

Editorial

Dynamik XXL

Respekt. Wer hätte das gedacht: fischertechnik, das Modellbausystem für Nerds und Technik-Freaks, wirtschaftlich längst abgehängt von Lego (mit Lego Technik und Mindstorms), gewinnt den Spielwaren-Oscar 2014 – mit einem Kasten der „Profi“-Serie: dem „Dynamic XL“. Ein Baukasten, der Anforderungen an Motorik und technisches Verständnis stellt, und dessen Modelle Geduld, Zeit und Feinjustierung erfordern, um zu funktionieren.

Da scheint irgendetwas nicht ganz zu stimmen mit dem Bild vom deutschen Xbox-Kinderzimmer. Zwar gibt es sie noch – die ebay-Verkäufer, die kaum bespielte fischertechnik-Kästen ihrer Kinder mit dem Argument feilbieten, jene seien inzwischen im „Zockeralter“; und die Kritiker, die dem „Dynamic XL“ vorwerfen, dass die konstruierte Kugelbahn nicht auf Antrieb funktioniert. Aber offenbar gewinnt eine Eltern-Fraktion an Bedeutung, bei denen Spielzeug mehr sein darf als ein beschäftigungstherapeutisches Hilfsmittel zur temporären Ruhigstellung des Nachwuchses.

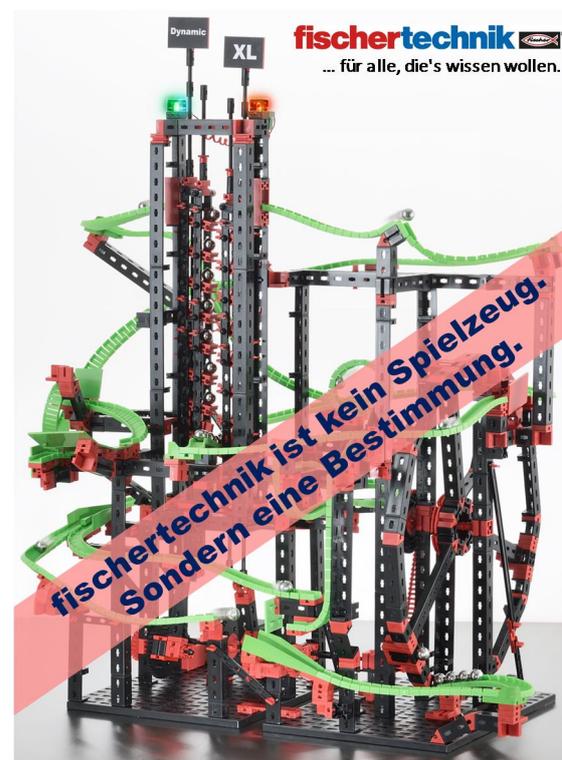
Bei Amazon erreichte der Kasten inzwischen in der Kategorie ‚Spielzeug‘ die TOP 1.000, obwohl sein Preis deutlich jenseits der 100 € bei vielen Erstkäufern zweifellos spontan Schnappatmung auslösen dürfte.

Abgezeichnet hat sich diese Entwicklung schon beim Vorgängerkasten „Dynamic“, der trotz seines Preises zum Verkaufsschlager wurde. Besonders erfreulich daran: Der Baukasten lässt Einsteiger das „System fischertechnik“ erkennen – Bausteine, die sich universell kombinieren lassen und

Dirk Fox, Stefan Falk

einem durchgängigen Grundmaß folgen, die vielseitige mechanische Konstruktionen erlauben und physikalisch-technische Zusammenhänge spielerisch erlebbar machen.

Die Einführung der Kugelbahnen könnte also eine anhaltende Begeisterungswelle für fischertechnik ausgelöst haben. Mit etwas Geschick sollte es fischertechnik gelingen, diese Welle zu reiten. Vielleicht wäre jetzt der richtige Zeitpunkt für eine pfiffige Plakat-Kampagne...



Und jetzt: Ran an die Kästen!

Beste Grüße, Euer ft:pedia-Team

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

Inhalt

Dynamik XXL	2
Die ‚neue fischertechnik‘ – 1989-1994	4
Mini-Modelle (Teil 3): Scheinwerfer	11
Mini-Modelle (Teil 4): Hubschrauber	12
Kaulquappen (Teil 5).....	14
Pneumatische Drehdurchführung.....	17
ft-Spezialteile made by TST (Teil 9).....	20
Druckluftsteuerungen (Teil 3).....	21
Druckmessung mit RoboTX Controller.....	32
Binäraddierer.....	36
Endliche Automaten in Robo Pro	42
Detail Engineering R2D3 (3) – IR-Fernbedienung am Robo TX Controller	51
LEDs in Leuchtsteinen	55
Parallel Interface – durch Arduino gesteuert (3).....	61
Strichcode-Leser am Robo TX Controller (1): Automatisiert mit RoboPro	66
Navigation	72

Termine

Was?	Wann?	Wo?
ft Convention 2014	27.09.2014	Erbes-Büdesheim
Science Days	16.-18.10.2014	Europapark Rust
Maker Space	20.-23.11.2014	Messe Stuttgart

Impressum

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

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

Autoren: Christian Andersch (Carrera124), Gerhard Birkenstock (gggb), Stefan Falk (steffalk), Dirk Fox (Dirk Fox), Johann Fox, Andreas Gail, Jens Lemkamp (Iemkajen), Thomas Püttmann (geometer), Harald Steinhaus (Harald), Andreas Tacke (TST), René Trapp (H.A.R.R.Y.)

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

ft-Geschichte

Die ‚neue fischertechnik‘ – 1989-1994

Christian Andersch

Vor ziemlich genau 50 Jahren erblickte fischertechnik im Herbst 1964, zunächst als Kunden-Weihnachtsgeschenk gedacht, das Licht der Welt. Der perfekte Zeitpunkt, um den Blick auf die Entwicklungsgeschichte der fischertechnik-Kästen zu richten. Der Beitrag beleuchtet die – für Sammler interessante und bei fischertechnik-Fans nicht unumstrittene – Phase der ‚neuen fischertechnik‘, die das einheitliche „grau-rot“ des Teilesortiments beendete.

Ich gebe es zu, in meiner Kindheit haben die Bauklötze aus Dänemark mein Kinderzimmer dominiert. Zusätzlich stand eine Carrera-Rennbahn im Hobbyraum, somit gab es zwei Systemspielzeuge, die an Geburtstagen und zu Weihnachten kontinuierlich erweitert wurden. Für fischertechnik war da kein Platz, sowohl im wörtlichen als auch im übertragenen Sinn.

Nichtsdestotrotz habe ich das fischertechnik-Sortiment immer mit einem Auge verfolgt. Es war Familientradition, in der Vorweihnachtszeit die Spielwarengeschäfte in der Stadt zu besuchen. Mit Vedes, Idee & Spiel, Spielzeug-Ring und weiteren lokalen Geschäften waren wir in den 80er Jahren noch gut versorgt. Ebenfalls war es gute Tradition, die Kataloge dieser Spielwarenketten mitzunehmen und zuhause intensiv durchzublättern. Im Gegensatz zu vielen anderen Haushalten wurden diese Kataloge aber nicht entsorgt sondern aufbewahrt, so dass ich hier eine nahezu durchgängige Kollektion angesammelt habe, beginnend mit den ausklingenden 70er Jahren.

Zurück zu fischertechnik. Das System der Grund- und Erweiterungsbaukästen der 80er Jahre fand ich als Kind immer etwas schwierig zu durchschauen. Die kleinen Führerhäuser und die Halma-ähnlichen Figuren sprachen mit ebenfalls nicht

wirklich an. Ganz anders verhielt es sich mit dem Modellbaukästen: Schwerlastkran, Hydraulik-Kipper, Universalfahrzeug und so weiter, damit hätte man mich durchaus begeistern können. Aus den oben erwähnten Gründen kam es leider nie dazu.

Die ‚neue fischertechnik‘

Die Jahre 1989 und 1990

Auch 1989 wurden Kataloge besorgt, so dass mir hier der Prospekt „Die neue fischertechnik“ in die Hände fiel. Alte Zöpfe wurden rigoros abgeschnitten, das Sortiment von Grund auf erneuert und völlig neu gegliedert. Besonders auffällig war die Abkehr vom rot-grauen Look hin zu mehr Farbe: Standardbausteine waren jetzt schwarz und Statikteile gelb. Die neue Aufteilung in Junior- und Master-Serie erschien mir sinnvoll. Als Grundbaukasten gedacht, hatten diese – im Gegensatz zu den früheren Start 100/200 Kästen – endlich auch große Führerhäuser und die dazugehörigen großen Figuren.

Los ging es mit drei Baukästen: Junior (30300), Master (30301) und Motor Master (30302). Ebenfalls neu war die Verpackung: Anstelle einer herkömmlichen Pappschachtel wurde eine Art „Werkzeugkoffer“ aus Kunststoff verwendet (Abb. 1). Hübsch anzusehen, aber in der Handhabung nicht ganz so praktisch wie gedacht, weil

der Sortiereinsatz nicht bündig mit dem Deckel abschließt: trägt man den Koffer am Tragegriff, purzeln innen kleinere Bauteile aus ihren Sortierfächern heraus und verteilen sich überall. Das war sicher nicht im Sinne des Erfinders.



Abb. 1: Der Junior-Koffer (30300), 1989

Weiterhin lässt sich bei den einzelnen Serien so etwas wie ein ‚Farbcode‘ feststellen: Der Junior-Koffer verfügte über ein rotes Fahrerhaus und gelbe Felgen. Ein Schema, das auch in nachfolgenden Kästen der Junior-Serie beibehalten wurde.

Der Master-Koffer (Abb. 2) war mit einem blauen Führerhaus und roten Felgen ausgestattet, die Teile für die LKW-Ladepritsche waren gelb. Auch dieses Schema wurde für nachfolgende Kästen der Master-Serie beibehalten.



Abb. 2: Master (30301), 1989

Abweichend davon war die Ausstattung des Motor Master (Abb. 3): Hier wurden nicht nur die zusätzlichen Teile für Motor und Getriebe beigelegt, sondern gegenüber dem nicht motorisierten Master auch die Farben

von Felgen und Ladepritsche getauscht. Vermutlich sollten diese Maßnahmen bewirken, dem Topmodell unter den Baukästen etwas mehr Eigenständigkeit zu verschaffen, anstatt einfach nur einen erweiterten Master-Baukasten darzustellen.



Abb. 3: Motor Master (30302), 1989

Auch wurden 1989 bereits die ‚Master Plus‘-Erweiterungen angekündigt und im Falle der Motorisierung 30310 auch bereits ausgeliefert (Abb. 4). 1990 folgten weitere Master Plus Kästen (Abb. 5).



Abb. 4: Master Plus – Motorisierung (30310)



Abb. 5: Master Plus – Figuren (30311)

Erste Umstellung 1991 – ab in den Karton

Den Kunststoffkoffern war nur ein relativ kurzes Leben beschieden, bereits 1991 erhielten die Grundbaukästen der Junior- und Master-Serie herkömmliche Pappschachteln mit tiefgezogenen Sortiereinsätzen aus Kunststoff (Abb. 6).



Abb. 6: Master (30301), 1991

Gemeinhin herrscht die Annahme, dass zeitgleich mit diesem Schritt auch die Umstellung beim Motor von 6 V auf 9 V erfolgte und der quadratische Power Block durch die rechteckige Version abgelöst wurde. Diese Annahme kann aber nicht zutreffen, denn im 1991er Katalog ist ein Motor-Master-Baukasten abgebildet, der trotz Kartonverpackung weiterhin den quadratischen Power Block sowie die gelben Felgen und die rote Ladepritsche auf dem Deckel zeigt, mithin also die identische Teilebestückung wie die Koffer-Version enthält.



Abb. 7: Motor Master (30302), 1991

Dabei handelt es sich nicht nur um eine Katalogabbildung, solche Kästen wurden

tatsächlich produziert und ausgeliefert (Abb. 7). Allerdings ist die Vermutung naheliegend, dass diese Version nur vergleichsweise kurzzeitig produziert wurde und heute entsprechend selten ist.

Profi Kästen und Special Sets ab 1991

Die Special Sets waren die Nachfolger der Service Sets aus den 80er Jahren. Motoren, Power Block und Standardbausteine konnten hier einzeln bzw. in kleinen Mengen erworben werden, insofern kann man diese Produktlinie als kleineren Bruder der Master-Plus-Ergänzungskästen betrachten.

Oberhalb der Master-Serie war die neue Profi-Serie angesiedelt, in der u. a. auch die Computing-Kästen eine neue Heimat fanden.

1992 – eine weitere Umstellung beim Motor Master

Abermals wurde der Motor Master überarbeitet, jetzt endlich erfolgte die Umstellung auf 9 V und den rechteckigen Power Block. Interessanterweise wurde nun auch die Farbe von Felgen und Ladepritsche an die Bestückung des nicht motorisierten Master angeglichen: rote Felgen und gelbe Pritsche, dies ist sowohl am Deckelbild als auch in der Anleitung sichtbar (Abb. 8).



Abb. 8: Motor Master (30302), 1992

Die Gründe für die Änderung dürften auf der Hand liegen: die Master-Plus-Ergänzungskästen, insbesondere Trucks (30314) und Special Trucks (30317), waren farblich auf den Master-Baukasten abgestimmt, die

daraus baubaren Modelle wirken harmonisch: einheitlich rote Felgen und gelbe Ladepritschen.

Ganz anders bei der Kombination mit einem der beiden frühen Motor-Master-Versionen: hier konnten die Modelle dann insgesamt nur mit gemischtfarbigen Rädern und bunt zusammengewürfelten Ladepritschen gebaut werden. Kein besonders glücklicher Umstand, erst recht nicht vor dem Hintergrund, dass der Motor Master ja eigentlich das Topmodell der Master-Serie darstellte, wozu solche farblichen Ungereimtheiten nun wirklich nicht passten.

Bemerkenswert ist weiterhin, dass sowohl auf dem Deckelbild als auch auf der Bauanleitung der etwas füllige junge Mann hinter dem Power Block wegetruschiert wurde (Abb. 8). Über die Gründe kann man nur spekulieren, möglicherweise waren dicke Kinder auf Produktverpackungen nicht mehr erwünscht.



Abb. 9: fischertechnik-Sortiment 1992
(aus: Katalog Idee & Spiel)

Schwächen des Junior- und Master-Systems

Die Grundidee der „neuen Fischertechnik“ ab 1989 bestand darin, nur ein vergleichsweise kleines Sortiment an Grundbaukästen anzubieten, die von der Kundschaft dann systematisch und nach Lust und Laune mit den Plus- und Special-Sets erweitert werden konnten (Abb. 9). Selbst der Aufstieg von der Junior- auf die Master-Serie war über eine entsprechende Erweiterungspackung namens Junior Plus problemlos möglich. In der Praxis hatte dieses eigentlich so klare und übersichtliche Konzept aber doch seine Tücken.

