung des Bootloaders](#)
- [13] fischertechnik Community: [Anleitung: Einrichtung der ftcommunity TXT Firmware](#)
- [14] Dr. Till Harbaum: [Raspi mit Touch](#). heise Verlag, 2016.

Computing

# Frische Apps für den TXT-Controller

Till Harbaum

*Die Community-Firmware an sich mag zwar nett anzusehen sein und ein moderneres Fundament haben als das Original. Das an sich stellt aber zunächst nur einen geringen Mehrwert dar. Wesentlich interessanter für den Endanwender sind die Möglichkeiten, die sich durch die Community-Firmware für die Steuerung von Modellen ergeben. Die Stärke der Firmware liegt dabei in dem schier unerschöpflichen Pool von fertigen Bibliotheken und Frameworks aus der Linux-Welt.*

## Rubiks-Cube-Solver

Mein ganz persönlicher Wunsch war es beispielsweise, einen autonomen Zauberwürfel- bzw. Rubiks-Cube-Lösungs-Roboter zu bauen. Der TXT bietet mit seiner Kamera die perfekte Basis, den Zustand des Würfels automatisch und

schnell zu erkennen. Für die nötige Bilderkennung gibt es unter Linux etablierte Bibliotheken wie die des OpenCV-Projektes [1]. Diese Bibliothek bringt bereits eine Python-Anbindung mit und eignet sich daher ausgezeichnet für die in der Community-Firmware bevorzugt genutzte Python-Programmierung.

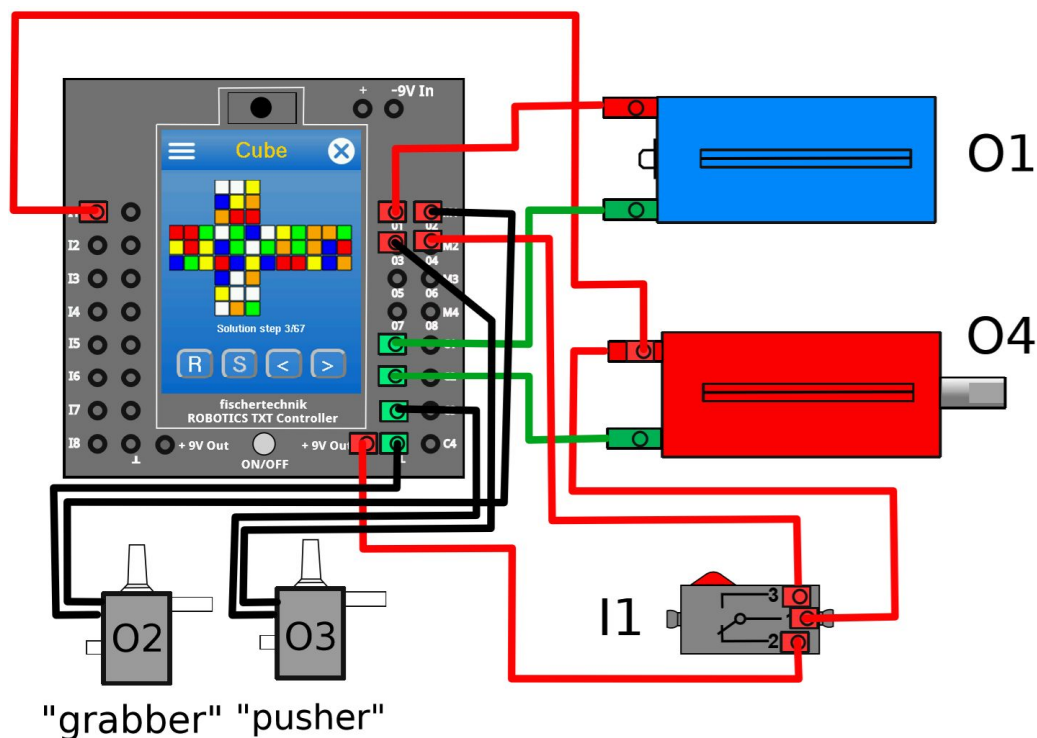


Abb. 1: Verdrahtungsschema des Cube Solvers

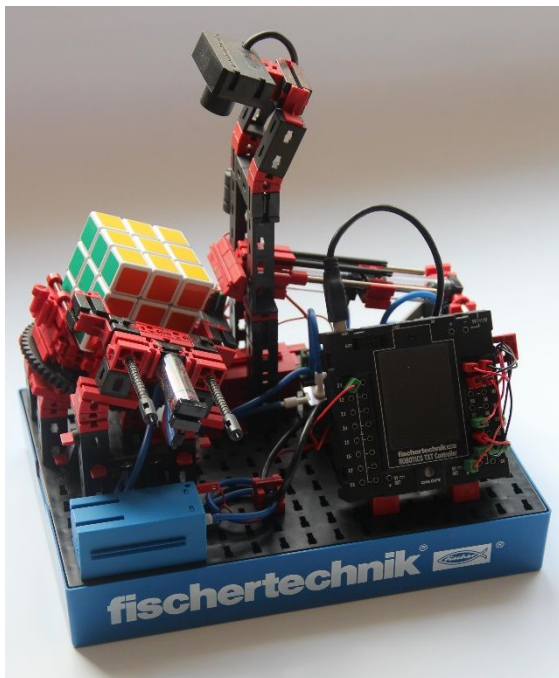


Abb. 2: Gesamtansicht des Cube Solvers

Auch für den mathematisch sehr anspruchsvollen Teil der Lösungsberechnung finden sich im Internet Code-Beispiele. Zunächst haben wir auch hier eine Python-Bibliothek nehmen wollen, die allerdings mit mehrminütigen Rechenzeiten keine Begeisterung auslösen konnte. Ein in der sehr viel maschinennäheren Sprache C geschriebenes Programm ließ sich aber dank kompletter Entwicklungsumgebung der Community-Firmware problemlos für den TXT übersetzen und von der Python-Oberfläche aus starten, so dass der TXT schließlich in ca. 2 Sekunden eine nahezu optimale Lösungsstrategie berechnet.

Der Bilderkennung und Lösungsberechnung zur Seite stehen die eigentliche Ansteuerung des Modells sowie eine ansprechende Benutzeroberfläche. Letztere (Abb. 3) zeigt während der gesamten Laufzeit die aktuelle interne Sicht auf den Zauberwürfel an und der Benutzer kann verfolgen, wie der TXT selbst – ohne den Würfel selbst ständig neu einzulesen – seine interne Darstellung aktualisiert und in Einklang mit dem tatsächlichen Würfel hält.



Abb. 3: Benutzeroberfläche

Während der Entwicklung stellte sich schnell heraus, dass eine der größten Hürden die TXT-Kamera ist. Die Kamera führt eigene Farbkorrekturen und einen automatischen Weißabgleich aus, weshalb die erkannten Farben nicht sehr stabil sind. Daher wurde eine Möglichkeit nachgerüstet, im Programm eine Anpassung der Farberkennung vorzunehmen:

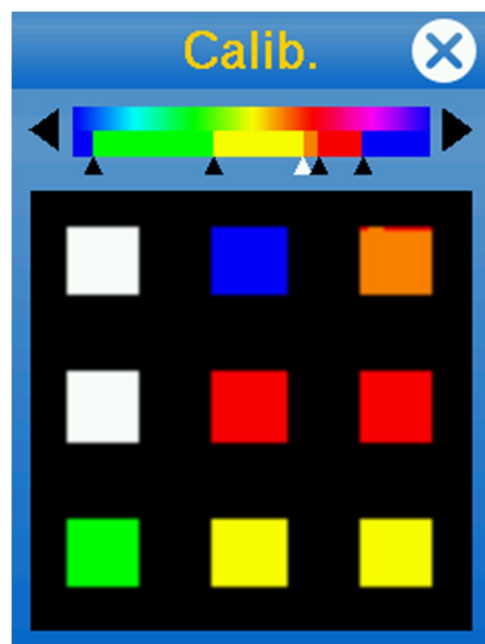


Abb. 4: Kamera-Kalibrierung

Das Farbdisplay des TXT erlaubt dabei eine direkte Vorschau am Modell und anwenderfreundliche Korrekturen direkt über den Touchscreen.

### 3D-Drucker

Der 3D-Drucker ist ein anderes Beispiel, wo die Community-Firmware ihre Stärken zeigen kann. Dank der USB-Host-Schnittstelle ist der TXT nämlich nicht nur in der Lage, die mitgelieferte Kamera anzusprechen. Es ist vielmehr möglich, fast jedes USB-Gerät vom TXT aus anzusprechen. Der 3D-Print-Controller des fischertechnik-3D-Druckers ist genau so ein USB-Gerät.

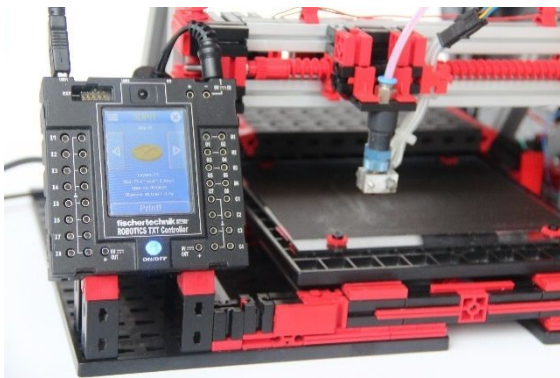


Abb. 5: 3D-Drucker vom TXT gesteuert

Softwareseitig kam wieder die bekannte QT-basierte Oberfläche und als Programmiersprache wieder Python zum Einsatz.

Wenige Kilobyte Code reichen aus, um alle wesentlichen Aspekte der Druckeransteuerung in einer kleinen App zusammen zu fassen. In erster Linie erlaubt es dem Benutzer der TXT auf diese Weise in sogenannten GCode-Dateien vorliegende Druckobjekte mit zwei einfachen Klicks ausdrucken zu lassen. Auch das 3D-Druckprogramm nutzt die Möglichkeiten des Farbbildschirms und erlaubt eine einfache Vorschau der zu druckenden Objekte.