Zum einen war nicht jedem Kunden – beispielsweise den Großeltern beim Weihnachtsgeschenke-Einkauf – klar, dass die Master-Plus-Kästen nur zur Ergänzung gedacht waren und beim Beschenken das Vorhandensein eines Master- oder Motor-Master-Grundbaukastens voraussetzten. Recht schnell wurden deshalb die Master-Plus-Kästen mit relativ großformatigen Aufklebern nachgerüstet, die auf diesen Umstand hinwiesen (Abb. 10).



Abb. 10: Hinweis auf Master-Plus-Kästen

Zum anderen gab es diverse ‚strategisch wichtige‘ Teile, die nur in den Grundbaukästen, nicht aber in den Plus-Ergänzungskästen enthalten waren. Exemplarisch sei hier das Führerhaus genannt; der gleichzeitige Bau eines Lastwagens und eines Ladekrans jeweils mit Führerhaus oder von zwei Lastwagen war somit nicht möglich.

Lenkungsteile und Räder waren ebenfalls Mangelware; letztere waren neben den Grundbaukästen immerhin in den beiden bereits erwähnten Master-Plus-Kästen

Trucks (30314) und Special Trucks (30317) enthalten (Abb. 11). Das Problem mit dem Führerhaus wurde 1992 gelöst, als dieses einzeln in Form eines Special Sets erworben werden konnte. Nichtsdestotrotz dürfte der Erwerb eines zweiten Master-Baukastens manchmal der einfachste (und einzige) Weg gewesen sein, um an weitere Vorräte wichtiger Bauteile zu kommen.

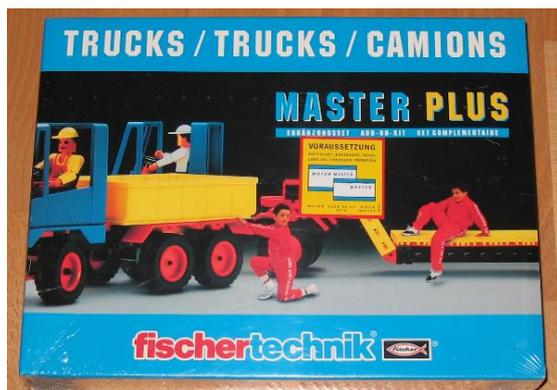


Abb. 11: Master Plus Trucks (30314)

Zu guter Letzt konnte der Wegfall der imposanten Modellbaukästen aus den 80er Jahren mit dem modularen System der Master-Serie nie so richtig kompensiert werden. Die stabilen Aluprofile waren weder in den Master-Grundbaukästen noch in den Master-Plus-Ergänzungskästen enthalten, wengleich man mit Hilfe der Master-Plus-Kästen durchaus beeindruckende Modelle bauen konnte, die den früheren Modellbaukästen nahe kamen.

Die Action Sets



Abb. 12: Action Set – Flying Circus (30431)

Ab 1992 wurden zusätzlich kleinere Modelle als sogenannte Action-Sets vermarktet

und relativ preisgünstig angeboten (Abb. 12). Ich persönlich vermute, dass damit zwei Ziele erreicht werden sollten:

- Erstens stellten sie einen sehr günstigen Einstieg in die Fischertechnik-Welt dar.

Ohne gleich einen relativ teuren Master-Grundbaukasten erwerben zu müssen, konnte man Fischertechnik ‚antesten‘. Hinzu kam, dass die Modelle durchaus attraktiv waren. Allerdings wurde das Rad nicht neu erfunden, zumindest in den ersten Jahren handelte es sich ausnahmslos um Modelle, die bereits aus den Anleitungen des Master-Baukastens bzw. der Master-Plus-Ergänzungskästen bekannt waren.

- Zweitens waren die Modelle auch für Besitzer eines Master Grundbaukasten interessant, denn sie enthielten die bereits weiter oben erwähnten ‚Mangelteile‘, die es ansonsten nur in den großen Grundbaukästen gab.

Weitere Entwicklung ab 1993

Stillstand ist Rückschritt, das mussten die Gedanken zu dieser Zeit gewesen sein. Seit vier Jahren war das neue Konzept am Markt, und das große Feuerwerk in Form diverser Master-Plus-Ergänzungskästen und Special Sets war bereits verschossen.

Erstmals wurden die Master Serie substantiell erweitert, indem man mit ihr mit dem Master-Discovery-Kasten ein neues Mitglied zur Seite stellte (Abb. 13), sogar in zweifacher Ausfertigung: wahlweise mit oder ohne Motor.



Abb. 13: Master Plus – Discovery Set (30299)

Wie üblich, dachte man auch an die Master-Besitzer und gibt ihnen mit dem Master-Plus-Discovery-Kasten die Möglichkeit, ihr vorhandenes Material auf den Umfang des neuen Master-Discovery-Grundbaukastens zu erweitern. Gewöhnungsbedürftig war allerdings die giftgrüne Farbgebung mancher Teile des Master-Discovery-Grundbaukastens.

1994 erschien mit dem Master Plus Galaxy noch ein Ergänzungskasten, der speziell für den Master Discovery konzipiert war (Abb. 14). Es sollte die letzte Neuheit für die 1989 begonnene Junior- und Master-Serie sein.



Abb. 14: Master Plus – Galaxy (30322)

In den Folgejahren wurden lediglich die Profi-Serie noch weiter ausgebaut; andere Neuheiten erschienen ohne explizite namentliche Zuordnung zu einer Serie.

Erst 1998 erschien wieder ein neuer Junior-Kasten, allerdings in anderer optischer Aufmachung und ohne erkennbare Zugehörigkeit zur ursprünglichen Junior-Serie.

Die heutige Situation aus Sammlersicht

Letztlich erstreckt sich die betrachtete Periode auf die Jahre 1989 bis 1994. Auf dem Gebrauchtmrkt sind nahezu alle Kästen der Junior- und Master-Serie aus dieser Epoche zu niedrigen Preisen verfügbar. Allerdings habe ich persönlich leider noch keinen Gebrauchtkauf getätigt, bei dem wirklich alle Teile komplett vorhanden waren: irgendetwas fehlte immer.

Mit etwas Geduld sind auch ungeöffnete Kästen zu bekommen, selbst bei den frühen Versionen im Werkzeugkoffer ist das nicht illusorisch. Die Preise sind meiner persönlichen Meinung nach als moderat zu bezeichnen, wenn man als Referenz die erzielten Preise für ungeöffnete Baukästen aus dänischer Produktion aus dem gleichen Zeitraum 1989-1995 heranzieht. Ähnlich ist die Lage bei den Master-Plus-Kästen, mit Ausnahme der beiden größten (und damals entsprechend teuren) Exemplare mit dem Zusatztitel ‚Top Edition‘.¹

Uneinheitlich sieht die Sache bei den Special Sets aus: normale Bausteine sind durchaus in ungeöffneter Originalverpackung zu bekommen, wohingegen die elektrischen Komponenten wie Motoren, Power Supply, Power Controller und Power Block als lose Gebrauchtware eher selten anzutreffen sind.

¹ Auch diese Kästen werden hoffentlich irgendwann den Weg in meine Sammlung finden. Und wer eine ungeöffnete Packung Special Trucks

(30317) abzugeben hat, ist ebenfalls willkommen.

Gruppe	Art.-Nr.	Name	Weitere Informationen	1989	1990	1991	1992	1993	1994	1995
Junior	30300	Junior	Koffer							
			Schachtel							
	30303	Super Junior	Koffer							
			Schachtel							
	30304	Little Junior								
	30305	Junior Plus								
30306	Big Junior									
Master	30301	Master	Koffer							
			Schachtel							
	30302	Motor Master	Koffer, Version mit 6 Volt							
			Schachtel, Version mit 6 Volt							
			Schachtel, Version mit 9 Volt							
	30307	Master Discovery								
	30309	Master Discovery mit Motor								
30327	Super Master Extra Plus 4									
95831	Master Discovery					?	?	?		
Master Plus	30299	Discovery								
	30310	Motor	Version mit 6 Volt							
			Version mit 9 Volt							
	30311	Figuren								
	30312	Raupen								
	30313	Rummelplatz								
	30314	Trucks								
	30315	Highlights								
	30316	Starlifters	Top Edition							
	30317	Special Trucks								
	30318	Adventure Park	Top Edition							
	30320	Sky Hoppers								
	30321	Heli Stars								
	30322	Galaxy								
Action	30328	Offroader								
	30331	Grasshoppers								
	30332	Gabelstapler								
	30334	Kleines Riesenrad								
	30430	Bergungsraupe								
	30431	Flying Circus								
	30439	Gabelstapler								
	95868	Bulldozer								
Special	30180	Power Supply								
	30181	Power Controller								
	30259	Bauteile								
	30261	Motor M								
	30262	Motor S								
	30263	Power Block								
	30264	Lampen								
	30265	Bauteile (Bogenstücke)								
	30266	Bauteile (Winkelträger)								
30267	Bauteile (Führerhaus)									
Profi	30490	Computing								
	30491	Sensoric								
	30492	Cartech								
	30493	Bionic								
	30494	Pneumatic								
Bi-Fi	30507	Drachen	Werbemodell							
	30508	Flugzeug	Werbemodell							
	30509	Tribike	Werbemodell							
	30510	Strandsegler	Werbemodell							
	30511	Hobbycar	Werbemodell							

Tabelle: Übersicht der fischertechnik-Kästen 1989-1995
(Quellen: [Baukastenübersicht von Karl Tillmetz](#) (Stand 2011),
eigene Sammlung und Spielwarenkataloge)

Modell

Mini-Modelle (Teil 3): Scheinwerfer

René Trapp

Das dritte Modell in der Reihe ‚Mini-Modelle im GiveAway-Format‘ bringt Licht ins Dunkel.

So langsam nähert sich die dunkle Jahreszeit. Die Tage werden kürzer und abends ist eine Lichtquelle gefragt. Warum nicht mal eine Designerleuchte aus der heimischen ft-Werkstatt?

Hier ist der Mini-Scheinwerfer, der in alle Richtungen gedreht und auch vertikal geschwenkt werden kann (Abb. 1 bis 3).

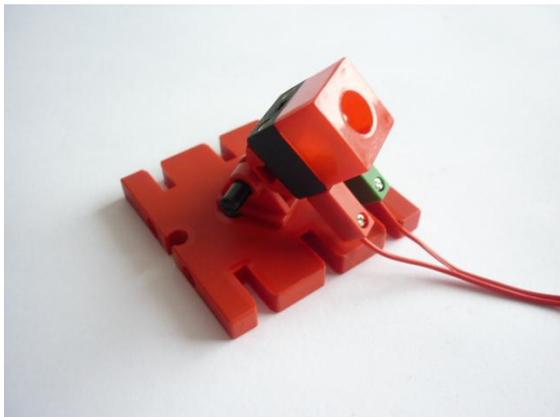


Abb. 1: Mini-Scheinwerfer

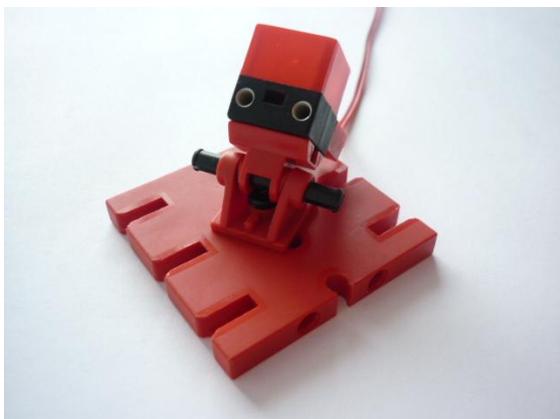


Abb. 2: Die ‚Schattenseite‘



Abb. 3: Das Fußgelenk

Der Fuß kann natürlich angepasst werden, zum Beispiel mit einem Lagerstück 1 (31771) anstelle von Bauplatte und V-Stopfen.

Und hier noch die komplette Stückliste:

Stück	ft-Nr.	Bezeichnung
1	36593	Bauplatte 45x45x5,5
1	32316	Verbindungsstopfen
1	31772	Lagerstück 2
1	36919	V-Achse 28
1	31426	Gelenkwürfel Zunge
1	38216	Leuchtstein
1	37875	Linsenlampe 9V
1	31321	Störlichtkappe, 8mm
2	31336	Flachstecker, grün
2	31337	Flachstecker, rot
1	36977	Litze, 30cm

Modell

Mini-Modelle (Teil 4): Hubschrauber

Johann Fox

Und gleich noch ein Modell im GiveAway-Format für die Minimodell-Sammlung: ein Mini-Hubschrauber aus lediglich 17 Teilen.

Immer wieder wurde sich an fischertechnik-Hubschraubern aller Art versucht, doch nie ist einer abgehoben. Das gelingt meinem Mini-Hubschrauber leider auch nicht, dafür besteht er nur aus 17 Bauteilen. Abbildung 1 zeigt ihn in der Gesamtansicht.



Abb. 1: Gesamtansicht

Das Kernstück des Hubschraubers ist ein Rollenlager (37636), das den Heckrotor, den Hauptrotor, das Cockpit und die Kufen zusammenhält.

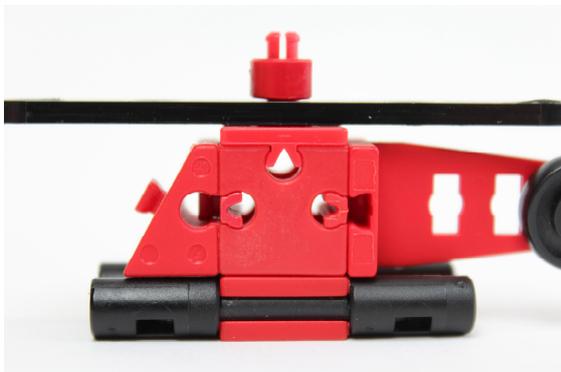


Abb. 2: Rollenlager als Kernstück des Hubschraubers

Den Hauptrotor bildet eine schwarze I-Strebe 60 mit Loch (36952), den Heckrotor ein Rad 14 (36573), das mit einem Zylinder-Anschlussstück (31719) und einem Verbindungsstopfen (32316) am zentralen Rollenlager festgemacht wird.



Abb. 3: Haupt- und Heckrotor

Die Kufen bestehen aus je einer Rastachse 30 (35063) und je zwei Rastkupplungen (35073). Befestigt an einem Baustein 7,5 (37468) bilden sie den Unterbau des Hubschraubers.



Abb. 4: Die Kufen

Das Cockpit ist ein Winkelstein 30 Grad rechtwinklig (31012).

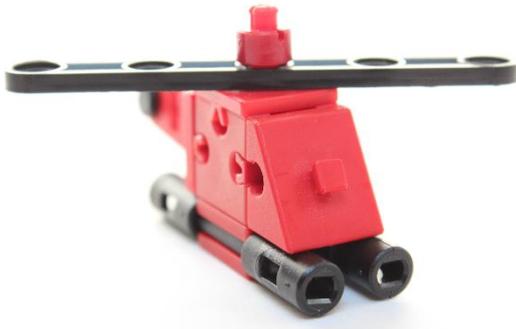


Abb. 5: Das Cockpit

Insgesamt besteht der Hubschrauber aus den in Abb. 6 gezeigten 17 Einzelteilen.

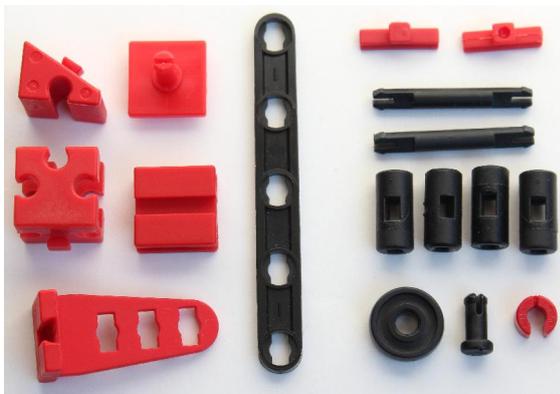


Abb. 6: Einzelteile des Hubschraubers

Die folgende Tabelle enthält die Einzelteil-
liste für alle, die den Hubschrauber gerne
nachbauen möchten.

Stück	ft-Nr.	Bezeichnung
1	37468	Baustein 7,5
1	37636	Rollenlager
1	31124	Aufnahmeachse
1	31012	Winkelstein 30 Grad rechtwinklig
1	31719	Zylinder Anschlussstück
1	36952	I-Strebe 60 mit Loch schwarz
1	36573	Rad 14 schwarz

Stück	ft-Nr.	Bezeichnung
1	32316	Verbindungs- stopfen
1	37679	Klemmbuchse 5
2	31982	Federnocken
2	35063	Rastachse 30
4	35073	Rastkupplung

Bisher erschienen:

- [1] René Trapp: *Minimodelle (Teil 1): Gabelstapler*. [ft:pedia 4/2013](#), S. 4-5.
- [2] Johann Fox: *Minimodelle (Teil 2): Panzer*. [ft:pedia 2/2014](#), S. 18-19.
- [3] René Trapp: *Minimodelle (Teil 3): Scheinwerfer*. [ft:pedia 3/2014](#), S. 8.

Tipps & Tricks

Kaulquappen (Teil 5)

Harald Steinhaus

Einen kleinen Vorteil hat so ein nasser verregneter Sommer wie der jetzige schon: es gibt genügend Zeit, sich mit Kaulquappen (siehe auch Teil 1-4 [1]) zu befassen und dafür zu sorgen, dass der eine oder andere Frosch daraus hervor geht, den man durch den Kuss einer Muse zum Prinzen verwandeln kann. Trotz alledem, mit echtem Sonnenlicht würde zumindest das Fotografieren leichter fallen.

Die Lok-Grundplatte 31557 der Bauspielbahn eignet sich als Träger für ein Winkelgetriebe (Abb. 1).

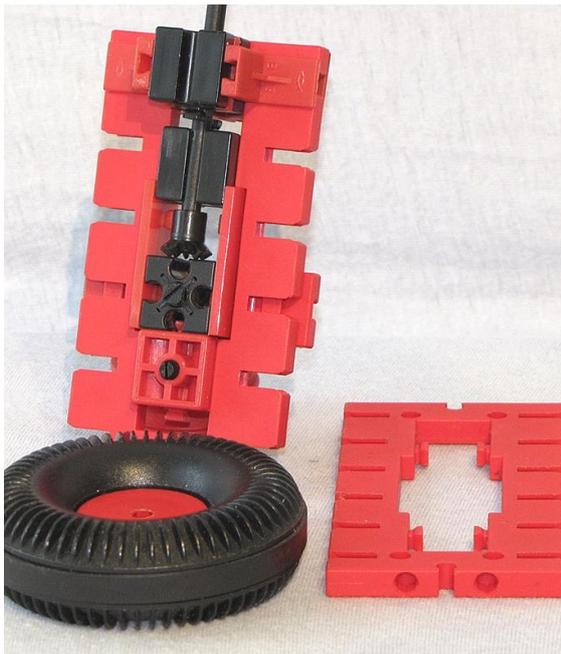


Abb. 1: Winkelgetriebeträger

Mit dem Reifen 60 auf Position kann das z. B. zum Obst- oder Weinbautraktor werden, der wie auf beräderten Stelzen über eine Reihe von Büschen oder Reben fährt (Abb. 2).

Durch Umschlingen runder Bauteile mit der schwarzen Kette erhält man neue Zahnräder (Abb. 3). Das große Innenzahnrad wurde außen etwas dünner gedreht, damit die Kette darauf passte, und ergibt ein Z46.



Abb. 2: Ein Traubenvollernter; Foto von Erhard Kühner ([wikipedia](#))

Die Drehscheibe 60 ergibt ein Z42.



Abb. 3: Z46 (aus Innenzahnrad) und Z42 (aus Drehscheibe 60)

Kupplungen: Die Achse links in Abb. 4 mit dem Z30 (die rote Kette mit den kurzen Stiften geht hier genauso gut) kann man auf der Freilaufnabe verschieben, bis die Stifte in die Ausnehmungen der Drehscheibe einfädeln und beide Elemente drehstarr verbinden. Das gleiche geht mit dem Z15 auf der rechten Seite, das auf einer Seilrolle beruht.

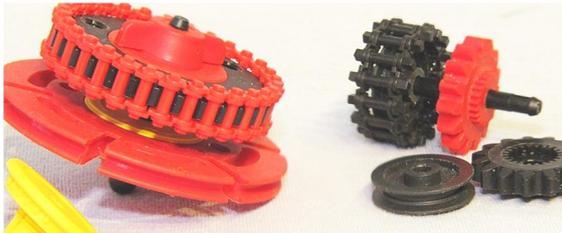


Abb. 4: Kupplungen mit Z30 und Z15

In der Kette auf dem rechten Z30 in Abb. 5 sind nur fünf schwarze Kettenglieder verbaut. Das ergibt ein Schrittschaltwerk: mit jeder Umdrehung des rechten Zahnrads (nur gegen den Uhrzeigersinn) wird das linke Zahnrad um drei Zahnpositionen weit mitgenommen.



Abb. 5: Schrittschaltwerk

Mit den Teilen eines schwarzen Differenzials erhält man ein Z14, das starr mit dem Z20 verbunden ist, weil auch hier die Stifte in Ausnehmungen des Zahnrads passen. Rechts ein Z9 mit zwei Gelenkmuttern 35019, das aber nur lose aufliegt (Abb. 6).



Abb. 6: Z14 und Z9 (aus Differentialteilen)

Die Stifte an den schwarzen Kettengliedern machen so ein Zahnrad zum Kronenrad, mit dem man Drehbewegung auch im Winkel übertragen kann. Als Innenzahnrad funktioniert so ein Ketten-umschlungenes Zahnrad ebenfalls: darauf beruhen die [Hypozykloiden-Getriebe](#) im Bilderpool.



Abb. 7: Z20 als Kronenrad

Um einen Container anzuheben, braucht man ein Containerschloss. Im einfachsten Fall ist das ein T-förmiger Hebel am Greifer, der in einem Langloch am Container längs (offen) oder quer (geschlossen) gedreht werden kann. Diese T-Form haben die ft-Statikriegel, und es fehlt nur noch der Übergang auf eine Achse, um sie per Hebel oder Motor zu drehen (Abb. 8).



Abb. 8: Containerschloss mit S-Riegel

Mit dem Lenkhebel (38473) kann man eine Achse genau auf der Verlängerung der Diagonalen des BS30 lagern.

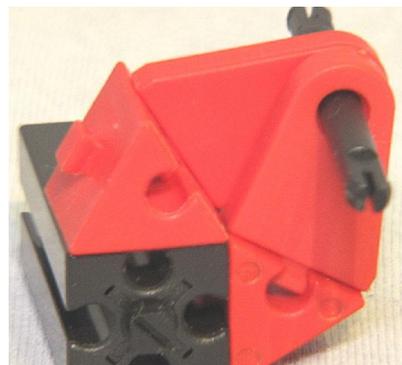


Abb. 9: Achslagerung mit Lenkhebel

Mit der gelenkig gelagerten Achse aus Abb. 10 kann man den Taster aus der Ferne betätigen. Gegebenenfalls muss die Achse etwas aufgepolstert werden, denn die ft-Taster streuen recht stark bezüglich der Auslösepunkte.

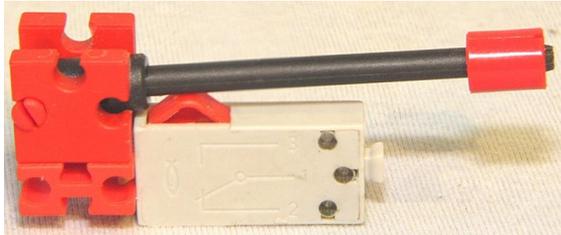


Abb. 10: Hebel zur Tasterbetätigung

Der Anlenkhebel (31894) wurde gemoddet und hat jetzt einen sauber runden Kugelkopf. Dieser gleitet in der Längsnut des BS30. Zieht man die gelbe Strebe in Abb. 11 nach links und rechts, schwenkt der BS30 nach hinten bzw. vorne.

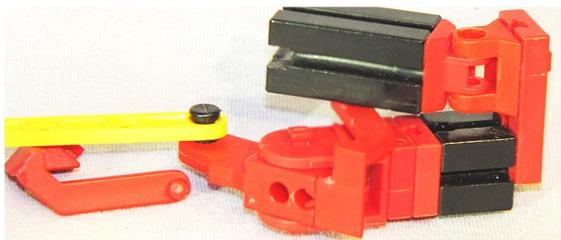


Abb. 11: Modifizierter Anlenkhebel

Der Prinz aus Abb. 12 ist eine Servolenkung: Man denke sich die gelbe Strebe als Spurstange einer Vorderachse und daran einen ft-Servo, sowie an der Laufrolle einen Hebel, den ein „ordentlicher“ Motor bewegt. Die Spurstange bewegt die beiden BS15-Lochsteine in Soll-Position und nimmt dabei die beiden Taster mit, soweit es deren Lagerung mit Gelenk und Gummiring zulässt. Ein schwarzes Kettenglied, das auf eine alte Seilrolle gesteckt ist, liefert die Ist-Position des Vorderrads. Beim Lenkeinschlag weichen Soll und Ist voneinander ab, einer der Taster ist gedrückt, und der Gummiring erlaubt die mechanische Fehlstellung. Der gedrückte Taster steuert den Lenkmotor solange an, bis das gelenkte Rad wieder mit den BS15 fluchtet.

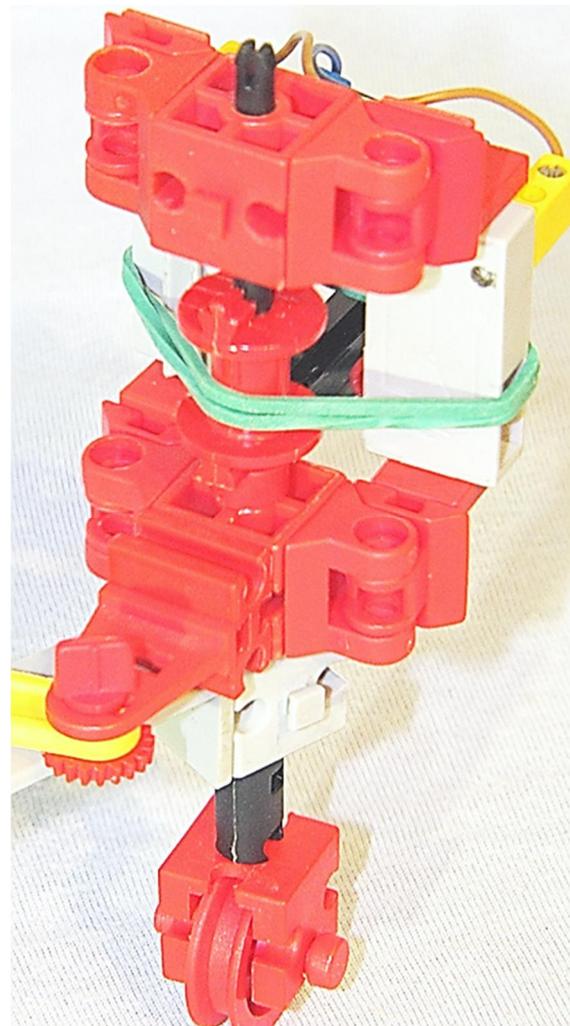


Abb. 12: Servolenkung

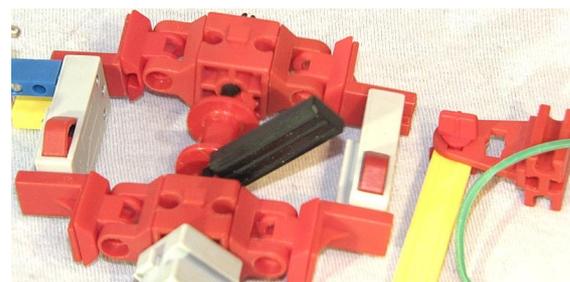


Abb. 13: Die Servolenkung, aufgeklappt.

Referenzen

- [1] Harald Steinhaus: *Kaulquappen*.
 Teil 1: [ft:pedia 1/2011](http://ft.pedia 1/2011), S. 22-28;
 Teil 2: [ft:pedia 2/2011](http://ft.pedia 2/2011), S. 9-13;
 Teil 3: [ft:pedia 3/2012](http://ft.pedia 3/2012), S. 24-26;
 Teil 4: [ft:pedia 1/2014](http://ft.pedia 1/2014), S. 21-23.

Tipps & Tricks

Pneumatische Drehdurchführung

René Trapp

Via Schleifring kann man drehende Motoren und Lampen mit Strom versorgen – wie aber bekommt man Luftdruck auf ein rotierendes Bauteil? Der Beitrag zeigt, wie sich eine voll funktionsfähige pneumatische Drehdurchführung mit Hausmitteln aus Zylindereinzelnteilen und einem Messingrohr konstruieren lässt.