Abb. 6: Oberfläche des 3D-Druck-Programms



Abb. 7: Das Druckprogramm in Aktion

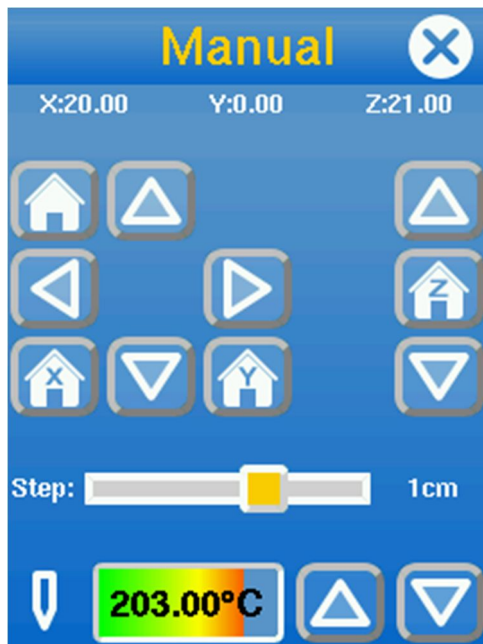


Abb. 8: Manuelle Steuerung

Auch eine manuelle Steuerung des Druckers zur Kalibrierung oder zum Filamentwechsel ist möglich. Die dazu nötige

Benutzerschnittstelle findet ebenfalls komplett auf dem kleinen Bildschirm des TXT Platz.

Der Entwicklungsaufwand selbst für so komplexe Projekte ist überschaubar, da für viele Funktionen auf fertige Bibliotheken zurückgegriffen werden kann. Code-Beispiele zur Bildverarbeitung finden sich ebenso im Internet wie Funktionen zum Auslesen und Verarbeiten von 3D-Druckdaten. Ich bin mir sicher, dass diese beiden Beispiele erst der Anfang sind und dass die Community-Firmware das Potenzial für einige weitere Überraschungen hat.

Der TXT ist technisch unter den intelligenten Spielzeug-Controllern nach wie vor nahezu konkurrenzlos. Aber erst mit der passenden Software kann er zeigen, was wirklich in ihm steckt.

## Quellen

[1] OpenCV-Projekt: <http://opencv.org/>

Computing

## Ein universeller I<sup>2</sup>C-Adapter für den TX(T)

Christian Bergschneider, Stefan Fuss

*Hersteller wie Adafruit oder Sparkfun haben für Arduino und Raspberry Pi viele Sensoren, Aktoren und Erweiterungs-Shields für kleines Geld im Angebot. Der Anschluss am TX(T) über den I<sup>2</sup>C-Bus ist nicht schwer zu realisieren. Allerdings ist es lästig, die Schaltungen jedes Mal auf einem Steckbrett neu zu erfinden. Mit dem hier vorgestellten I<sup>2</sup>C-Universal-Adapter läuft jedes Board im Handumdrehen am TX(T).*

### Die Idee

*Breakoutboards* sind Erweiterungsplatinen für den Arduino oder den Raspberry Pi. Die Boards enthalten meist einen IC, bspw. einen LED-Treiber, einen Gyrosensor oder einen Servotreiber, und können über den I<sup>2</sup>C-Bus angesprochen werden. Sie lassen sich daher mit wenig Aufwand an den TX(T) anschließen. Dafür benötigen sie in der Regel nur vier Anschlüsse: zwei I<sup>2</sup>C-Bus-Signale, GND und eine 3,3 V- oder 5 V-Stromversorgung.

Mit etwas Geschick lassen sich die Anschlüsse und erforderlichen Bauteile auf einem Steckbrett oder einer Lochrasterplatine zusammenbauen. Einfacher geht es mit dem hier vorgestellten universellen I<sup>2</sup>C-Adapter ftUNIpi: Ist die fertige Platine mit den wenigen Bauteilen bestückt, kann das Breakoutboard angeschlossen und direkt in ROBO Pro programmiert werden.

Die in unserem I<sup>2</sup>C-Universaladapter verwendeten Grundschaltungen sind unkompliziert und vielfach im Internet bis ins Detail beschrieben. Dieser Beitrag ist eine Bauanleitung. Wer die einzelnen Teile der Schaltung tiefgehend verstehen möchte, kann dies in den angegebenen Quellen nachlesen.

Die Programmierung ist nicht schwierig. Man muss sich dafür auch nicht durch das

Datenblatt des IC kämpfen: Zu den Breakoutboards gibt es in der Regel gut beschriebene Beispielprogramme in Python oder C – Abschreiben ist ausdrücklich erlaubt.

### Alltagstauglich

Bei der Entwicklung des Adapters stand die Alltagstauglichkeit im Vordergrund: klein und einfach nachzubauen (Abb. 1). Für den Aufbau sind nur geringe Elektronik- und normale Lötkenntnisse notwendig. Benötigt werden ein LötKolben mit „Bleistiftspitze“, eine Pinzette und ein einfaches Multimeter. Die vier verwendeten SMD-Bauteile werden nicht immer benötigt und sind auch für den SMD-Neuling gut zu verarbeiten.

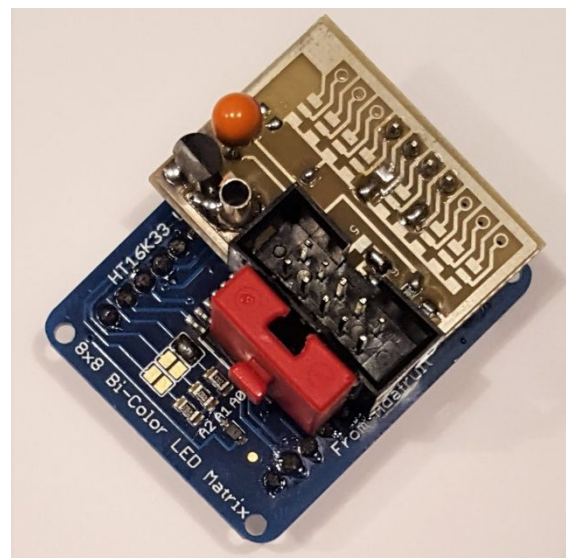


Abb. 1: ftUNIpi im Einsatz am ftpiLED



## Die Schaltung

Der Adapter kann in vier verschiedenen Varianten realisiert werden, sodass entweder 3,3 V- oder 5 V-Boards entweder an den TX oder an den TXT angeschlossen werden können. Diese verschiedenen Schaltungsvarianten werden durch unterschiedliche Bestückung und Lötjumper erreicht. Der Schaltplan in Abb. 2 ist übersichtlich.

### Stromversorgung

Mit einem kleinen Spannungsregler können bis zu 100 mA aus der 9V-fischertechnik-Bordspannung erzeugt werden. Durch Einsatz des passenden Reglers können alternativ 3,3 V oder 5 V bereitgestellt werden. Für Displays, Tastaturen und Sensoren reicht diese Leistung aus, für Motor- und Servo-HATs ist der Adapter nicht geeignet.

Die in [1] beschriebene Methode, die 3,3 V beim TXT an Pin 9 abzuzweigen, ist problematisch: Der Anschluss stellt keine belastbare Stromversorgung dar; sie ist der TX-Ausgang der seriellen Konsole.

Da GND vom TX(T) bereits über den EXT-Anschluss am Adapter anliegt, reicht es aus,

die 9 V-Versorgungsspannung über ein zusätzliches fischertechnik-Kabel anzuschließen. Damit ist der Adapter auch gegen Verpolung geschützt.

### Logikspannung

Entsprechend der Stromversorgung ist auch die Logikspannung am I<sup>2</sup>C-Bus unterschiedlich. Ältere ICs, wie auch der TX, arbeiten mit 5 V, neuere Bausteine und der TXT mit 3,3 V. Schließt man 5 V- und 3,3 V-Bauteile direkt zusammen, so funktioniert der I<sup>2</sup>C-Bus nicht. Der 3,3 V-Teilnehmer wird dabei dauerhaften Schaden nehmen.

Um Bausteine mit unterschiedlichen Spannungen zusammenzuschalten, wird ein *Levelshifter* benötigt. Viele der neueren Boards verfügen bereits über einen integrierten Levelshifter, so dass sie wahlweise mit 3,3 V oder mit 5 V betrieben werden können.

Der hier vorgestellte Adapter verwendet eine preiswerte Variante aus zwei MOSFETs und zwei Widerständen. Wird der Levelshifter nicht benötigt, so können die Bauteile einfach weggelassen werden. Die genaue Funktionsweise des Levelshifters ist in [2] beschrieben.

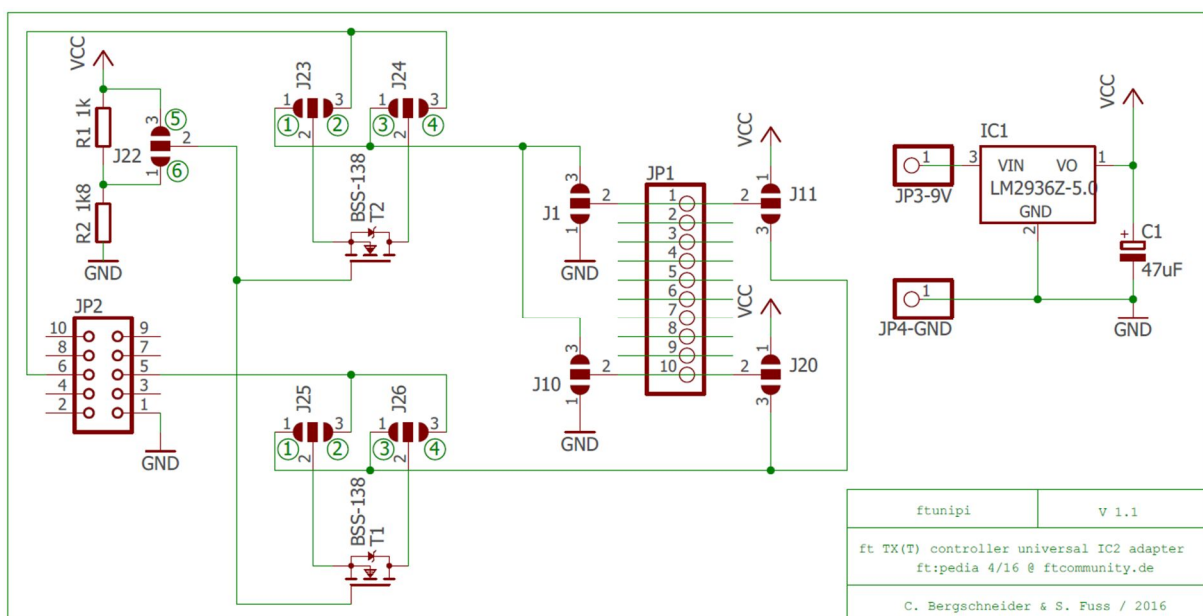


Abb. 2: Schaltplan

## Steckerleisten

TX und TXT haben unterschiedlich große EXT-Stecker, die Belegung der Pins ist aber kompatibel. Es reicht also aus, wahlweise einen 6-poligen oder einen 10-poligen Stecker für den Anschluss des Controllers zu verwenden [3].