Zur Übertragung elektrischer Energie auf rotierende Teile gibt es bekanntlich die Schleifringe. Die sind zwar nur noch gebraucht zu bekommen, aber immerhin hatte sich fischertechnik des Problems einmal angenommen. Für die Pneumatik fehlt solch ein Teil jedoch und es ist auch keine Lösung aus dem Schwarzwald in Sicht. Hier ein Vorschlag, wie ihr euch im Eigenbau eine voll funktionsfähige pneumatische Drehdurchführung aus ft-Zylindereinzelnteilen und einem Messingrohr konstruieren könnt.

Die Idee

Die Idee entstand beim Umgang mit Pneumatik-Zylindern. Die Kolbenstange lässt sich bekanntlich drehen – damit ist klar, dass die Dichtungen sich prinzipiell auch für eine Drehdurchführung eignen müssen.

Die Kolbenstange muss dabei durch ein Rohr ersetzt werden, das es (nicht nur) im Baumarkt gibt. Als einziges ‚echtes Problem‘ blieb, die Zylindereinzelnteile zu bekommen.

Die Beschaffung

Eine Anfrage bei den Fischer-Werken ergab leider eine Absage. Dafür war unser fischerfriendsman hingegen sicher froh, seinen sonst unverkäuflichen Zylinderschrott loszuwerden – zumindest die besseren Teile davon. An dieser Stelle mal

einen dicken herzlichen Dank an ihn. Wer in seiner Sammlung einen oder gar mehrere defekte Zylinder hat, kann natürlich diese als ‚Teilespender‘ verwenden.

Das Messingrohr ist Standardware aus dem einschlägigen Fachhandel (Modellbaushop, Baumarkt, ...). Der Außendurchmesser muss 4 mm betragen, für den Innendurchmesser empfehle ich 3 mm.

Die Fertigung

In dem gelieferten Zylinderschrott fanden sich alle benötigten Teile: Zylinderköpfe, Zylinderfüße, Zylinderrohre, Kolbenstangen und Dichtungspakete. Von einem blauen Pneumatikzylinder war der Fuß mit angeklebtem Rohr vorhanden. Es entsprach damit dem neuen Ballonadapter (122647).



Abb. 1: Ballonadapter (122647) [1]

Am oberen (offenen) Ende war das Zylinderrohr allerdings gesplittert. Da eine Drehdurchführung keinen Hub machen

muss, sägte ich den zersplitterten Teil einfach ab. Dabei achtete ich darauf, das Rohr auf 14 mm zu kürzen, sodass die fertige Durchführung im ft-Raster verbaut werden kann.

Einen einzelnen blauen Zylinderkopf säuberte ich mit einer Schlüsselfeile vorsichtig am Klebekragen, da die Reste eines Zylinderrohres und des werksseitigen Klebers sehr hart sind. Der Kragen an den Zylinderköpfen und -füßen selbst ist hingegen recht dünn und weich.

Nachdem der Zylinder so vorbereitet war, zog ich ein Dichtungspaket von einer Kolbenstange. Das Dichtungspaket, bestehend aus zwei Gummilippendichtungen und einem roten Stabilisatorring aus Kunststoff, wurde dann auf ein Stück Messingrohr (Außendurchmesser 4 mm, Innendurchmesser 3 mm) aufgeschoben.

Durch die Klemmwirkung ist bereits ein guter Sitz gewährleistet. Abb. 2 zeigt die vorbereiteten Hälften. Die Klemmbuchse 5 (37679) verhindert Hubbewegungen, kann aber auch weggelassen werden.



Abb. 2: Drehdurchführung offen

Das Dichtungspaket wurde vorsichtig in das kurze Rohrende eingefädelt, der Zylinderkopf aufgesetzt, jedoch nicht verklebt.

Damit erhält man einen Zylinder 30, der zwar nur bestenfalls (ohne die Klemmbuchse) 5 mm Hub macht, aber als pneumatische Drehdurchführung sehr gute Dienste leistet (Abb. 3).



Abb. 3: Fertige Drehdurchführung

Solange der Zylinderkopf nicht mit dem kurzen Zylinderrohr verklebt ist, kann die Durchführung jederzeit z. B. für Wartungszwecke zerlegt werden.

Nun wird noch eine geeignete Montage der Drehdurchführung benötigt. Besonders elegant ist die Unterbringung in einem Schleifring, damit man gleichzeitig elektrische Energie oder Signale übertragen kann; Abb. 4 zeigt ein Beispiel.



Abb. 4: Drehdurchführung am Schleifring

Ist der Schlauchstutzen am Zylinderkopf (Abb. 5) bei der Montage im Weg, darf er problemlos gekürzt oder auch ganz abgeschnitten werden.

Mögliche Modifikationen

Natürlich darf die Drehdurchführung auch länger sein (z. B. unter Verwendung eines alten Zylinders 60). So wird es möglich, die Durchführung auch als Längenausgleichselement zu nutzen.



Abb. 5: Der (hier nutzlose) Schlauchstutzen

Ein simpler gebogener Schlauch ist dafür allerdings kostengünstiger, sofern keine Rotation benötigt wird.

Baut man am anderen Ende des Messingrohres einen zweiten Zylinder in gleicher Weise an, entsteht ein Zylinder mit doppelter Hublänge (aber auch doppelter Baulänge). Die jeweils andere Zylinderkammer wird dann durch das Messingrohr mit Druckluft beaufschlagt und benötigt keinen eigenen Schlauchanschluss.

Bekanntlich haben die neuen Pneumatikzylinder eine Dichtung im schwarzen Zylinderkopf, um die Kolbenstangendurchführung abzudichten. Auch diese Zylinder lassen sich in eine Drehdurchführung verwandeln. Dazu werden nicht einmal die Dichtungspakete benötigt. Lediglich das Messingrohr wird durch den Zylinderkopf in den Druckraum geführt. Allerdings ist bei dieser Variante das Zylinderrohr am Zylinderfuß und am Zylinderkopf dicht zu verkleben. Eine Wartung ist hierbei nur bedingt möglich. Diese Variante habe ich allerdings nicht getestet.

Quellen

- [1] [www.ft-Datenbank.de: Ballonadapter 122647](http://www.ft-Datenbank.de:Ballonadapter%20122647).

Tipps & Tricks

ft-Spezialteile made by TST (Teil 9)

Andreas Tacke

In einer lockeren Reihe stellt TST einige von ihm entwickelte Spezialteile vor, die so manche Lücke beim Bauen mit fischertechnik schließen. In diesem Beitrag geht es um Pneumatik, genauer gesagt um eine Alternative zu den Drosselventilen aus den 80er Jahren.

Wer kennt sie noch, die Drosselventile (36077), die in den Pneumatik Kästen der 80er waren? Und: Wer weiß, was der Einsatzzweck einer Drossel ist?

Eine Drossel dient dazu den Luftstrom zu reduzieren. Somit lässt sich z. B. ein Pneumatikzylinder in seiner Aus- oder Einfahrgeschwindigkeit steuern [1]. Durch die Verengung des Leitungsquerschnitts wird der Luftstrom reduziert. In der Industrie werden solche Drosseln sehr häufig eingesetzt.

Die Drosselventile 36077 sind super, aber leider auch rar geworden. Welche Alternative gibt es also?

Peter Damen hat mich mit seiner Idee, eine Lüsterklemme zu nehmen (Abb. 1), auf eine Idee gebracht.



Abb. 1: Lüsterklemme als Drossel [2]

Ich nehme statt der Lüsterklemmen einen Baustein 15 mit Bohrung (32064) und versehe ihn mit einer zusätzlichen 3 mm-Bohrung von der Nut zur 4-mm Bohrung.

Dann wird in die Nut ein Stück Messingwelle mit einem M3-Gewinde eingesetzt. Jetzt kommt noch eine Rändelschraube M3 x 10 hinein – und fertig ist unsere Drossel im ft-Lock (Abb. 2).



Abb. 2: links: selbstgebaute Drossel, rechts die Original-ft-Drossel (36077)

Damit erhält man eine Drossel, die sich perfekt in das fischertechnik-System integrieren lässt. Des Weiteren bracht man, wie bei dem Original, kein Werkzeug, um sie einzustellen.

Und wieder mal zeigt sich, dass durch eine tolle Idee ein noch tollereres Bauteil entstehen kann...

Referenzen

- [1] Stefan Falk: *Druckluftsteuerungen (Teil 1)*, [ft:pedia 1/2014](#), S. 58
- [2] Peter Damen: [Drosselventil-Alternative](#), [ftcommunity.de](#)

Pneumatik

Druckluftsteuerungen (Teil 3)

Stefan Falk

Nach Schwellwertschalter und Zeitglied im letzten Beitrag dieser Serie geht es in dieser Ausgabe darum, besonders empfindliche pneumatische Sensoren zu bauen – nur mit aktuellen fischertechnik-Teilen. Wir werden in zwei Modellen zum Nachbauen eine neue Bauart von Drossel, einen neuen, empfindlichen pneumatischen Taster, eine enorm empfindliche pneumatische Staudüse und als Krönung einen Druck-Verstärker kennenlernen.

Modell 1: Ein pneumatisches Digital-Zählwerk

Ein mechanisches Zählwerk (siehe auch *Vom Zählen und Abzählen (1)* in ft:pedia 1/2012 [1]) kann natürlich auch durch einen Pneumatikzylinder angesteuert werden. Einen Modellvorschlag dafür zeigen Abb. 1. und 2, mit den folgenden Komponenten:

- Ein Kompressor liefert die Druckluft.
- Die Druckluft muss zunächst durch eine Drossel. Deren neue Bauart und warum sie notwendig ist, werden wir gleich noch ausführlich besprechen.
- Die so gedrosselte Druckluft geht gleichermaßen zu einem einfachwirkenden Pneumatikzylinder (also einem mit Rückstellfeder, von dem nur ein Anschluss verwendet wird) sowie
- zu einer ebenfalls neuen Bauart von pneumatischem Taster.

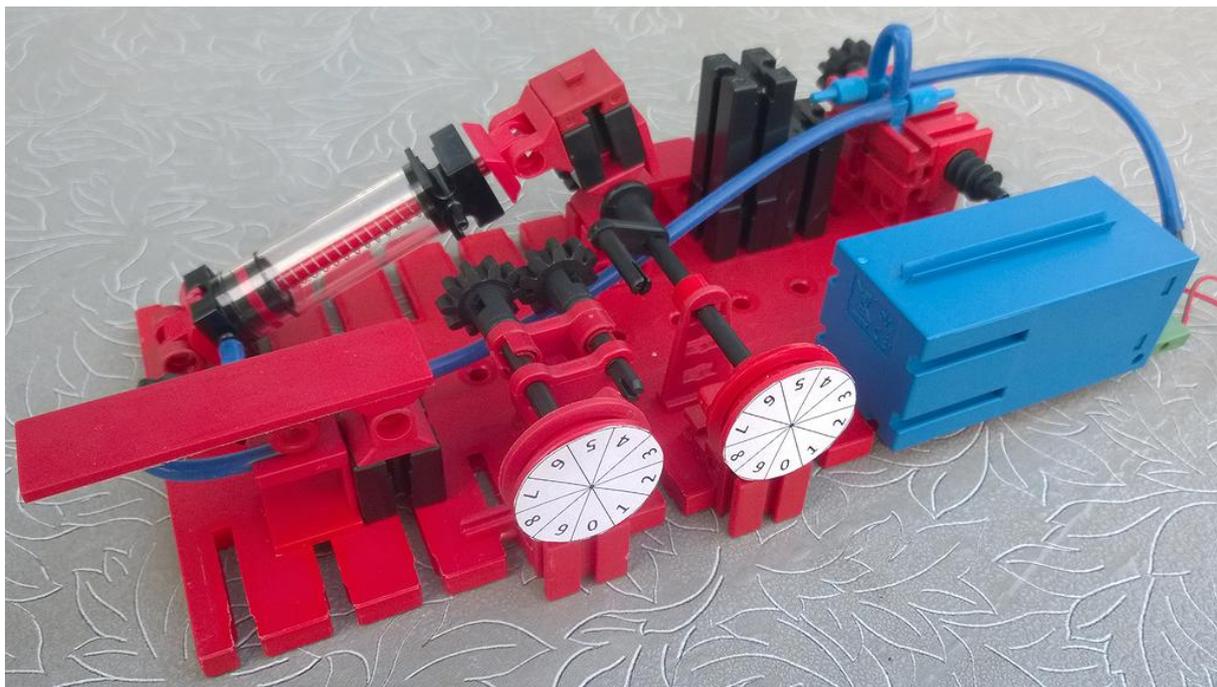


Abb. 1: Pneumatisches Zählwerk mit Taster-Ansteuerung

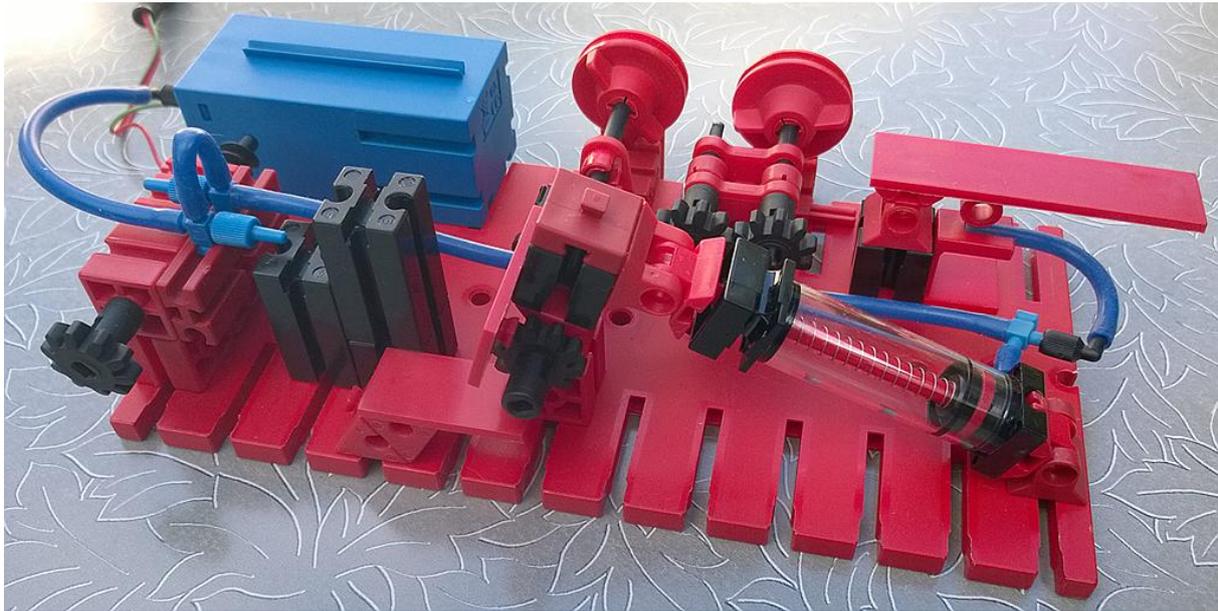


Abb. 2: Rückansicht des pneumatischen Zählwerks

- Der Pneumatikzylinder betätigt ein Schrittschaltwerk,
- welches schließlich ein zweistelliges digitales Zählwerk ansteuert.

Pneumatisches Schrittschaltwerk

Auch ohne Drossel und Taster könnt ihr das Zählwerk selbst schon aufbauen und testen. Jedes Mal, wenn der Zylinder mit Druckluft beaufschlagt wird, fährt er aus und schaltet ein Rast-Z10 um genau einen Zahn weiter. Eine Sperre verhindert, dass beim Zurückfahren des Zylinders das Z10 mit zurück genommen wird. Abb. 3 zeigt die Mechanik im Detail.

Die Anordnung kommt mit wenigen Bauteilen aus: Die Achse des zählenden Z10 sitzt in einem Gelenkbaustein (siehe auch *Raffiniertes mit Achsen* in ft:pedia 3/2013 [2]). Am beweglichen BS5 sitzt oben eine Verkleidungsplatte 30x15. Die greift mit ihrer Unterkante in die Zähne des Z10 ein, sodass beim Bewegen des BS5 durch den Pneumatikzylinder das Zahnrad gedreht wird. Links in Abb. 3 ist ein stabil gebauter Anschlag in Form einiger Grundbausteine zu sehen. Der sorgt dafür, dass der Zylinder

auch bei vollem Druck das Z10 nicht um mehr als einen Zahn weiter schaltet.

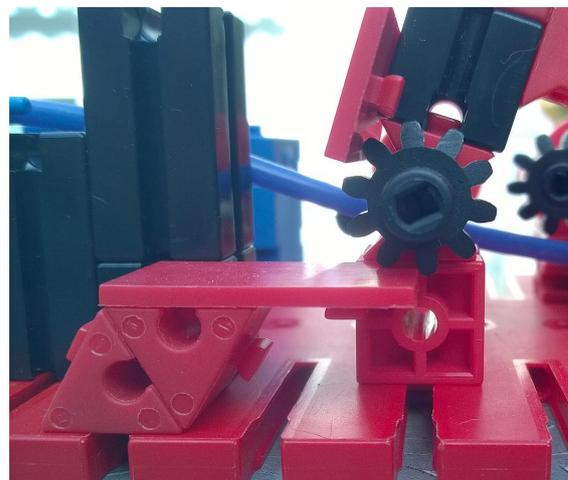


Abb. 3: Zähl- und Sperrklinke

An der Unterseite des Zahnrad befindet sich, einfach von zwei Winkelsteinen 60° getragen, eine weitere Platte 30x15. Die ist so angeordnet, dass sie ein Zurückdrehen des Z10 beim Wiedereinfahren des Zylinders zuverlässig verhindert, denn auch sie greift in die Zähne des Z10 ein.

Am besten probiert ihr die Mechanik zunächst durch Verdrehen des Zählhebels von Hand aus, und dann mit einfacher Ansteuerung des Zylinders.

Das Zählwerk

Die Mechanik zum digitalen Zählen ist ähnlich wie in [1] aufgebaut (Abb. 4). Auf der gegenüberliegenden Seite der Rastachse des zählenden Z10 steckt eine Rastkurbel, und zwar eine solche, in der man wiederum eine Rastachse so einführen kann, dass diese mitgedreht wird. Letztere Achse führt direkt zu der Flachnabe für die Einer-Stelle der Zählung. Da zum Zählen ein Z10 um genau einen Zahn pro Schritt weitergedreht wird, können wir die Flachnaben einfach mit den Ziffern 0–9 beschriften. Eine Vorlage zum Ausschneiden zeigt Abb. 5.



Abb. 4: Zählwerk mit Zehner-Übertrag

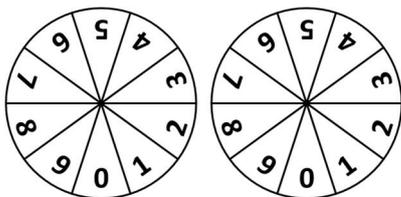


Abb. 5: Ziffernaufkleber für die Flachnaben

Der Übertrag von der Einer- zur Zehnerstelle geschieht dadurch, dass die Rastkurbel ein Mal pro Umdrehung ein weiteres Z10 wiederum um genau einen Zahn weiterdreht. Damit die Flachnaben der Einer- und Zehnerstelle sich in dieselbe Richtung drehen und also identisch beschriftet werden können, ist noch ein Zwischen-Z10 zur Drehrichtungsumkehr eingefügt.

Damit das sauber funktioniert, muss das von der Kurbel angesteuerte Z10 hinreichend genau im richtigen Abstand zur

Kurbel stehen. Deshalb sitzen die Z10 der Zehnerstelle auf einer per Federnocken querverschiebbaren Anordnung aus BS7,5, BS5 mit zwei Zapfen und [Rollenbock 32085](#) (Abb. 6).



Abb. 6: Justierung der Zehnerstelle

Die Ansteuerung des Zählwerks

Besitzer der älteren fischertechnik/Festo-Pneumatikteile würden zum Ansteuern des Pneumatikzylinders mit einem Taster nicht lange grübeln müssen. Ein [Ventil mit blauem Stößel \(ein Öffner\) 36082](#) und vielleicht noch ein [Rollenhebel 31641](#) [3], und fertig ist die Laube!

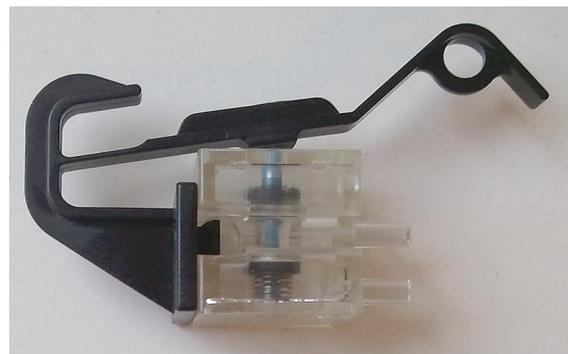


Abb. 7: Kombination aus Festo-Ventil und Betätigungshebel der ft-Ur-Pneumatik

Es könnte also alles ganz einfach sein – wenn diese Teile noch produziert würden und nicht bloß selten und zu horrenden Preisen auf dem Gebrauchtmart zu bekommen wären. (*fischertechnik, hört ihr?*)

Ein neuer Pneumatik-Taster

Nun haben wir ja bereits pneumatische Taster mit Schlauch-Logik in der ft:pedia mit aktuellen Teilen selbst konstruiert [4]. In dieser Ausgabe stellen wir eine weitere Variante vor: Unser neuer Taster verliert zwar ein wenig Druckluft, aber dafür

- kommt er mit geradezu banal wenig Bauteilen aus,
- hat einen ganz geringen Betätigungsweg,
- benötigt nur wenig Kraft zur Betätigung und
- dürfte in seiner Konstruktion auch unerwartet und überraschend sein:

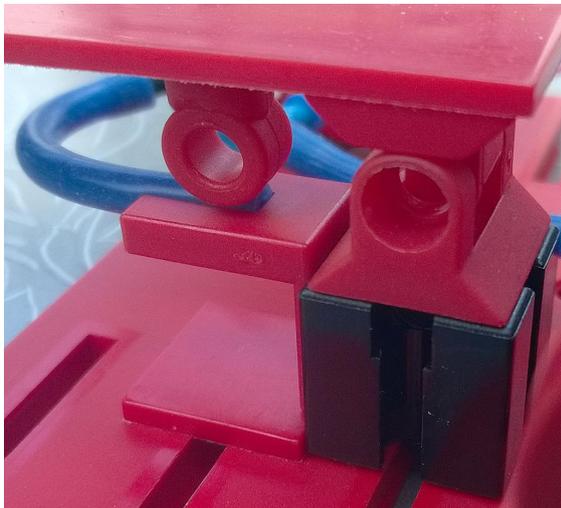


Abb. 8: Der neue Pneumatik-Taster

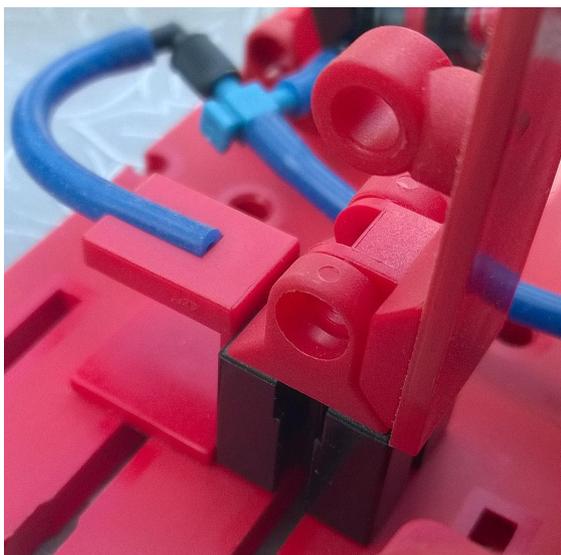


Abb. 9: Der winzige Abluftausgang des Tasters



Abb. 10: Bauplatte 15x15 als Abstandshalter

Um die Wirkungsweise zu verstehen, schauen wir uns aber zunächst das komplette Schaltbild des Zählwerks an:

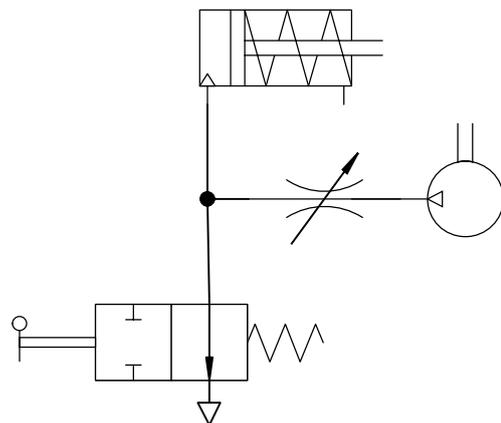


Abb. 11: Schaltbild des Zählwerks

Das funktioniert so:

- In Ruhestellung des Tasters ist die anliegende Druckluft direkt mit der Abluft verbunden und strömt einfach heraus. Das ist in Abb. 9 durch das kleine Bisschen vom Ausgang des Schlauchs realisiert, welches im Freien endet. Dadurch, dass das in der Nut „übrig bleibende“ Loch klein ist, geht die Luft sogar nur etwas gedrosselt ab. In Ruhestellung strömt also die Druckluft so stark ins Freie, dass der Pneumatikzylinder nicht genügend Druck bekommt, um gegen seine Rückstellfeder auszufahren.
- Drücken wir aber den Taster, wird das in der Nut befindliche Schlauchende aber

durch einen [Pneumatik-Achsadapter 31422](#) dergestalt zugequetscht, dass sein kleiner Abluftausgang praktisch vollständig verschlossen wird. Dadurch kann die durch die Drossel strömende Druckluft, wenn auch nicht schlagartig, den Pneumatikzylinder füllen und zum Ausfahren bringen.

- Lassen wir den Taster los, wird deutlich, warum die Drossel notwendig ist: Es darf nur weniger Luft vom Kompressor nachströmen, als durch das Rest-Loch des Schlauchs im Taster abströmen kann. Ohne die Drossel wäre das nicht der Fall, und der Zylinder bliebe ständig ausgefahren. Bitte ausprobieren!

Abb. 10 zeigt schließlich, dass für die richtigen Abstände unterhalb des [38423 Winkelstein 10x15x15](#) einfach eine untergeschobene Bauplatte 15x15 genügt. Die sorgt dafür, dass sich der Winkel auch nach häufiger Betätigung nicht in der Nut des BS15 absenken kann.

Eine neue Drossel

Unsere bisherigen Selbstbau-Drosseln [4] funktionierten dadurch, dass wir eine Schraube oder ft-Schnecke mehr oder weniger stark auf einen Schlauch drücken ließen. Ein Nachteil dieser Bauart ist aber, dass der Schlauch nicht nur zugeedrückt, sondern auch durch das auf ihm sich drehende Stellglied bei jedem Drehen etwas beansprucht wird. Das führt nach vielen Einstell-Drehungen mitunter zu Materialermüdung des Schlauches, weil schlicht ein Loch hineingebohrt wurde.

Diesen Nachteil vermeiden wir, indem wir uns auf die ja schon in Ventilen [4] erfolgreich verwendete Methode des Abknickens eines Schlauches besinnen:

Die in Abb. 12 vorgestellte neue Drossel-Variante knickt ein kurzes Stückchen Schlauch per Schneckenantrieb mehr oder weniger stark ab und bewirkt so die einstellbare Drosselwirkung.

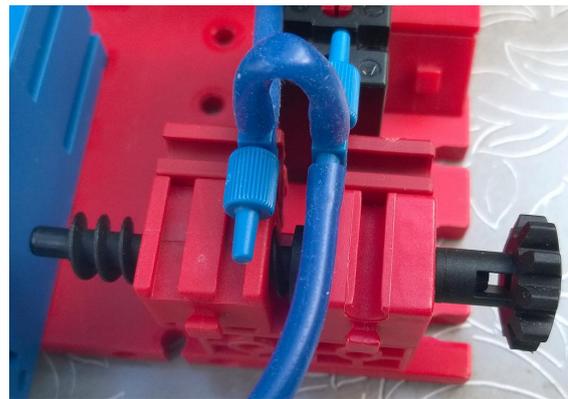


Abb. 12: Die neue Drossel

Nur der rechte BS15 mit Bohrung im Bild ist mit dem darunterliegenden BS15 verbunden. Der untere linke soll nur verhindern, dass sich der Schneckenbaustein auf der Schnecke verdreht. Durch das Einstell-Z10 kann also der Abstand des Schneckenbausteins (links) vom Führungsbaustein (rechts) eingestellt werden.

An zwei BS7,5 ist wieder je ein Pneumatik-T-Stück befestigt, von denen je ein Ausgang per P-Stopfen verschlossen ist. Die ankommende Luft muss also durch das kurze, variabel stark abgknickte Schlauchstückchen gedrosselt durch, um zum Ausgang auf der anderen Seite zu gelangen.