Die meisten Breakoutboards legen die Anschlüsse auf eine einreihige Stiftleiste. Es gibt dafür keine Standardbelegung.<sup>3</sup> Der ftUNIpI kann über bis zu 10-polige einreihige Stiftleisten mit dem Breakoutboard verbunden werden. Die Zuordnung der Signale wird über Lötjumper festgelegt.

## Zusammenbau

Bei der Auswahl eines geeigneten Breakoutboards müssen nur zwei Dinge beachtet werden: Das Board muss die I<sup>2</sup>C-Schnittstelle unterstützen und die Stromaufnahme

darf 100 mA nicht überschreiten. Um den Universaladapter passend zum Board zu bestücken, muss die Betriebsspannung (3,3 V oder 5 V) in der Beschreibung oder dem Datenblatt nachgeschlagen werden.

## Bauteile

Für den Adapter werden nur wenige Bauteile benötigt. Die Bestückung ist je nach Anwendungsfall unterschiedlich. Dazu muss in Tab. 1 die richtige Kombination von TX(T) und der Versorgungsspannung des Breakoutboards ausgewählt werden. In der zugehörigen Spalte stehen die benötigten Bauteile sowie die Beschaltung der Lötjumper. An den Bauteilen sind Links zu verschiedenen Web-Shops hinterlegt, damit sich die Bauteile leichter bestellen lassen. Es gibt die gleichen Bauteile jedoch auch bei vielen anderen Lieferanten.

Controller	TXT		TX	
	VCC Board	5 V	3,3 V	5 V
Levelshifter	nicht benötigt	T1, T2: 2x <a href="#">BSS138</a> R1: 1 kOhm SMD R2: 1,8 kOhm SMD	T1, T2: 2x <a href="#">BSS138</a>	nicht benötigt
Stromversorgung	IC1: <a href="#">LP2950CZ-3,3</a> C1: <a href="#">Tantal-Elko 47uF</a>	IC1: <a href="#">LP2950CZ-5</a> C1: <a href="#">Tantal-Elko 47uF</a>	IC1: <a href="#">LP2950CZ-3,3</a> C1: <a href="#">Tantal-Elko 47uF</a>	IC1: <a href="#">LP2950CZ-5</a> C1: <a href="#">Tantal-Elko 47uF</a>
Anschlüsse	JP2: <a href="#">Wannenstecker, 10-pol, gerade</a> JP3: <a href="#">Bundhülse zum Einlöten, l=8,4 mm</a>		JP2: <a href="#">Wannenstecker, 6-pol, gerade</a> JP3: <a href="#">Bundhülse z. Einlöten, l=8,4 mm</a>	
Flachbandkabel	1x <a href="#">Flachbandkabel 10pol, grau</a> 2x <a href="#">Pfofenbuchse, 10pol, RM 2,54 mm</a>		1x <a href="#">Flachbandkabel 10pol, grau</a> 2x <a href="#">Pfofenbuchse, 6pol, RM 2,54 mm</a>	
Lötjumper	Positionen 1, 2, 3, 4 brücken	Positionen 2, 3, 6 brücken	Positionen 1, 4, 5 brücken	Positionen 1, 2, 3, 4 brücken
Platine	<a href="#">Doppelseitige Standardplatine</a> (100x80 mm, kein Bestückungsaufdruck, keine Lötstopmmaske, keine Durchkontaktierung, Bohrungen: 3,2 mm (JP3), 0,6 mm Durchkontaktierungen, sonstige Bohrungen 1 mm)			
Befestigung	fischertechnik Baustein 5 ( <a href="#">37237</a> )			

Tab. 1: Teileliste

<sup>3</sup> Breitere Breakoutboards haben manchmal zwei Steckerleisten, die auf den Arduino angepasst sind. Um den ftUNIpI klein zu halten, wurde auf

das Arduino-Design keine Rücksicht genommen. Hier muss ggf. die Stromversorgung über ein zusätzliches Kabel realisiert werden.

## Platine

Das Platinenlayout des Adapters steht auf der ft-community-Seite zur Verfügung (s. u.). Es gibt einige preiswerte Platinenbeleichter, die mit der Vorlage den Adapter herstellen. Wer keinen Bohrstand zur Verfügung hat, sollte die Platine fertig gebohrt bestellen. Die Herstellung dauert zwischen 10 und 20 Tagen.

Beim Herstellen von Platinen wird Basismaterial mit Standardmaßen verwendet. Der Preis richtet sich nach der Größe des Basismaterials. Der Adapter ist so klein, dass er sechs Mal auf eine 100x80 mm Platine passt – Material für den Anschluss vieler I<sup>2</sup>C-Breakout-Boards.

## ftExtender

Der Anschluss von I<sup>2</sup>C-Bausteinen über den ftUNiPi am TX(T) ist so einfach, dass man schnell einige Breakoutboards zusammenbekommt. Will man diese in einem Modell einbauen, so lässt sich dies mit dem ftExtender flexibel gestalten.

Der ftExtender besteht in der TXT-Variante aus sieben 10-poligen Steckern, bei denen auf der Platine die Anschlüsse für den I<sup>2</sup>C-Bus verbunden sind. In der TX-Fassung sind es zehn 6-polige Stecker. Es kann mit Sekundenkleber direkt auf einer Bauplatte 38248 befestigt werden. Ein ähnliches Verteilermodule mit sechs Steckern passend für den TX gibt es auch bei Conrad ([192659-62](#), Abb. 3).

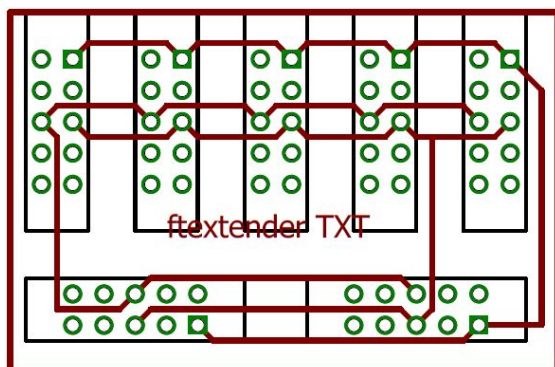


Abb. 3: Platinenlayout ftExtender

Auf dem Layout für das Standardmaß war noch Platz, sodass wir einen ftExtender mit untergebracht haben.

## Platinenlayout

Im [Downloadbereich](#) der ft:pedia steht ein ZIP-File mit mehrere Layoutvarianten zum Download bereit:

- Die Dateien `ftunipi_TXT_multi.brd` und `ftunipi_TX_multi.brd` sind große Vorlagen (100x80 mm), die sechs gleiche Universaladapter und einen ftExtender für TXT oder TX enthalten.
- `ftunipi_TXT_v11.brd` enthält nur einen Universaladapter – für alle die ihn selbst noch verändern wollen.

Die Dateien liegen im EAGLE-Format vor, das alle Belichtungsdienstleister verarbeiten können.

## Lötarbeiten

Die Lötarbeiten sollten in der hier beschriebenen Reihenfolge erfolgen. Die Position der Bauteile kann im Bestückungsplan (Abb. 4) abgelesen werden.

- Zunächst wird in den drei Durchkontaktierungen ein dünner Draht eingelötet.
- Wird der Levelshifter benötigt, so müssen nun die beiden MOSFETs (T1, T2) auf der Rückseite aufgelötet werden.
- Um SMD-Bauteile zu montieren, empfiehlt es sich die Löt pads zunächst ein wenig zu verzinnen. Das MOSFET mit der Pinzette in Position halten und kurz anlöten.
- Anschließend die beiden SMD-Widerstände (R1, R2) auf der Vorderseite auflöten.
- Die Lötjumper an den Widerständen und an den MOSFETs gemäß der Bestückungsliste oben setzen. Die Lötjumper 1-4 sind an beiden MOSFETs vorhanden. Es müssen immer die Lötjumper an beiden MOSFETs gesetzt werden.

- Nun die Bundhülse einlöten.
- Es folgen der Spannungsregler, der Elko und der Stecker für den EXT-Anschluss.

Beim Elko muss auf die Polung geachtet werden. Der +-Pol ist mit einem Strich und/oder einem +-Zeichen gekennzeichnet.

Als Nächstes wird die einreihige Stiftleiste für den Anschluss des Breakoutboards auf der Unterseite angebracht. Die lange Seite der Stifte wird am Adapter eingelötet. Somit ist es möglich alle Stifte auf der Ober- und der Unterseite anzulöten. Das Breakoutboard noch nicht verbinden!

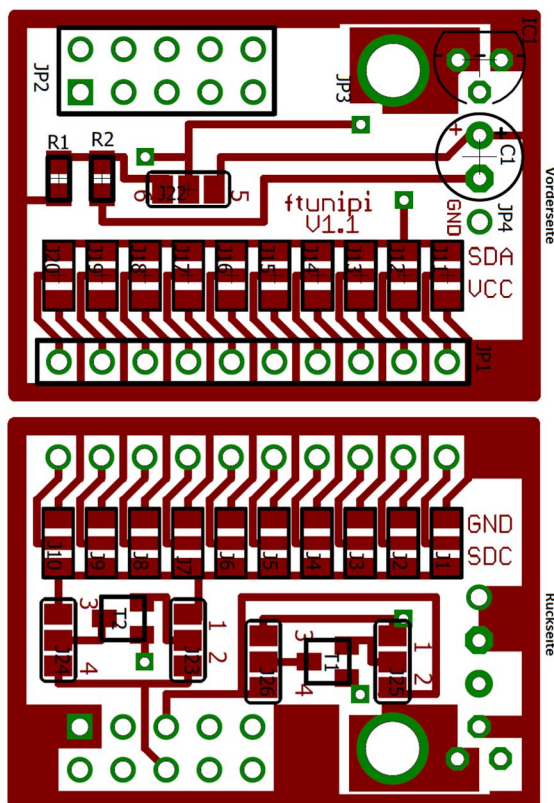


Abb. 4: Platinenlayout

Die Lötjumper neben der Stiftleiste müssen jetzt gesetzt werden. Die Belegung bitte in der Dokumentation des Breakout-Boards nachschlagen (Tab. 2).

Name	Funktion
SDA, D	I <sup>2</sup> C-Bus Daten
SDC, SCL, C	I <sup>2</sup> C-Bus Clock
VCC, VSS, +	Betriebsspannung, Pluspol
GND, -	Betriebsspannung, Minuspol

Tab. 2: Unterschiedliche Bezeichnungen der Anschlüsse bei den Boards

An den mit GND bezeichneten Anschluss JP4 neben dem Elko C1 kann für die nachfolgende Qualitätskontrolle ein Kabel zur Spannungsversorgung ohne TX(T) angelötet werden. Es wird nach der Kontrolle wieder entfernt.