Das ergibt

- eine sehr fein einstellbare Drosselwirkung,
- eine sehr leichtgängige Einstellung sowie
- die Schonung des Schlauchmaterials, weil nichts an ihm reibt.

Die Drossel muss nun so stark geschlossen werden, dass im Pneumatikzylinder befindliche Druckluft bei unbetätigtem Taster ins Freie abströmen kann. Durch die gedrosselte Zufuhr und (über den winzigen Ausgang des Tasters) Abfuhr der Druckluft geht ein Zähler Schritt zwar nicht besonders schnell, aber zuverlässig.

Damit ist das Zählwerk vollendet. Jeder Druck auf den Taster zählt einen Schritt weiter, von 00 bis 99 immer wiederholend.

Modell 2: Pneumatischer Bausteinauswurf

Im nächsten Modell wollen wir eine Anordnung konstruieren, die auf eine ganz sachte, leichte Betätigung hin einen Pneumatikzylinder mit vollem Druck ausfährt. Wir brauchen also einen besonders empfindlichen pneumatischen Sensor und eine Verstärkung des von diesem Sensor stammenden Druckluftsignals.

Die herzustellende Maschine soll Bausteine, die sich in einem Magazin befinden, einzeln auswerfen, bis das Magazin leer ist. Immer wenn sich ein Baustein direkt vor dem auswerfenden Pneumatikzylinder befindet, soll dieser ausfahren, um den Baustein auszustößen – und dann natürlich wieder einfahren, um Platz für den nächsten Baustein zu schaffen. Die Maschine ist in Abb. 13 von vorne zu sehen.

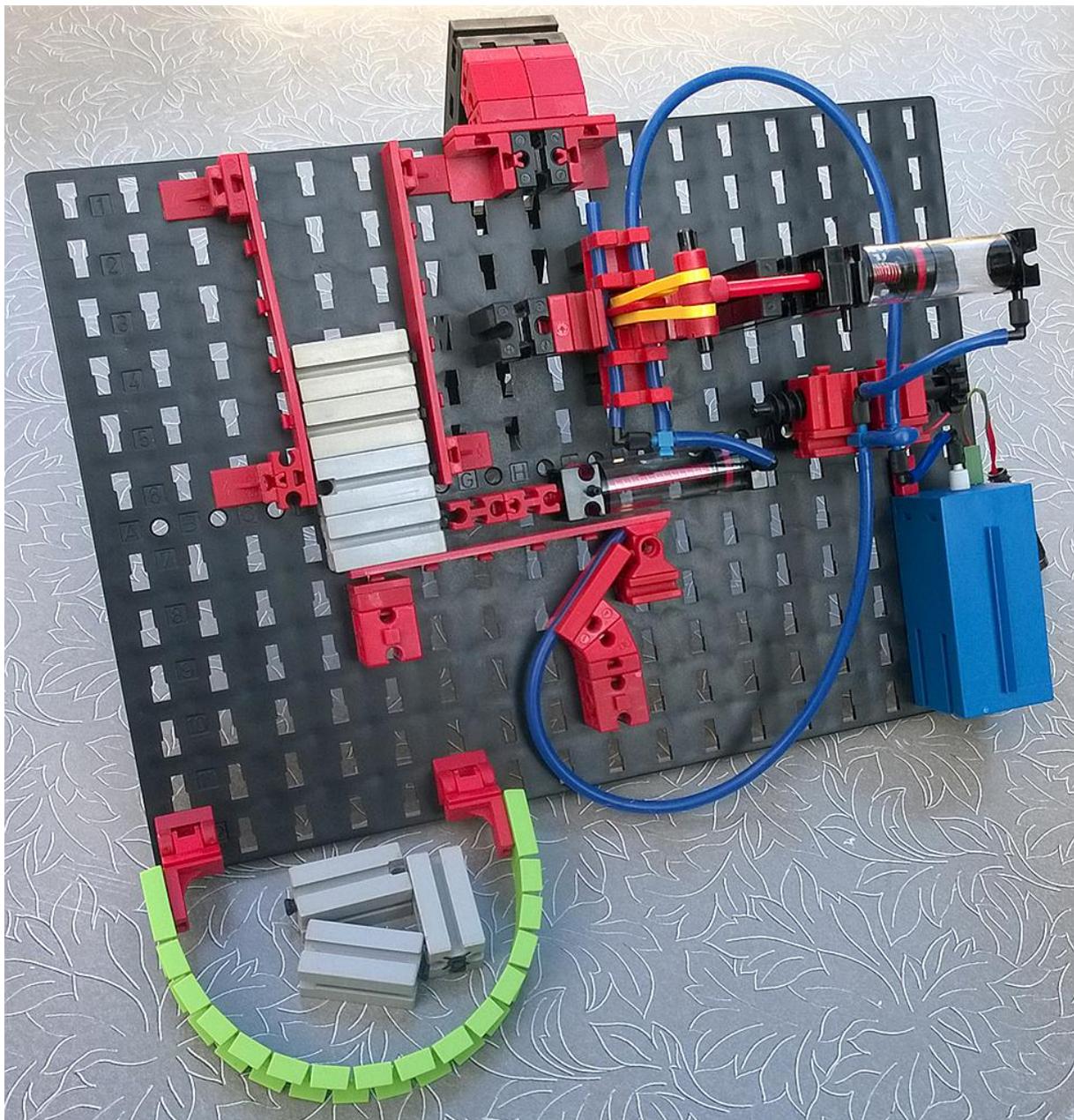


Abb. 13: Pneumatischer Auswurf mit empfindlichem Staudüsen-Sensor und Druckverstärker

Die Maschine steht schräg, sodass die (im Bild grauen) Bausteine 30 im Teilemagazin herunterrutschen. Um Bausteine leicht zuführen zu können, ist das Magazin oben eine Spur breiter ausgeführt also unten (man beachte, dass links oben ein BS5, links unten aber ein BS7,5 verwendet wurde).

Die Sensor-Konstruktion unten ist extrem leichtgängig, und ihr (schwaches) Druckluftsignal wird über den Druckverstärker rechts oben dem Pneumatikzylinder zugeführt, der schließlich den untersten Baustein im Magazin nach links auswirft. Unten werden die Bausteine gesammelt. All dies besprechen wir nun der Reihe nach:

Staudüsen

Wenn man mit Pneumatik etwas abtasten möchte, was keinen starken Druck (etwa auf ein Ventil) ausüben kann, ist eine der Möglichkeiten der Einsatz einer *Staudüse*. fischertechnik hatte ein solches Bauteil unter der Bezeichnung [Pneumatik-Düsenbaustein 31631](#) im Ur-Pneumatik-Programm:



Abb. 14: Staudüse

Für eine Beschreibung und ein Anwendungsbeispiel siehe [3]. Kurz gesagt, strömt in einen der beiden Anschlüsse Druckluft, die durch ein kleines Loch (in Abb. 14 nach oben) abströmt. Da das recht ungehindert passiert, muss die Zuluft gedrosselt werden. Wird nun das Loch abgedichtet (von Hand oder durch ein zu registrierendes geeignetes Bauteil), entsteht

am zweiten Anschluss Druck. Dieser kann mit den ursprünglichen Ventilen und Betätigern verstärkt und weiterverarbeitet werden.

Aber auch mit aktuellen Teilen findet sich eine Lösung, das zu realisieren. Ganz ähnlich zu unserem neuen Taster (Abb. 8/9) verwenden wir wieder nur einen einfachen Schlauch, dem wir – gedrosselt mit unserer neuen Drossel – Druckluft zuführen. Wenn wir den Schlauch an seinem freiliegenden Ende zuhalten, baut sich im inneren des Schlauchs also je nach Einstellung der Drossel mehr oder weniger schnell Druck auf.

Ein BS30 auf einer Schräge bringt nun nicht viel Gewichts- und damit Betätigungskraft auf. Damit der Schlauch auch mit dieser geringen Kraft sauber abgedichtet wird, verwenden wir eine frappierend einfache Konstruktion, die uns aber dennoch eine riesige Hebelwirkung beschert (Abb. 15).

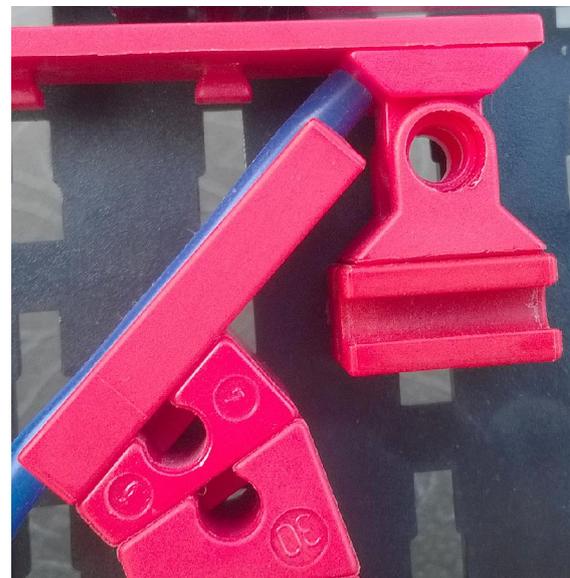


Abb. 15: Selbstbau-Staudüse

Wir lassen tatsächlich einen – sorgfältig gerade abgeschnittenen – Schlauch im Winkel von 45° genau so enden, dass die kleine flache Seite eines Gelenkbausteins genau auf ihm endet. Zusammen mit der Bauplatte 15x60 ergibt das eine gigantische Hebelwirkung: Ohne vom Magazin

kommende BS30 genügt die Elastizität des Schlauches selbst, den Gelenkstein gerade so weit hoch zu drücken, dass sein plan abgeschnittenes Ende nicht plan vom Gelenkstein abgedeckt wird. Dadurch kann Druckluft entweichen.

Sobald aber nur ein Hauch von Belastung auf das entfernte Ende der Bauplatte 15x60 kommt, wird das Schlauchende zuverlässig verschlossen. Das winzige Stückchen Schlauch, was aus dem [Baustein 5 15x30 \(35049\)](#), der ihn führt, auf dem Weg zum Gelenkstein herausragt, genügt als federndes Element völlig. Etwas Justierung ist wohl nötig, aber weniger als befürchtet - die Anordnung funktioniert überraschend perfekt.

Die Schaltung der Anlage

Um die weitere Besprechung zu vereinfachen, führen wir uns zunächst das gesamte Schaltbild der Maschine zu Gemüte. Die Schaltung funktioniert ganz ähnlich wie die unserer Zählmaschine (siehe Abb. 11), nur dass hier vor dem Pneumatikzylinder ein noch zu besprechender Druckverstärker (im Schaltbild ein Dreieck) eingefügt ist.

Die Versorgungsluft kommt nur gedrosselt an. Deshalb kann evtl. im Zylinder befindliche Druckluft durch unsere in unbetätigtem Zustand ja etwas geöffnete Staudüse entweichen und der Zylinder durch seine Rückstellfeder einfahren.

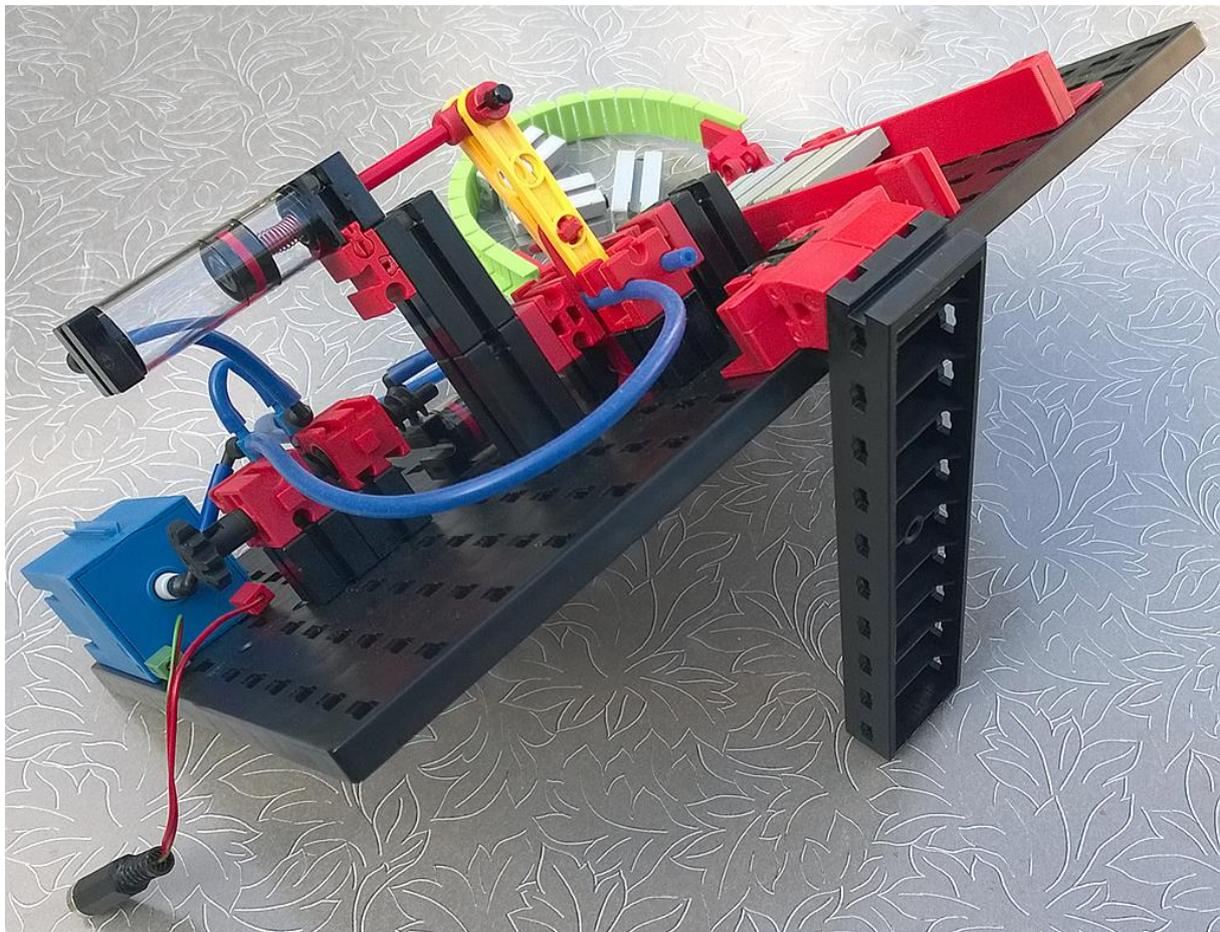


Abb. 16: Blick auf die Rückseite des Modells und des Druckverstärkers

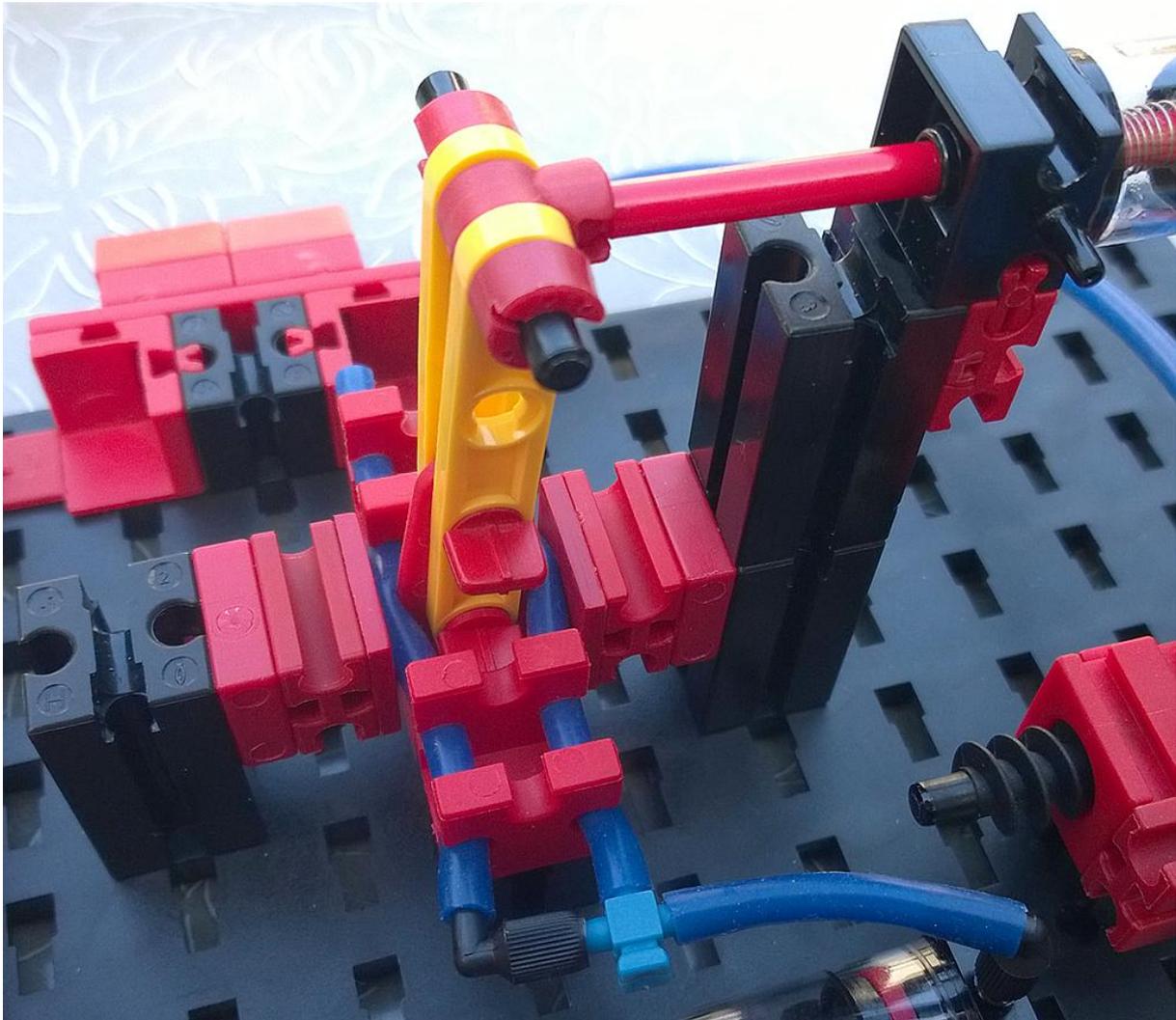


Abb. 18: Pneumatischer Druckverstärker

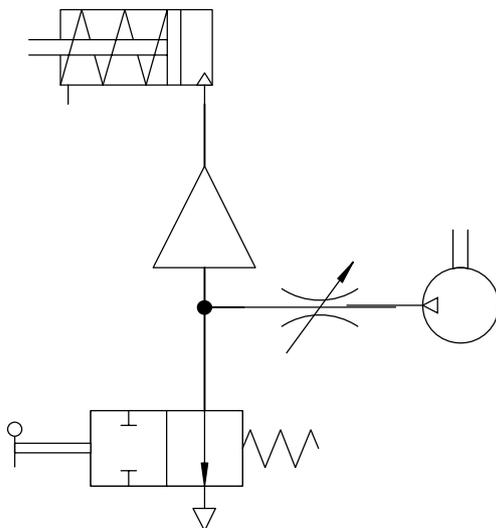


Abb. 17: Schaltung des Bausteinauswurfs

Wird nun die Staudüse (im Schaltbild der Taster unten) verschlossen, baut sich etwas Druck auf. Allerdings nicht sehr viel, denn ein zu großer Druck würde nur dazu führen, dass die Druckluft doch aus dem weichen Schlauchende unserer Staudüse entweicht. Der Druck muss also verstärkt werden, um für die Betätigung des Pneumatikzylinders auszureichen.

Ein aktiver Proportional-Druckverstärker

Wie bekommen wir eine Druckverstärkung mit aktuellen fischertechnik-Teilen hin? Überlegen wir:

Wir müssen dem Zylinder sowohl Druckluft zuführen als auch diese wieder ins Freie abströmen lassen können. Der von der Staudüse bewirkte schwache Eingangsdruck muss dabei in einen hohen Druck übersetzt werden – stark genug für den Zylinder. Anders gesagt: Eine nur kleine Druckänderung am Eingang muss eine große Druckänderung am Ausgang zur Folge haben.

Wenn wir mal in die vielleicht gewohntere Mechanik schweifen: Eine kleine Kraft übersetzt man dort in eine große mittels eines geeigneten *Hebels*. Was liegt also näher, als einen *langen, leichtgängigen* Hebel mit einem (weiteren) Pneumatikzylinder um ein relativ *langes Maß* zu bewegen, um damit einen *kurzen* Hebel mit *viel Kraft* um eine *kurze Strecke* zu bewegen. Dazu würde es ja genügen, einfach einen *langen* Hebel von einem Zylinder bewegen zu lassen, und an einer dem Hebeldrehpunkt näherliegenden Stelle die benötigte *kräftige* Bewegung (wenn auch nur über eine *kurze* Distanz) zu verwerten.

Bleibt also noch die Aufgabe, mit einer *kurzen, aber kräftigen* Bewegung ein selbstgebautes Pneumatikventil zwischen „Druckluft durchlassen, Abluft verschließen“ und „Druckluft absperren, Abluft ins Freie strömen lassen“ umzusteuern. Das wiederum gelingt mit der in Abb. 18 gezeigten Konstruktion.

Der Trick ist, je einen Pneumatikschlauch für Zu- und Abluft *immer fast vollständig zugeklemmt* zu halten. Damit ist nur noch ein geringes Stückchen Hebelweg notwendig, um den einen oder eben den anderen Schlauch *vollständig* zu verschließen und sein jeweiliges Gegenstück zumindest soweit zu öffnen, dass es für die Ansteuerung des Auswurfzylinders genügt.

Da hier ganz ordentliche Kräfte zusammenkommen, muss diese Anordnung hinreichend stabil gebaut sein. Abb. 20 zeigt

eine Detailaufnahme. Hier ist eine der äußeren Begrenzungen abgenommen. Dabei ist bitte zu beachten: Damit auch die S-Riegel 8 genug Platz haben und die „Absperrflosse“ beweglich bleibt, sitzt letztere nicht etwa mittig auf dem Gelenkstein, sondern versetzt. Der Gelenkstein muss also mit seiner Nut in der richtigen Richtung verbaut werden. Aus demselben Grund sitzen die Rollenböcke, die die Schläuche aufnehmen, nicht ganz nach innen geschoben, sondern mit etwas Abstand.

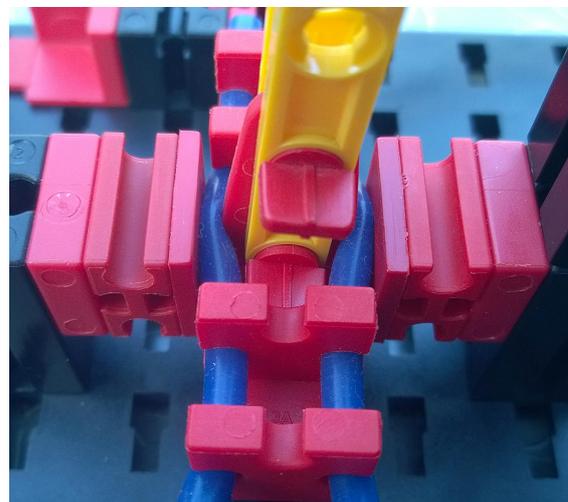


Abb. 19: Beide Schläuche sind immer fast zugeklemmt

Der steuernde Pneumatikzylinder sitzt weit oben an einem langen Hebel. Für ihn genügt also wenig Druck (der der Staudüse nämlich), um über den langen Hebel aus zwei gelochten Statikstreben 45 eine gelenkig aufgehängte [38260 S-Kupplung 22,5](#) zwar nur ein wenig, aber dafür sehr kräftig, zu verdrehen. Deren schmale Kanten sind prima geeignet, scharf in die Schläuche hinein zu drücken und sie zuverlässig abzuklemmen. Dadurch, dass dieses Bauteil so dünn ist, braucht es die Schläuche ja nur an einer kleinen Stelle einzudrücken; deshalb genügt hier die erreichbare Kraft.

Links und rechts sind beide Schläuche sehr stramm zwischen der beweglichen Absperrflosse und den drum herum angebrachten

Bauteilen eingeklemmt. Je ein BS5, BS7,5 und eine Bauplatte 15x15 bringen die richtigen Abstände.

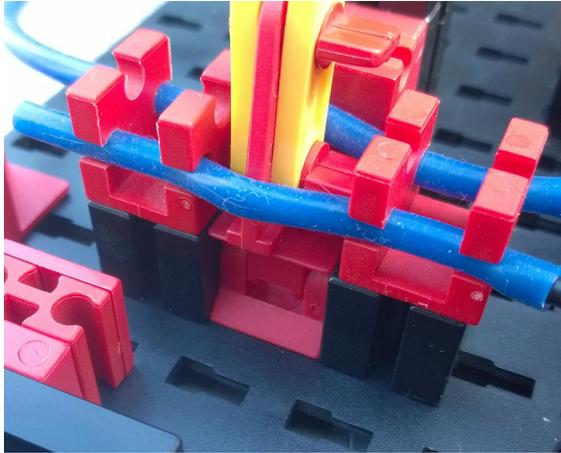


Abb. 20: Detailblick auf den Verstärker

Durch den in Abb. 19 rechten Schlauch liegt ungedrosselte Druckluft direkt vom Kompressor an – also mit vollem Druck! Sofern diese durchgelassen wird, landet sie am Anschluss des Auswurfzylinders. Andernfalls kann die Druckluft im Zylinder durch den linken Schlauch als Abluft entweichen – der endet nämlich einfach im Freien.

Ein winziger Verschiebeweg (ein oder zwei Millimeter genügen!) am oberen Hebelende reichen nun aus, um mit großer Verstärkungswirkung Druckluft in den Auswurfzylinder zu pumpen bzw. sie wieder abzulassen.

Damit ergibt sich die tatsächliche Gesamtschaltung mitsamt den Details des Druckverstärkers wie in Abb. 21 dargestellt.

Auch bei dieser Maschine muss die Drossel wieder hinreichend stark geschlossen sein, sodass die Abluft auch entweichen kann, ohne dass durch zu viel Druckluftzufuhr permanent Druck am Verstärkereingang anliegt.

Wenn alles richtig justiert ist, funktioniert die Anlage aber geradezu ungeheuerlich gut. Versucht einmal, mit einem Finger nur ganz sacht am Ende der Bauplatte 15x60 zu drücken und beobachtet, wie sauber

verstärkt und sogar proportional zur Betätigung der Auswurfzylinder ausfährt. Eine Spur weniger stark drücken, und der Zylinder fährt wieder ein. Da braucht es nicht mal das Gewicht eines BS30 – Experimente zeigten, dass sogar ein roter BS15 mit Bohrung, der ja wirklich leicht ist, mit seinem Gewicht genügt und zuverlässig ausgeworfen wird.

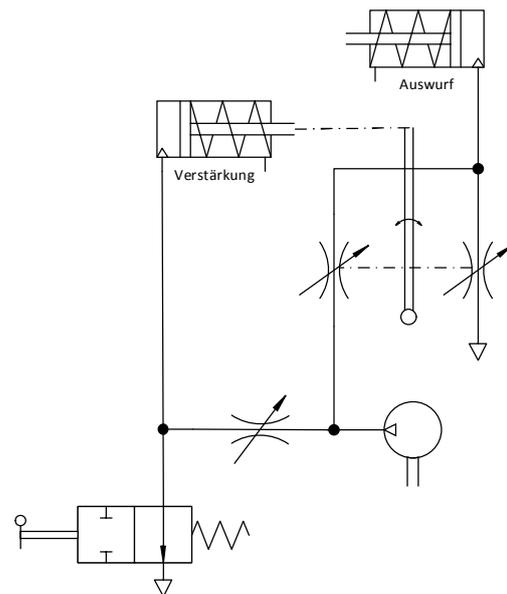


Abb. 21: Vollständige Schaltung des Bausteinauswerfers

Wer weitere Teile und Ideen hat, mag die Maschine natürlich beliebig ausbauen. Vielleicht wollt ihr die ausgeworfenen Bauteile mit einem Zählwerk wie im ersten Modell zählen lassen? Nur zu!

Quellen

- [1] Falk, Stefan: *Vom Zählen und Abzählen*. [ft:pedia 1/2012](#), Seite 22.
- [2] Falk, Stefan: *Raffiniertes mit Achsen*. [ft:pedia 3/2013](#), Seite 38.
- [3] Falk, Stefan: *Perlentauchen (Teil 5)*. [ft:pedia 4/2013](#), Seite 6.
- [4] Falk, Stefan: *Druckluftsteuerungen (Teil 1)*. [ft:pedia 1/2014](#), Seite 69.

Computing

Druckmessung mit RoboTX Controller

Andreas Gail

Seit der Einführung von Pneumatik-Bauelementen durch Fischertechnik ist die Druckmessung des Betriebsdrucks in den Pufferbehältern eine bislang nur bedingt gelöste Aufgabenstellung. Mit der Kenntnis des Betriebsdrucks kann beispielsweise das Ein- und Ausschalten des Kompressors geregelt werden (Zweipunktregler) oder die Bewegungskraft der Pneumatikzylinder bei konstantem Druck reproduzierbar eingestellt werden. Für alle diese Funktionen ist eine einfache und verlässliche Druckmessung die gemeinsame Grundlage.