### Qualitätskontrolle

Bevor der Adapter mit dem TX(T) und dem Breakoutboard verbunden wird, sollte der Adapter auf Lötfehler untersucht werden. Schließt man den Adapter an eine 9V-Batterie (JP3, JP4) an, so können die Spannungsversorgung und die Logikpegel an TX(T)- und Breakout-Anschluss mit dem Multimeter kontrolliert werden. Dieser Schritt sollte beim TXT unbedingt durchgeführt werden: **5 V am EXT-Stecker des TXT können diesen beschädigen.**

### Fertigstellen

Zum Abschluss wird das Hilfskabel an JP4 wieder entfernt.

Viele Breakoutboards haben Lötjumper um die I<sup>2</sup>C-Adresse des Bausteins einzustellen. An einem I<sup>2</sup>C-Bus darf jede Adresse nur einmal vergeben werden.<sup>4</sup> Das Breakoutboard nun an der Stiftleiste anlöten. Die Boards sind in der Regel durchkontaktiert, so dass es ausreicht, den Stecker auf einer Seite anzulöten.

Um Kurzschlüsse zu vermeiden, muss darauf geachtet werden, dass genügend

<sup>4</sup> Wer auch noch einen PCA9685 wie in [3] im Einsatz hat, darf die Adresse 0x70 nicht vergeben. Auf diese Adresse reagiert der PCA9685 unabhängig von seiner Basisadresse.

Abstand zwischen dem Adapter und den Bauteilen des Boards vorhanden ist.

Den fischertechnik-Baustein 5 mit Sekundenkleber an den EXT-Stecker des ftUNipi ankleben, so dass das Breakoutboard fest ins fischertechnik-Modell eingebaut werden kann. Dabei kann auch das eigentliche Breakoutboard entlastet werden.

Zuletzt noch das Verbindungskabel zum TX(T) herstellen. Dazu mit einer Rohrzange vorsichtig die Kabel crimpen. Das rot markierte Kabel muss an der Pfeil-Markierung der Buchse sein.

Sechspoliges Flachbandkabel für den TX ist nur schwierig zu bekommen; alternativ kann man zehnpoliges Kabel einfach um vier Adern kürzen.

## Programmierung

Für einige Bausteine gibt es bereits fertige ROBO Pro-Module. Zu den Breakoutboards findet man auf den Seiten der Hersteller Beispielprogramme in C oder Python. Die Sprachen lassen sich gut lesen, sodass man die notwendigen Kommandos leicht in ROBO Pro übersetzen kann.

### ROBO Pro

Die I<sup>2</sup>C-Programmierung in ROBO Pro besteht aus zwei Kommandos. Um die Kommandos zu aktivieren, muss im Menüpunkt Level mindestens „Level 4: Benutzerdefinierte Befehle“ ausgewählt werden. Es stehen dann unter Grundelemente/Senden die Befehle „I<sup>2</sup>C Schreiben“ (Abb. 5) und „I<sup>2</sup>C Lesen“ zur Verfügung [4].

Die Parametrierung findet in den Eigenschaften des Befehls statt. Jeder I<sup>2</sup>C-Baustein hat eine Geräteadresse über die er adressiert wird. Diese ist entweder fest durch das Breakoutboard vorgegeben oder wird per Lötjumper gesetzt. In den Dokumentationen der Boards wird die Geräteadresse häufig hexadezimal angegeben. Eine hexadezimale Zahl kann in ROBO Pro mit dem Präfix „0x“ dargestellt werden (0x60 = 96).

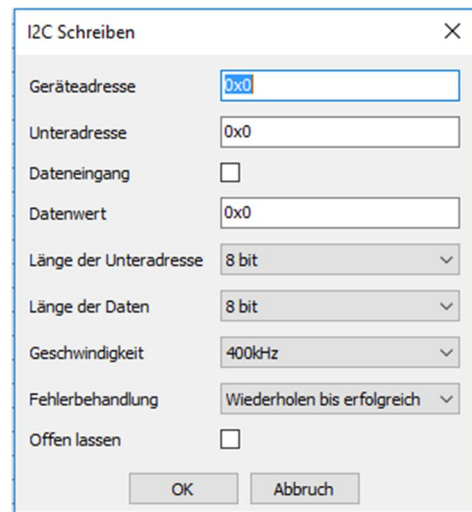


Abb. 5: Eigenschaften der I<sup>2</sup>C-Befehle

Hat der Baustein mehrere Register, so werden diese über Unteradressen adressiert. Bei Bausteinen mit nur einem Register muss das Feld auf 0x0 gesetzt werden und bei Länge der Unteradresse muss „Keiner“ ausgewählt werden.

Bei der Verwendung mehrerer Register wird die Adresse des Registers in der Unteradresse eingetragen. In den meisten Fällen ist der Wert „8-bit“ für die Länge der Unteradresse korrekt.

Einige Bausteine arbeiten mit mehreren Registern ohne Unteradressen. Das Register ist dann in einigen Bits des zu schreibenden Wertes enthalten [4, 5]. Mit einem Schreibbefehl kann ein bis zu 16 Bit großer Wert als „Datenwert“ an den Baustein geschrieben werden. Die meisten Bausteine arbeiten allerdings mit acht Bit, 16-Bit-Bausteine vertragen in der Regel auch zwei aufeinanderfolgende 8-Bit-Übertragungen. Wegen eines Fehlers in der TX(T)-Firmware bei der 16-Bit-Übertragung im LSB-Format sollte besser die 8-Bit-Übertragung verwendet werden.

Wenn das Breadboard den I<sup>2</sup>C *Fast Mode* unterstützt, sollten als Geschwindigkeit 400 kHz gewählt werden. Führt dies zu Problemen, kann die Geschwindigkeit auf 100 kHz reduziert werden. Die Kabellänge ist beim I<sup>2</sup>C-Bus unproblematisch, da der

Bus mit bis zu 400 m Kabellänge spezifiziert ist. Bei den meisten Modellen verzichtet man auf eine Fehlerbehandlung und wählt „Wiederholen bis erfolgreich“.

Der I<sup>2</sup>C-Bus kennt die Übertragung eines ganzen Datenblocks aus mehreren Bytes. Technisch wird über den Bus zunächst ein Start-Sequence-Kommando mit Bausteinadresse und ggf. Unteradresse gesendet. Anschließend werden ohne Wiederholung der Adressen die Datenbytes gesendet. Zum Abschluss der Übertragung wird ein Stop-Sequence-Kommando übertragen [4, 5].

Bei ROBO Pro gibt es keine expliziten Start/Stop-Sequence-Befehle. Der erste I<sup>2</sup>C-Senden- oder I<sup>2</sup>C-Empfangen-Befehl startet immer die Übertragung auf dem Bus mit dem „Start-Sequence-Kommando“. Ist die CheckBox „offen lassen“ nicht ausgewählt, sendet ROBO Pro nach Ende des I<sup>2</sup>C-Befehls das Stop-Sequence-Kommando über den Bus.

Ist „Offen lassen“ ausgewählt, so werden die nächsten I<sup>2</sup>C-Bus-Befehle in dem gleichen Datenblock gesendet. Beim letzten I<sup>2</sup>C-Bus-Befehl eines Datenblocks darf „Offen lassen“ nicht gesetzt werden; der TX(T) sendet dann automatisch das Stop-Sequence-Kommando.

Die Interpretation des Datenblocks ist bausteinspezifisch. Bausteine mit Unterregistern adressieren in der Regel mit jedem neuen Datenbyte das nächste Unterregister.

In C und Python sind die Start-Stop-Kommandos über eigene Befehle (`begin-transmission`, `end-transmission`) implementiert.

### Erste Tests

Als erster Schritt sollte mit einem einfachen Kommando versucht werden, den Baustein anzusprechen. Dazu kann man in den Beispielprogrammen des Herstellers nach der Initialisierung des Bausteins suchen und das erste Kommando daraus mit einem ein-

fachen ROBO Pro-Programm an den Baustein senden. Läuft der TXT in einen Fehler, so ist entweder die Adresse des Bausteins falsch eingestellt, oder es liegt ein Bestückungsfehler vor. Läuft der Befehl problemlos, so ist die Hardware richtig angeschlossen und man kann sich rein um die Programmierung kümmern.

### Bug in ROBO Pro

In ROBO Pro gibt es (zumindest beim TXT) einen weiteren Bug mit der I<sup>2</sup>C-Schnittstelle: Wartet man zwischen zwei Buskommandos mehr als fünf Sekunden, so schaltet sich der I<sup>2</sup>C-Bus ab. Alle folgenden Kommandos schlagen fehl.

Mit einem I<sup>2</sup>C-Bus-Kommando ohne echte Funktion (z. B. Auslesen eines Registers) in einer eigenen nebenläufigen Schleife kann man das Abschalten verhindern (Abb. 6).

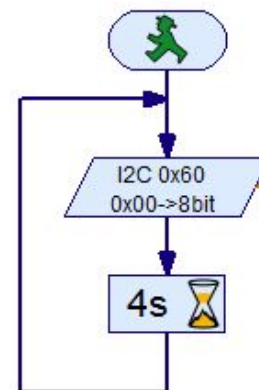


Abb. 6: Workaround ROBO Pro-Bug

### Namensänderungen

Schnell hat man Bibliotheken mit vielen I<sup>2</sup>C-Befehlen aufgebaut. Achtung: größere ROBO Pro Programme können beim Speichern schon mal einige Sekunden Zeit benötigen.

Den Namen von Unterprogrammen sieht man ihre Herkunft nicht an. Braucht man für zwei unterschiedliche Breakoutboards je ein Unterprogramm für die Initialisierung, so ist die mehrmalige Verwendung der Bezeichnung „Init“ zwar zulässig, aber unpraktisch.

Die einfache Lösung ist, bei allen Unterprogrammen einer Bibliothek ein Präfix für die Bibliothek zu verwenden. In der Bibliothek `ftpi.rpp` würde dann aus dem Unterprogramm `init` `ftpi_init`.

Das zweite Problem ist die feste Zuweisung der I<sup>2</sup>C-Adressen. Es ist nicht möglich, diese über einen Dateneingang am Befehl zu setzen. Will man zwei gleiche Breakoutboards anschließen, so muss man die Bibliothek duplizieren – um Konflikte zu vermeiden, am besten mit der Bausteinadresse im Namen.

Die Namenskonvention

```
<eigentliche Bibliothek>_  
<Bausteinadresse>_<Funktion>
```

ergibt leicht lange Namen. ROBO Pro kann diese dann nicht mehr vollständig darstellen. Sinnvolle Abkürzungen des Bibliotheknamens helfen da weiter.

Unterprogrammnamen können in ROBO Pro nicht geändert werden. Auch die massenweise Änderung der I<sup>2</sup>C-Kommandos ist fehleranfällig, so dass man für diese beiden Schritte auf *Workarounds* ausweichen muss: ROBO Pro-Programme werden als XML-Dateien gespeichert; man kann diese in einem ASCII-Editor wie notepad bearbeiten und sich mit Suchen & Ersetzen die Arbeit erleichtern.

Bei Unterprogrammnamen reicht es aus, nach dem genauen Namen zu suchen und diesen durch den neuen Namen zu ersetzen. I<sup>2</sup>C-Adressen sind schwieriger. Um von

Adresse 0x71 auf 0x72 umzustellen, muss man nach

```
device="113" deviceRadix="16"
```

suchen und alle Vorkommen mit

```
device="114" deviceRadix="16"
```

ersetzen.

### **Risiken und Nebenwirkungen**

Der Nachbau der Platine erfolgt auf eigenes Risiko. Durch die Lötjumper lassen sich schnell 5 V-Signale auf die falschen Anschlüsse legen, daher sollte die Schaltung immer zunächst ohne TX(T) und ohne Breakoutboard nachgemessen werden.

### **Quellen**

- [1] Dirk Fox, [I<sup>2</sup>C mit dem TX\(T\) – Teil 12: Temperatursensor](#), ft:pedia 4/2015, S. 44-48.
- [2] playground.arduino.cc: [I<sup>2</sup>C bi-directional level shifter](#).
- [3] Christian Bergschneider, Stefan Fuss: [Alternative Controller \(3\): Der ftPi – ein Motor Shield für den TX\(T\)](#), ft:pedia 2/2016, S. 68-72.
- [4] Dirk Fox: [I<sup>2</sup>C mit dem TX und ROBO Pro – Teil 1: Grundlagen](#). ft:pedia 3/2012, S. 32-37.
- [5] embed journal: [I<sup>2</sup>C Protocol \(2-Wire Interface\) in a nut shell](#). 13.05.2013.
- [6] Christian Bergschneider, Stefan Fuss: [ftpiLED, LED-Backpack im Retrodesign](#). ft:pedia 4/2016 (in dieser Ausgabe).

Computing

## LED-Backpack im Retrodesign

Christian Bergschneider, Stefan Fuss

*Handy-Apps mit pixeliger Grafik sind cool, Retro ist in. Höchste Zeit, dem TX(T) etwas 80er Look zu verpassen. Warum nicht mit einer 8x8-LED-Anzeige mit C64-Zeichensatz? Wer will, kann sich mit einfachen Mitteln selbst seinen Zeichensatz für die ftpiLED definieren.*

### Die Matrix

Die ftpiLED besteht aus einer 8x8-Bicolor-Matrix von Adafruit (Abb. 1). Wie die meisten Arduino- und Raspberry Pi-Breakoutboards kommt die kleine Matrix mit I<sup>2</sup>C-Bus-Interface.

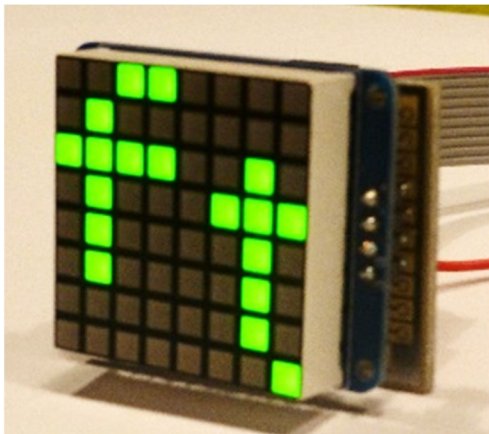


Abb. 1: LED-Bicolor-Matrix

Jedes der 64 Pixel besteht aus einer roten und einer grünen LED. Obwohl das Breakoutboard als Bicolor-Matrix verkauft wird, lassen sich so drei Farben darstellen: Schaltet man die rote und die grüne LED eines Pixels gleichzeitig an, so leuchtet das Pixel gelb.

Kern der Matrix ist der 16x8-LED-Treiber HT16K33. Adafruit hat eine ganze Reihe unterschiedlicher Breakoutboards mit diesem Baustein im Programm: Mehrere 8x8-Matrizen in verschiedenen Farben sowie 7-Segmentanzeigen mit vier Digits [2].

Die 8x8-Matrizen lassen sich alle mit der hier beschriebenen ROBO Pro-Bibliothek ansteuern; für die 7-Segmentanzeige können nur Teile aus der vorliegenden Bibliothek genutzt werden.

### Der Anschluss am TX(T)

Mit dem ftUNipi aus [1] ist der elektrische Anschluss schnell erledigt. Der ftUNipi wird mit einem 5 V-Spannungsregler bestückt. Für den Anschluss am TX(T) wird zusätzlich der beschriebene Levelshifter (R1, R2, T1, T2) benötigt. Der TX als 5 V-I<sup>2</sup>C-Busdevice kommt ohne Levelshifter aus. Lady Ada [2] gibt zwar an, der Baustein würde auch direkt mit 3,3 V Betriebsspannung funktionieren; dies liegt aber außerhalb der Spezifikation des HT16K33.

Die Matrix von Adafruit wird fast fertig geliefert. Es muss nur noch die LED-Matrix auf die Platine aufgelötet werden. Beim anlöten der Matrix muss auf die Ausrichtung geachtet werden; sie funktioniert nur, wenn der Aufdruck an der Matrix zum weißen Punkt auf der Platine passt.

Im Standard hat das Board die I<sup>2</sup>C-Bus-Adresse 0x70. Diese sollte vermieden werden – u. a. benutzt der I<sup>2</sup>C-PWM-IC PCA9685 diese Adresse. Die rpp-Bibliothek im Downloadbereich arbeitet mit Adresse 0x71. Dazu muss auf dem Board der Lötjumper A0 gesetzt werden. Damit der ftUNipi komplett hinter der Matrix verschwindet, wird das Breakoutboard am



ftUNIpI über die vier Pins JP1.4 bis JP1.8 angeschlossen. Die Lötjumper werden wie in Tab. 1 beschrieben gesetzt.

Matrix	Lötbrücken am ftUNIpI	Funktion
C	J4 auf SDC	I <sup>2</sup> C Clock
D	J15 auf SDA	I <sup>2</sup> C Data
-	J6 auf GND	GND
+	J17 auf VCC	+5V

Tab. 3: Anschluss der Matrix an den ftUNIpI

### Die Register

Der Baustein benötigt nur acht Register. Diese werden nicht über Unteradressen adressiert, sondern über die I<sup>2</sup>C-Adresse des Bausteins.

Beim Schreiben an den HT16K33 wird in den oberen vier Bits eines Befehls das Register definiert, über die unteren vier Bits werden die gewünschten Einstellungen übertragen.

Um Pixel auf der Matrix zu setzen, muss in den internen Display Buffer geschrieben werden. Dieser ist ein 16 Byte großer RAM-Bereich im Baustein. Jedes Bit entspricht darin einer LED. Ist das Bit gesetzt, so ist die LED angeschaltet. Die Bytes an den geraden Adressen (0, 2, 4, ...) steuern die grünen LEDs, die Bytes an den ungeraden Adressen (1, 3, 5, ...) steuern die roten LEDs an.

Der Display Buffer wird über den *Display Data Address Pointer* angesprochen. Bei diesem Wert sind die vier oberen Bits 0, in den vier unteren Bits wird eine Adresse im Display Buffer mitgegeben.

0x0E	0	0	0	0	1	1	0	0
0x0A	0	0	0	0	1	0	1	0
LED								

Tab. 2: Farbmischung über zwei aufeinander folgende Bytes im Display Buffer

Werden die Daten 0x06, 0x0E, 0x0A an den ftpiLED in einem Block übertragen, so erhalten im Display Buffer das Byte #6 den

Wert 0x0E und das Byte #7 den Wert 0x0A. Dies führt dazu, dass in der vierten Zeile jeweils eine LED grün, eine rot und eine gelb leuchtet. Die anderen LEDs sind ausgeschaltet (Tab. 2).

### Listen in ROBO Pro

Für die Programmierung ist es einfacher, einen zweiten Display Buffer in ROBO Pro zu implementieren. Alle Operationen finden zunächst auf dem ROBO Pro Display Buffer statt. Soll die Änderung auf die Matrix übertragen werden, so werden alle 16 Bytes in den Baustein übertragen.

Arrays sind in ROBO Pro als Listen implementiert. Eine Liste ist eine Aufzählung von  $n$  Werten. Die einzelnen Werte können über deren Position in der Liste [0 ..  $n-1$ ] angesprochen werden.

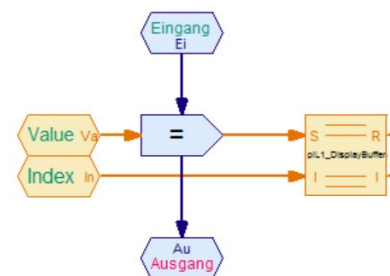


Abb. 2: Schreiben eines Wertes in ein ROBO Pro-Listenelement

Um in einer Liste an Position „Index“ den Wert „Value“ zu schreiben, wird die Position an den I-Eingang der Liste und der Wert mit einem = Befehl an den S-Eingang übergeben (Abb. 2).

Zum Auslesen wird die Position mit einem = Befehl an den I-Eingang der Liste gesendet. Der Wert wird dann am R-Ausgang ausgegeben (Abb. 3).

Multidimensionale Listen gibt es in ROBO Pro nicht. Dies muss über mehrere getrennte Listen abgebildet werden. Listen in ROBO Pro können ihre Werte in CSV-Dateien speichern oder zum Programmstart auslesen. Diese Funktionen können über das Eigenschaftsfenster der Liste eingestellt werden (Abb. 4).

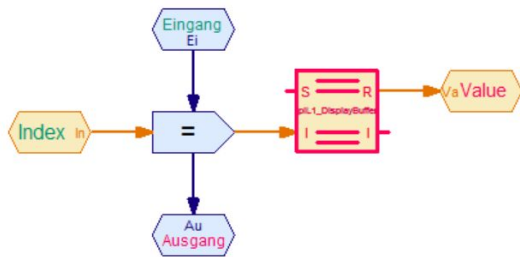


Abb. 3: Auslesen eines Werts aus einer Liste

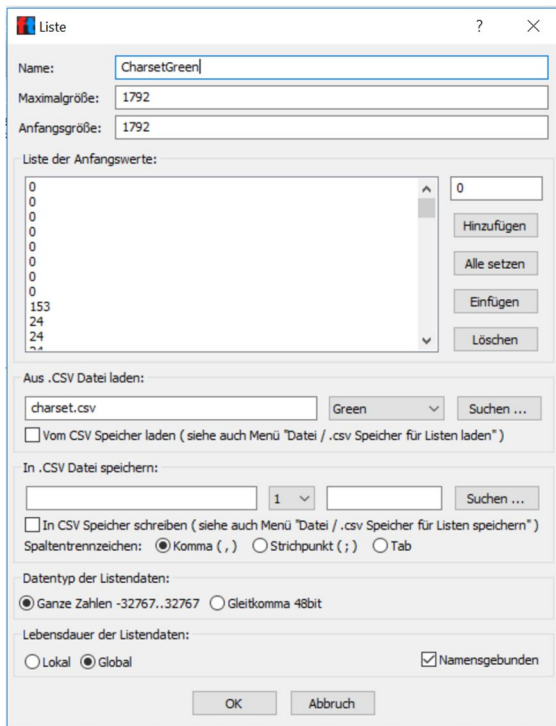


Abb. 4: Eigenschaften einer ROBO Pro-Liste

Im Feld „Aus .CSV Datei laden“ kann man eine CSV-Datei auf dem PC angeben, deren Werte in die Liste geladen werden. Im Standard wird die erste Spalte aus der CSV-Datei in die Liste übernommen. Sind in der CSV-Datei mehrere Spalten vorhanden, so kann man die gewünschte Spalte über die Combobox hinter dem Dateinamen auswählen. Die erste Zeile in der CSV-Datei wird als Überschrift interpretiert und nicht in die Liste übernommen.

Nach dem Laden stehen die Daten im Programmtext. Die Datei muss dann nicht mehr vorhanden sein. Ist sie vorhanden, so aktualisiert ROBO Pro den Inhalt im Programm automatisch.

### Der Zeichensatz

Der Zeichensatz wird als CSV-Datei in die ROBO Pro-Bibliothek übertragen. Da das Editieren des Zeichensatzes in der CSV-Datei kompliziert ist, haben wir eine Excel-Datei erstellt, in der man jedes Zeichen selbst gestalten kann.

Jedes Pixel entspricht einer Zelle im Excel-Arbeitsblatt. Je nach Wert der Zelle wird das Pixel später ausgeschaltet, grün, gelb oder rot sein.

Über ein kleines VBA-Makro wird die CSV-Datei mit dem Zeichensatz geschrieben. Dazu muss man einfach nur auf dem Arbeitsblatt den Button „Zeichensatzdatei für den ftpiLED schreiben“ drücken. Die Datei landet im aktuellen Verzeichnis, in der Regel ist dies „%USER%/Dokumente“. Bei Bedarf kann auch im Feld für den Dateinamen der gewünschte Pfad ergänzt werden.

65										130									
0	0	1	1	1	1	0	0	0	0	0	0	3	3	3	3	3	0	0	0
0	1	1	0	0	1	1	0	0	0	0	3	3	3	0	3	3	0	0	0
0	1	1	0	1	1	1	0	0	0	1	1	1	0	3	3	3	3	3	3
0	1	1	0	1	1	1	0	0	0	1	1	1	0	3	3	3	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	3	3	3	0	0	0
0	1	1	0	0	0	0	1	0	0	0	0	0	0	3	3	3	3	3	3
0	0	1	1	1	1	0	0	0	0	0	0	0	3	3	3	3	3	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	3	0	0	0

Abb. 5: @ und Pacman im Zeichensatz

Nun muss noch der Zeichensatz in der Bibliothek geladen werden. Normalerweise lädt ROBO Pro den Zeichensatz selbständig nach. Bei Problemen können im Unterprogramm piLl\_WriteChar über die Eigenschaftsfenster der Listen CharsetGreen und CharsetRed die Datei neu geladen werden.

### Die Bibliothek

Die ROBO Pro-Bibliothek ist recht umfangreich. Das Hauptprogramm zeigt dabei ein Beispiel für die Nutzung des Bausteins. Sie kann im [Downloadbereich](#) der ft:pedia heruntergeladen werden.

Am Anfang eines ROBO Pro-Programms muss der Baustein immer mit dem Kommando `piL1_init` initialisiert werden. Danach können beliebige Kommandos benutzt werden. Kommandos, deren Namen mit einem „\_“ beginnt, sind interne Funktionen und sollten von außen nicht angesprochen werden. `piL1_ftNOP` führt alle vier Sekunden eine Leseoperation auf den `ftpiLED` durch und vermeidet so den in [1] beschriebenen „Fünf-Sekunden-Bug“.

### Nebenwirkungen

Die Bibliothek ist recht umfangreich. ROBO Pro benötigt erstaunlich lange um

sie zu speichern. Importiert man die Bibliothek in sein Projekt, so wird das Speichern des Projektes ebenfalls sehr träge. Der Nachbau der Platine erfolgt auf eigenes Risiko.

### Quellen

- [1] Christian Bergschneider, Stefan Fuss: [ftUNipi: universeller PC-Adapter für den TX\(T\)](#). ft:pedia 4/2016, in dieser Ausgabe.
- [2] Lady Ada: [Adafruit LED Backpacks](#).
- [3] Holtec: [RAM Mapping 16\\*8 LED Controller Driver with keyscan HT16K33](#).

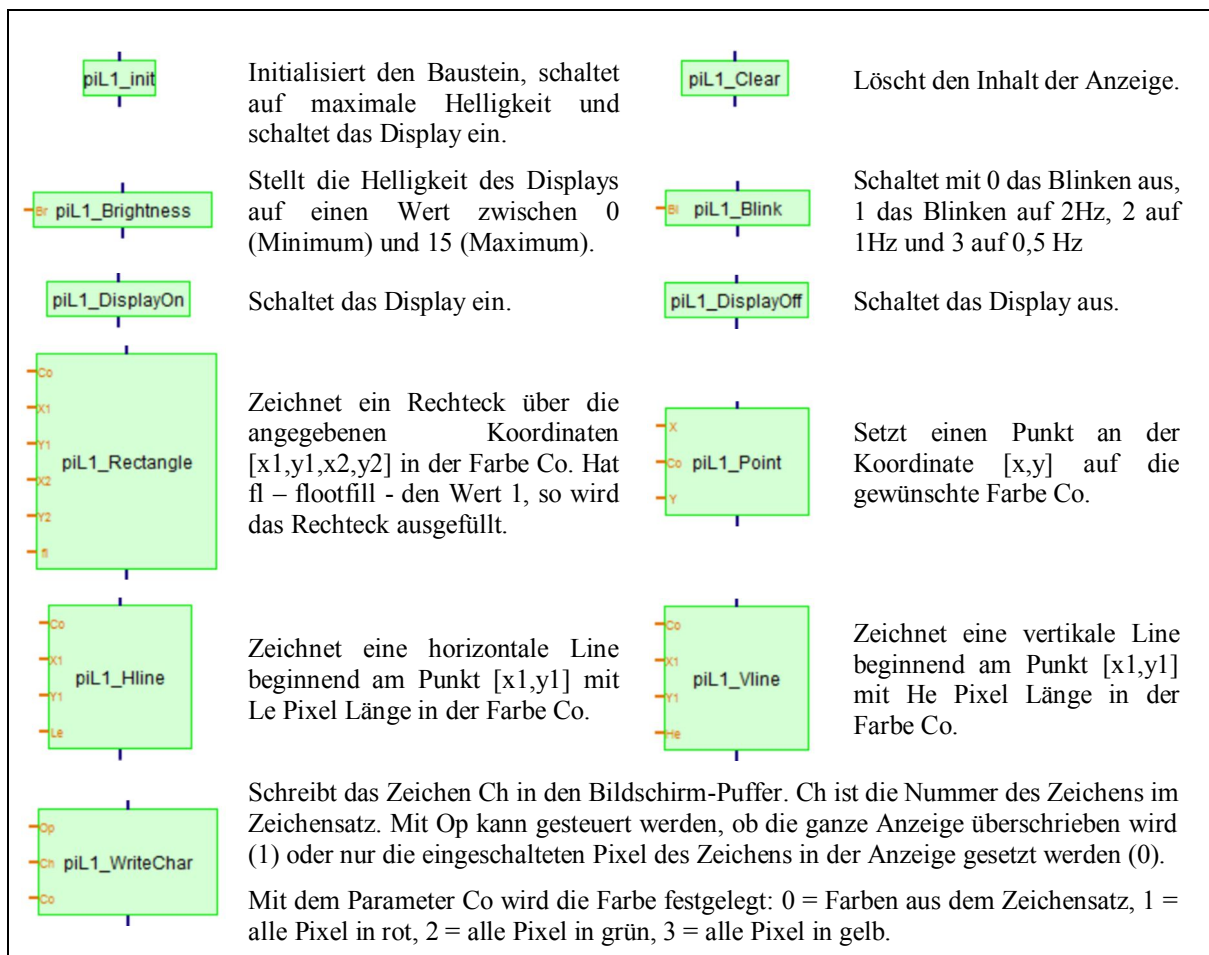


Abb. 6: Funktionsreferenz



Computing

## I<sup>2</sup>C mit dem TX(T) – Teil 14: LED-Display (2)

Dirk Fox

*In einem der ersten Beiträge der I<sup>2</sup>C-Serie habe ich Ende 2012 die Ansteuerung des I<sup>2</sup>C-LED-Displays SAA1064 von Philips vorgestellt. Inzwischen gibt es zahlreiche attraktive Alternativen zu diesem LED-Modul – schneller (Fast Mode), günstiger, in verschiedenen Farben und weit einfacher anzusteuern. Eines dieser Module und den passenden ROBO Pro-Treiber stelle ich hier vor.*

### Das Sparkfun-Display

Von Sparkfun gibt es seit 2011 ein vierstelliges LED-Display, das in der aktuellen Version auch über eine I<sup>2</sup>C-Schnittstelle verfügt (Abb. 1). Es ist in fünf verschiedenen Farben (blau, gelb, rot, weiß und grün) für die Hälfte des Preises des in [ft:pedia 4/2012](#) [1] vorgestellten Philips-Displays SAA1064 z. B. bei [EXP](#) für 14,25 € erhältlich.



Abb. 1: Sparkfun, COM-11440 bis -3 (CC)

Das Display ist mit etwa 1,5 x 4 cm etwas kleiner als das von Conrad angebotene Philips-Display. Allerdings besitzt es eine Reihe praktischer Vorzüge:

- es kann sowohl am TX als auch am TX(T) betrieben werden,
- es hat einen signifikant geringeren Stromverbrauch,
- es können via Software 126 verschiedene I<sup>2</sup>C-Adressen eingestellt werden, sodass Kollisionen mit anderen I<sup>2</sup>C-

Boards vermieden und zahlreiche Displays für unterschiedliche Anzeigen gleichzeitig angesteuert werden können,

- es unterstützt den I<sup>2</sup>C *Fast Mode* (400 kHz),
- es verfügt über zusätzliche Dezimalpunkte und Trennzeichen und über eine deutlich einfachere Ansteuerung.

Schließlich beherrscht es auch das SPI-Protokoll und die direkte Ansteuerung über ein serielles RX/TX-Protokoll.

### Die Hardware

Das Sparkfun-Display ist eine vierstellige 7-Segment-Anzeige, die mit 2,4-5,5 V betrieben werden kann. Der Stromverbrauch liegt bei einer Betriebsspannung von 5 V mit maximal 14,1 mA bei einem Dreißigstel des Verbrauchs des Philips-Moduls; bei einer Betriebsspannung von 3,3 V liegt er maximal bei 7,9 mA.

Vorkonfiguriert ist die I<sup>2</sup>C-Adresse 0x71; per I<sup>2</sup>C-Kommando können 126 verschiedene Adressen eingestellt und damit ebenso viele Devices ohne Multiplexer auf demselben I<sup>2</sup>C-Bus unabhängig von einander genutzt werden.

Der ATmega328 (Abb. 2) bietet einen umfangreichen Befehlssatz, mit dem das Display auch befehlskompatibel zum Philips

SAA1064 angesteuert werden kann. Effizienter ist es allerdings, die optimierten Befehle des Sparkfun-Displays zu verwenden.

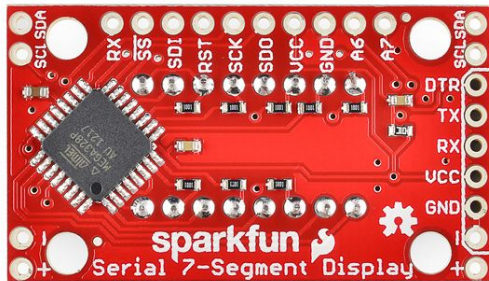


Abb. 2: Sparkfun, COM-11440 bis -3 (CC)

Die I<sup>2</sup>C-Anschlüsse des Moduls sind gedoppelt und so angeordnet, dass sich mehrere Module sehr elegant in Reihe schalten lassen – einfach die rechten Anschlüsse des einen Moduls (+/- bzw. SDA/SCL) mit den linken des nächsten via Jumper verbinden. Für den Anschluss an den TX empfiehlt sich die Verwendung des in [ft:pedia 4/2013](#) vorgestellten „Universal-Adapters“ [2].

Über vier Bohrungen lassen sich die Module gut befestigen. Die wunderbare Alternative mit selbst „gecutteten“ fischer-technik-kompatiblen Plexiglas-Fenstern in Abb. 3 und 7 stammt von Georg Stiegler.

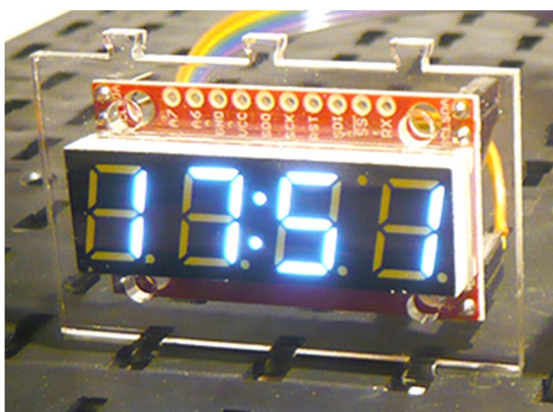


Abb. 3: Plexiglas-Fenster von Georg Stiegler

Die Farbe des Moduls kann man im ausgeschalteten Zustand an einem farbigen Punkt an der Seite der Segmentanzeige erkennen. Aufpassen muss man allerdings, wenn man die I<sup>2</sup>C-Adresse ändert: Will man

später nicht alle Adressen durchprobieren, muss man sie notieren.

## Befehle

Die Ansteuerung des Moduls ist sehr einfach. Will man vier Zeichen ausgeben, werden einfach diese vier Zeichen via I<sup>2</sup>C-Befehl an das Modul geschickt.

Zu den darstellbaren Zeichen gehören neben den Ziffern 0-9 zunächst die sechs Buchstaben, die für die Wiedergabe von Hexadezimalzahlen (A-F) benötigt werden. Aber auch einige Groß- und Kleinbuchstaben lassen sich mit dem Display darstellen.

Dabei können wahlweise die Werte (Ziffern), ASCII-Codes oder „ähnliche“ Buchstaben (wie „O“ = „0“ oder „I“ = „1“) an das Modul übermittelt werden. Tabelle 1 gibt eine Übersicht der unterstützten Zeichen und der zugehörigen Befehlsbytes.

Zeichen	Befehlsbyte
0 = O	0x00, 0x30 („0“), 0x4F („O“)
1 = l = I	0x01, 0x31 („1“), 0x49 („l“), 0x6C („I“)
2	0x02, 0x32 („2“)
3	0x03, 0x33 („3“)
4	0x04, 0x34 („4“)
5 = S	0x05, 0x35 („5“), 0x53 („S“), 0x73 („s“)
6	0x06, 0x36 („6“)
7	0x07, 0x37 („7“)
8	0x08, 0x38 („8“)
9	0x09, 0x39 („9“)
A	0x10, 0x41 („A“), 0x61 („a“)
B	0x11, 0x42 („B“), 0x62 („b“)
C	0x12, 0x28 („(“), 0x43 („C“), 0x5B („[“)
c	0x63 („c“)
D	0x13, 0x44 („D“), 0x46 („d“)
E	0x14, 0x45 („E“)
e	0x65 („e“)

Zeichen	Befehlsbyte
F	0x15 („F“), 0x46 („f“)
G	0x47 („G“), 0x67 („g“)
H	0x48 („H“)
h	0x68 („h“)
J	0x4A („J“), 0x6A („j“)
L	0x4C („L“)
n	0x4E („N“), 0x6E („n“)
o	0x6F („o“)
P	0x50 („P“), 0x70 („p“)
q	0x51 („Q“), 0x71 („q“)
r	0x52 („R“), 0x72 („r“)
t	0x54 („T“), 0x74 („t“)
U	0x55 („U“), 0x75 („u“)
Y	0x59 („Y“)
-	0x2D („-“)
“	0x22 („““)
'	0x27 („'“)
_	0x5F („_“)
	0x78

Tab. 1: Darstellbare Zeichen

Das Kommando 0x78 (letzte Zeile in Tab. 1) gibt ein Leerzeichen aus.

Diese Zeichenkommandos werden immer in Vier-Byte-Paketen an das Display geschickt und werden von links nach rechts auf den vier Sieben-Segment-Anzeigen dargestellt.

Neben diesen Zeichenkommandos gibt es 11 Sonderbefehle, die aus einem Ein-Byte-Kommando bestehen, meist gefolgt von einem Ein-Byte-Parameter. Eine Übersicht dieser Sonderbefehle zeigt Tabelle 2.

Wert	Kommando	Parameter
0x76	<i>Clear Display</i>	-
0x77	<i>Decimal Points</i>	0x00-0x3F
0x79	<i>Move Cursor</i>	{0, 1, 2, 3}
0x7A	<i>Brightness</i>	{0, ..., 255}
0x7B	<i>Single Char 1</i>	0x00-0x7F

Wert	Kommando	Parameter
0x7C	<i>Single Char 2</i>	0x00-0x7F
0x7D	<i>Single Char 3</i>	0x00-0x7F
0x7E	<i>Single Char 4</i>	0x00-0x7F
0x7F	<i>Baud Rate</i>	0x00-0x0B
0x80	<i>Set I<sup>2</sup>C Address</i>	0x01-0x7E
0x81	<i>Factory Reset</i>	-

Tab. 2: Sonderbefehle

### Clear Display

Der Befehl 0x76 löscht das Display und setzt den Positions-Zähler auf 0 (Zeichen ganz links).

### Decimal Points

Anders als bei Philips' SAA1064 werden die Dezimalpunkte unabhängig von den ausgegebenen Zeichen gesetzt. Daher sind mehr als vier Dezimalpunkte möglich: So bietet das Display auch einen Doppelpunkt nach den ersten beiden Ziffern und ein Apostroph hinter der dritten Ziffer.

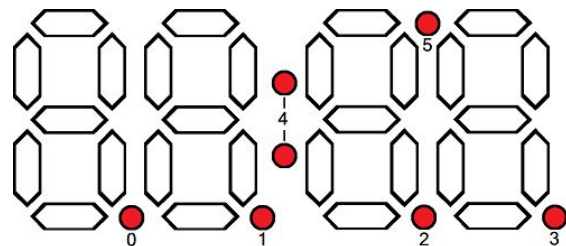


Abb. 4: Kodierung der Punkte  
(Quelle: [Sparkfun @ Github \[3\]](#))

Angesteuert werden die fünf Punkte durch ein einfaches Bitmuster: Die Dezimalpunkte werden in den Bits 0 bis 3 (links nach rechts) gesetzt, der Doppelpunkt in Bit 4 und das Apostroph in Bit 5 (Abb. 4). Zum Setzen bzw. Löschen der Punkte wird das Kommando 0x77 gefolgt vom Bitmuster-Byte an das Modul gesendet.

### Move Cursor

Der Befehl 0x79 gefolgt von einem Byte-wert aus {0, 1, 2, 3} setzt den Positions-zähler auf eines der vier Sieben-Segment-Anzeigen 0 bis 3.

## Brightness

Die Helligkeit des Displays wird mit dem Kommando 0x7A eingestellt, gefolgt von einem Byte-Wert von 0 (dunkel) bis 255 (maximale Helligkeit). Der aktuelle Helligkeitswert wird im nichtflüchtigen Speicher des Displays abgelegt und bleibt daher auch nach dem Ausschalten der Stromversorgung erhalten.

Bei den Helligkeitswerten 117 und 118 (0x76 und 0x77) gibt es offenbar einen Firmware-Bug: das Display wird ganz oder teilweise ausgeschaltet, da das Datenbyte offenbar als Befehlsbyte interpretiert wird. Diese beiden Helligkeitswerte sollte der Treiber daher nicht an das Display weitergeben.

## Single Char

Mit den Kommandos 0x7B, 0x7C, 0x7D und 0x7E können die LED-Segmente einer Anzeige – wie bei Philips' SAA1064 – einzeln gesetzt werden.

Dazu wird ein 7-Bit-Wert, der die sieben Segmente der Anzeige einzeln ein- bzw. ausschaltet, an das erste, zweite, dritte bzw. vierte LED-Anzeigemodul gesendet. Die Nummerierung der Segmente (Abb. 5) entspricht der des SAA1064. Alle vier Kommandos haben keine Auswirkung auf den Positionszähler.

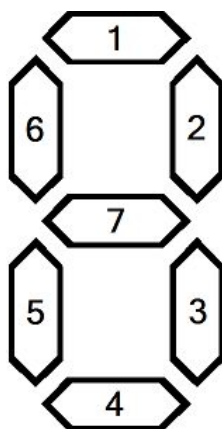


Abb. 5: Nummerierung der Segmente  
(Quelle: [Sparkfun @ Github](#) [3])

Vorteil dieser Ansteuerung: Kommt es zu einem Übertragungsfehler, ist lediglich eine

Anzeige betroffen – und es kommt beim nächsten Befehl nicht zu einem ‚Versatz‘ der Anzeige um eine oder mehrere Stellen (der sich allerdings durch die Positionierung mit dem Move Cursor-Kommando begrenzen lässt).

## Set I<sup>2</sup>C Address

Die I<sup>2</sup>C-Adresse des Moduls kann man mit dem Befehl 0x80, gefolgt von der neuen 7-Bit-Adresse (0x01 bis 0x7E) ändern. Achtung: Die neue Adresse unbedingt notieren, da es sonst nur die Möglichkeit gibt, an den gesamten Adressraum einen *Factory Reset*-Befehl (siehe folgender Absatz) zu senden und damit die Adresse auf 0x71 zurück zu setzen.

## Factory Reset

Der Befehl 0x81 setzt das Modul auf die Fabrikeinstellungen wie die ursprüngliche I<sup>2</sup>C-Adresse 0x71 und die maximale Display-Helligkeit zurück. Dabei erfolgt keine Löschung des Displays oder eine Zurücksetzung des Positionszählers.

## RoboPro-Treiber

Von Georg Stiegler gibt es eine Sammlung von [Robo Pro-Treibern](#) für das Sparkfun-LED-Display im Downloadbereich der ft:c. Darunter ist auch ein kleines Programm, das eine neue I<sup>2</sup>C-Adresse setzt – sowie eines, das bei der Suche nach einer vergessenen Adresse hilft, sprich: alle zulässigen Adressen (bis auf die des TX-EEPROMs) ausprobiert.

Der folgende, ebenfalls im Downloadbereich der ft-Community-Webseite zu findende ROBO Pro-Treiber [LED-S7SD](#) enthält einige Zusatzfunktionen, die die Nutzung des Displays für unterschiedliche Anwendungszwecke erleichtern:

- Die Funktion `S7SD_Init` löscht das Display und setzt den Positionszähler zurück (auf die linke Ziffer).
- `S7SD_MoveCurser` setzt den Display-Zeiger (Position 0-3).

- Die Display-Helligkeit setzt man mit der Funktion `S7SD_SetBrightness` (0-255).
- Die vier 7-Segment-Anzeigen lassen sich mit den Kommandos `S7SD_SetChar1` bis `S7SD_SetChar4` einzeln setzen. Dabei muss die korrekte Segmentkodierung übergeben werden. Für Zahlenwerte von 0-9 übernimmt das das Makro `BCD2LED`.
- Die Dezimalpunkte, der Apostroph und der Doppelpunkt werden mit dem Kommando `S7SD_SetDots` aktiviert.
- Die Funktion `S7SD_DisplayChar` gibt ein einzelnes Zeichen an einer ausgewählten Position aus. Übergeben wird eine Ziffer oder ein (gültiger) ASCII-Code.
- Die Funktion `S7SD_DisplayHex` gibt einen Integer-Wert als vierstellige Hexadezimalzahl auf dem Display aus.
- Die Funktion `S7SD_DisplayDezimal` gibt eine Dezimalzahl zwischen -1.000 und 10.000 mit bis zu drei Nachkommastellen aus.
- Die I<sup>2</sup>C-Adresse kann man mit dem Befehl `S7SD_SetI2CAddress` auf eine Adresse von 0x70 bis 0x77 ändern – acht Adressen sollten in der Praxis genügen, und die Beschränkung erleichtert dem Reset-Befehl die Arbeit (s. u.).
- Mit `S7SD_Reset` wird das Display zurückgesetzt; damit werden alle angeschlossenen Displays mit den Adressen 0x70 bis 0x77 auf die initiale I<sup>2</sup>C-Adresse 0x71 konfiguriert.

Die drei folgenden Funktionen erleichtern die Ausgabe einer BCD-kodierten Datums- und Zeitangabe, bspw. von einer Echtzeituhr [2], auf dem Display:

- Die Funktion `S7SD_DisplayTime` gibt eine BCD-kodierte Zeitangabe (Minuten, Sekunden) mit Doppelpunkt auf dem Display aus.

- Die Funktion `S7SD_DisplayDate` gibt eine BCD-kodierte Datumsangabe (Tag, Monat) mit Punkten auf dem Display aus.
- Die Funktion `S7SD_DisplayYear` gibt eine BCD-kodierte Jahreszahl (nur die beiden letzten Ziffern) auf dem Display aus.

Das Anzeigeprogramm wird damit sehr elegant und kurz (Abb. 6). In den Unterprogrammen sind lediglich die I<sup>2</sup>C-Adressen der drei Module anzupassen, da die I<sup>2</sup>C-Adresse in ROBO Pro leider nicht als Parameter übergeben werden kann.

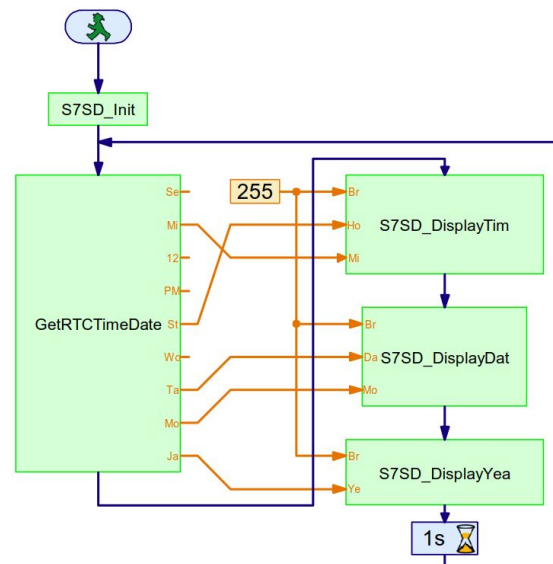
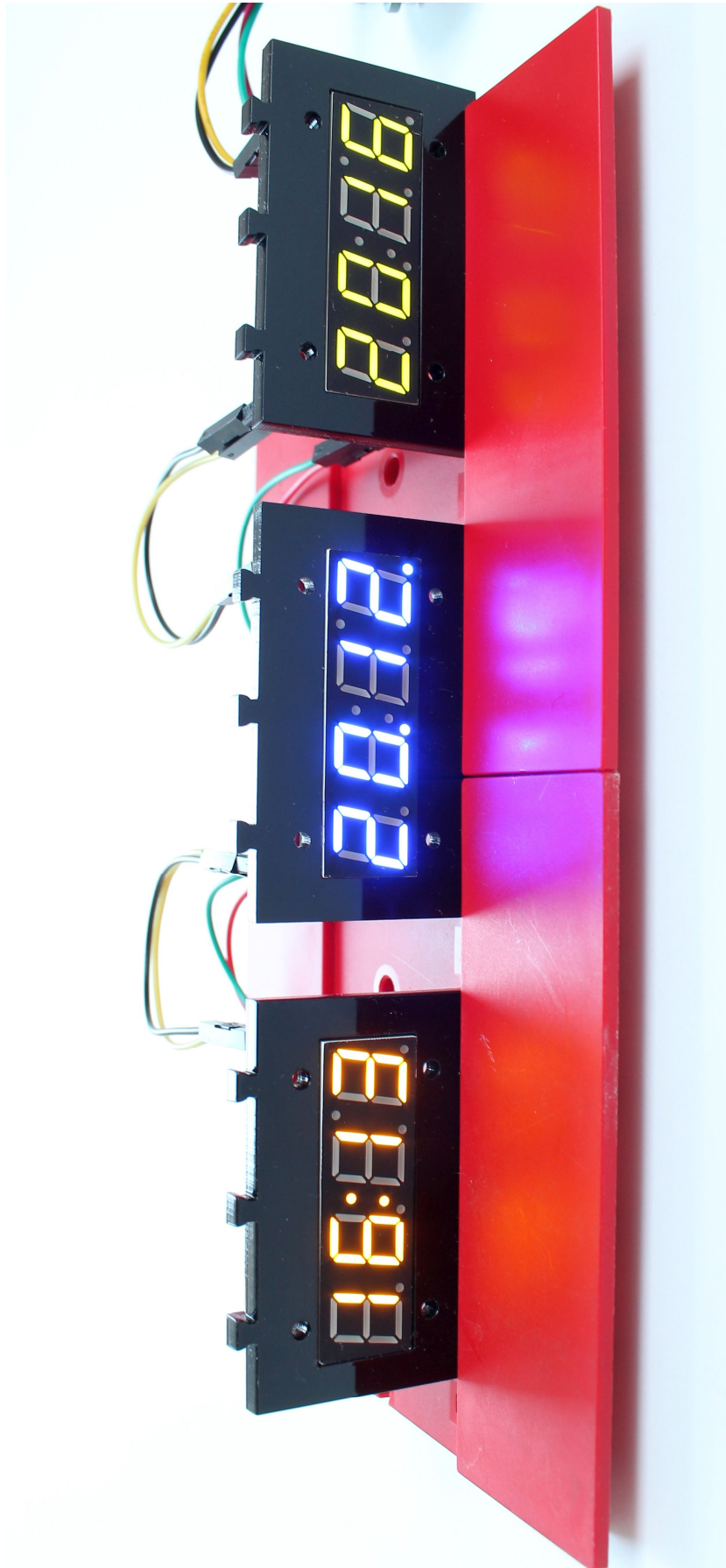


Abb. 6: ROBO Pro-Programm zur Anzeige von Datum und Uhrzeit

## Referenzen

- [1] Dirk Fox: *I<sup>2</sup>C mit dem TX – Teil 2: LED-Display*. [ft:pedia 4/2012](#), S. 32-37.
- [2] Dirk Fox: *I<sup>2</sup>C mit dem TX – Teil 7: Real Time Clock (RTC)*. [ft:pedia 4/2013](#), S. 28-34.
- [3] Sparkfun: [Serial 7 Segment Display Datasheet](#). Github.
- [4] Sparkfun: [Serial 7 Segment Display Datasheet](#). 17.05.2010.





*Abb. 7: Zeit- und Datumsanzeige mit drei 7-Segment-Displays (Sparkfun; Fenster von Georg Stiegler)*