Druckmesszelle und Messumformer

Die Grundlage der Druckmessung ist eine Druckmesszelle, in der der anliegende Druck in ein elektrisches Signal umgewandelt wird, welches anschließend weiterverarbeitet werden kann. Der Gesamtaufbau wird als Messumformer bezeichnet. Hierzu wird ein fertiges Bauteil (Motorola, Senseon, MPX5100) verwendet (z. B. Conrad Electronic [183890](#)).



Abb. 1: Druckmessumformer
Motorola MPX5100

Für weitere Detailinformationen zum Bauteil empfiehlt es sich, das zugehörige Datenblatt von der Internetseite des Herstellers oder Händlers zu beschaffen [2].

Systemintegration

Um den o. g. Druckmessumformer einschließlich Spannungsversorgung für fischertechnik verwendbar zu machen, wird gemäß Datenblätter der Hersteller folgende Schaltung benötigt [1, 2]:

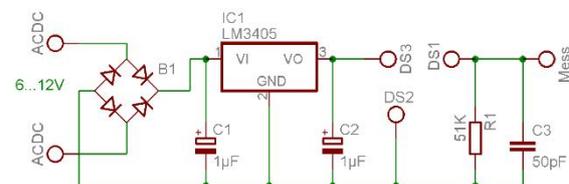


Abb. 2: Schaltung zum Betrieb des
Druckmessumformers Motorola MPX5100

Der Brückengleichrichter B1 hat die Aufgabe, die versehentliche falsche Polung der Versorgungsspannung an die beiden Anschlüsse ACDC zu verhindern. IC1 erzeugt eine geregelte Gleichspannung von 5V, die für eine genaue Messung erforderlich ist. Die Anschlüsse DS1-DS3 werden entsprechend 1-3 an den Druckmessumformer angeschlossen. Die Anschlüsse 4-6 des Druckmessumformers werden nicht beschaltet. Das eigentliche elektrische Messsignal wird am Ausgang Mess bereitgestellt. Die Bauteile R1 und C1 sind zur Stabilisierung des Messsignals vorgesehen. Mit diesen Informationen kann nun der Aufbau begonnen werden.

Schaltungsaufbau

Für den Schaltungsaufbau wird eine fertige Lochrasterplatte empfohlen, die nur auf die passende Größe zurechtgesägt werden muss. Die etwas eigenartige Bestückung dieser Platine wurde so vorgenommen, um den Gesamtaufbau in den roten fischer-technik-Batteriekästen zu ermöglichen.

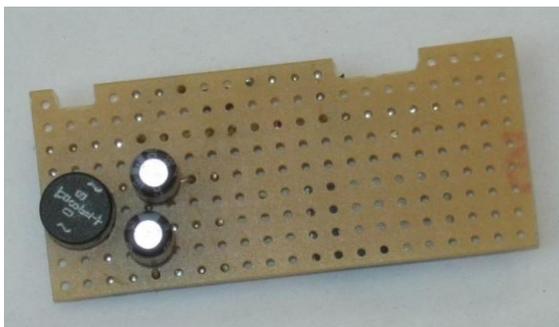
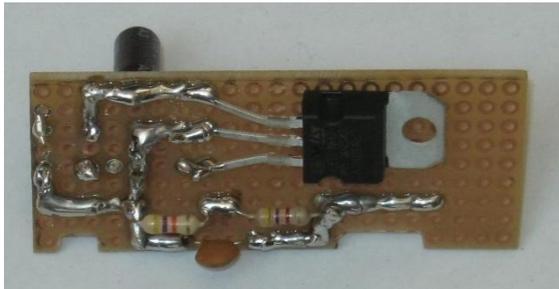


Abb. 3: Schaltungsaufbau zum Betrieb des Druckmessumformers Motorola MPX5100
Ansicht a) von unten, b) von oben, c) seitlich

Die vorgefertigte Platine wird in folgender Art und Weise in das Gehäuse eingebaut, in welchem sich bereits der Druckmessumformer befindet (Abb. 4):

Für den Einbau in das Gehäuse werden die beiden Druckluftanschlüsse des Druckmessumformers mit einer Schlüsselfeile so verkleinert, dass je eine M5-Mutter aufgeschraubt werden kann.

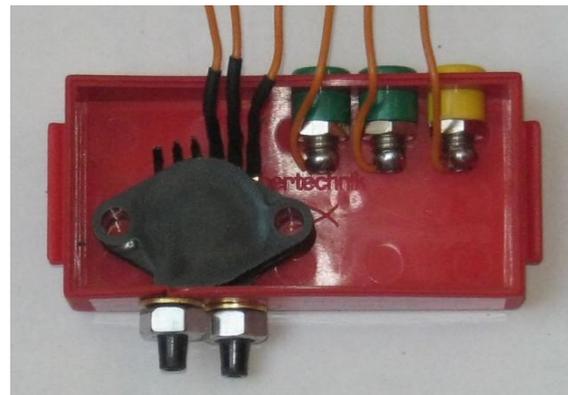


Abb. 4: Gehäuse mit Druckmessumformer Motorola MPX5100 und Steckanschlüssen

Den fertigen Aufbau zeigt Abb. 5.

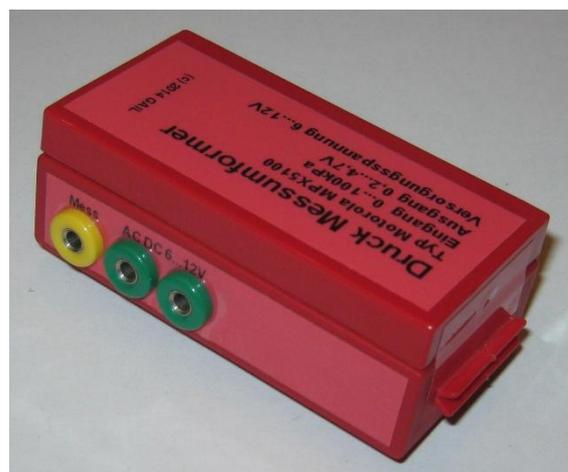


Abb. 5: Gesamtaufbau des Druckmessumformers im Gehäuse einschließlich eingebauter Zusatzschaltung aus Abb. 3

Software

Zur Verarbeitung des vom Druckmessumformer bereitgestellten Spannungssignals soll der Robo TX Controller (TX) verwendet werden, als Programmierumgebung die Robo Pro-Software. Zum Anschluss kann ein beliebiger Eingang am TX genutzt werden.

Dieser Eingang muss zur analogen Spannungsmessung konfiguriert sein. Der eingestellte Sensortyp ist dabei unerheblich; diese Einstellung beeinflusst nur die bildliche Darstellung (Abb. 6).

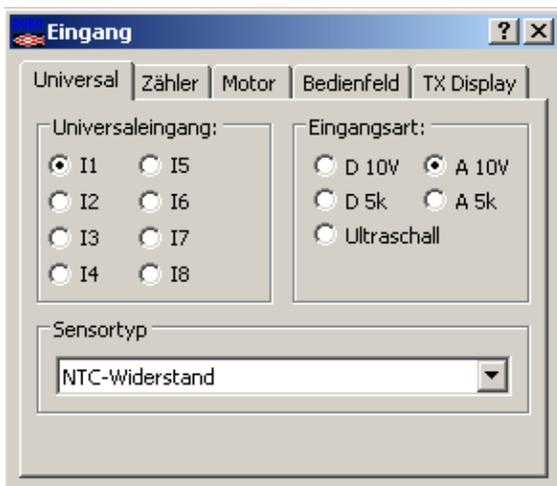


Abb. 6: Konfiguration des elektrischen Eingangs am Robo TX Controller mithilfe der Robo Pro-Programmierungsumgebung

Die Software zur Messung ist recht einfach aufgebaut und zeigt nur den jeweiligen Istwert der Druckmessung an. Zunächst wird periodisch das Messsignal des Druckmessumformers erfasst (siehe Abb. 7). Dieses Messsignal ist in mV angegeben (1000 mV = 1 V) und wird in der Variablen *Spannung* gespeichert. Nachfolgend erfolgt die Umrechnung ausgehend vom Messsignal in mV in eine Druckeinheit in kPa (100 kPa = 1000 mbar = 1 bar). Das Rechenergebnis wird unter der Variablen *Druck_kPa* ausgegeben. Schließlich wird der Druckwert in mbar umgerechnet und unter der Variablen *Druck_mbar* ausgegeben (rot markiert).

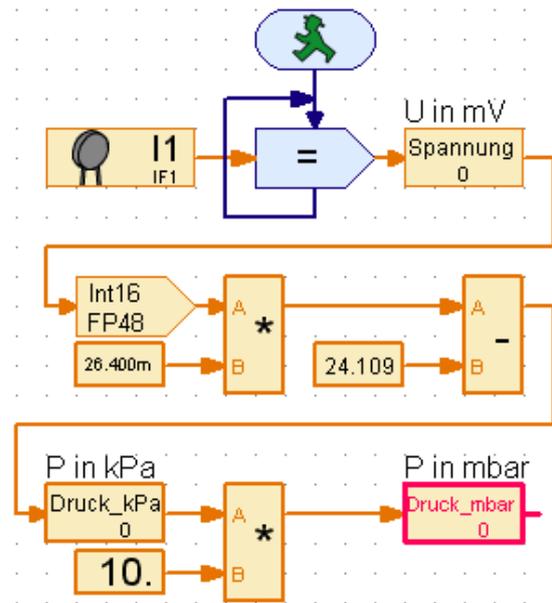


Abb. 7: Robo Pro-Konfiguration zur Umrechnung eines Spannungseingangssignals in eine Druckmessgröße in kPa / mbar (rot markiert)

Kalibrierung / Justierung

Laut Datenblatt des Herstellers liefert der Druckmessumformer folgende Spannungen in Abhängigkeit vom angelegten Druck:

0...100 kPa entsprechen 200...4700 mV

Bei der Betrachtung des Druckwertes wird im Grunde immer der Differenzdruck der beiden Druckeingänge registriert. Da der untere Druckeingang stets offen bleibt beschreibt die Angabe des Druckwertes immer den Überdruck in Bezug auf den Umgebungsdruck. Beim vorliegenden Druckmessumformer zeigt sich bei 0 kPa eine Spannung von 913 mV statt der spezifizierten 200 mV. Die Kalibrierung (Überprüfung) dieses Druckwertes ergab demnach einen erheblichen Messfehler. Zur Kalibrierung eines weiteren Messpunktes müssten ein höherer Druck angelegt (möglichst am oberen Messbereichsende) und mit einem geeigneten Messmittel (also einem möglichst genauen Druckmessgerät) der angelegte Druck und der gemessene Spannungswert notiert werden. Da kein weiteres Druckmessgerät zur Verfügung stand, wurde am oberen Grenzwert die Herstellerspezifikation verwendet. Das

führt sicherlich zu einer recht ungenauen Messung, für die weiteren Erläuterungen sollte es jedoch ausreichend sein.

Die zuvor bei einer Kalibrierung gewonnene Erkenntnis soll nun zur Verbesserung des Messergebnisses eingesetzt werden. Dieses wird als Justierung der Messeinrichtung bezeichnet. Stellen wir uns vor, wir hätten bei einer Zweipunktkalibrierung folgende Messpunkte erhalten:

P [kPa]	U [mV]
0	913
100	4700

Tab. 1: Zwei Kalibrierwertepaare des Druckmessumformers

Jetzt kommt ein Beispiel für alle, die in der Schule nie etwas mit Koordinatensystemen und Geradengleichungen anfangen konnten: Aus den beiden Punkten gemäß Tab. 1 kann mithilfe einer Tabellenkalkulation einfach die zugehörige Geradengleichung berechnet werden. Im klassischen Koordinatensystem wird dabei der Druck P auf der y -Achse aufgetragen und die Spannung U auf der x -Achse. Dieses ergibt folgendes Bild (Abb. 8):

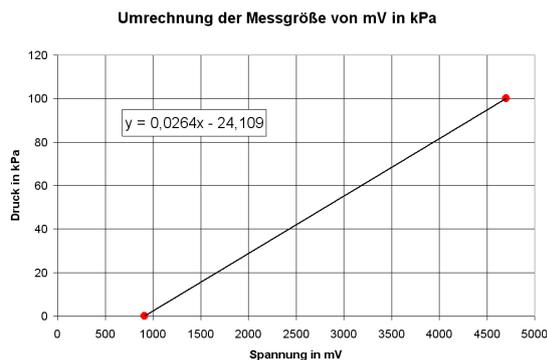


Abb. 8: Berechnung einer Geradengleichung mithilfe einer Tabellenkalkulation

Werden die beiden rot markierten Punkte gemäß Abb. 8 mit einer linearen Trendlinie verbunden ergibt sich folgende Geradengleichung: $y = 0,0264 \cdot x - 24,109$ (allgemeine Geradengleichung: $y = m \cdot x + b$). Das bedeutet eine Steigung der Geraden

$m = 0,0264$ und einen y -Achsenabschnitt $b = -24,109$. Diese beiden Werte wurden zuvor in die RoboPro-Konfiguration gemäß Abb. 7 eingetragen (RoboPro schreibt statt 0,0264 die Zahl 26.400m, was technisch recht ungebräuchlich ist). Wer mehr über nicht-lineare Kalibrierungen wissen will, der werfe einen Blick in [3].

Kalibrierung vs. Eichung

Oft werden oben genannte Tätigkeiten auch als Eichung bezeichnet. Fachlich gibt es dabei keinen Unterschied. Es ist jedoch zu beachten, dass eine offizielle Eichung nur vom Eichamt oder einer zugelassenen Einrichtung durchgeführt werden darf.

Wie geht es weiter?

Mit dem erhaltenen Druckmesswert kann nun kräftig experimentiert werden. Neben den oben bereits genannten Zweipunkt-Druckreglern können auch PI- und PID-Regler aufgebaut werden, oder auch nur einfache Messwertvisualisierungen im zeitlichen Verlauf. Mit einem geregelten Druck wäre auch eine Fahrzeugstabilisierung denkbar, die sich an die Untergrundverhältnisse anpassen könnte.

Viel Spaß bei weiteren Experimenten!

Quellen

Die Links zu den Datenblättern sind nicht angegeben, da sie sich leider häufig ändern.

- [1] STMicroelectronics, *Spannungsregler Datenblatt L78Sxx*, (Doc ID 2148 Rev 7), 2012.
- [2] Motorola, Inc, *Druckmessumformer Datenblatt MPX5100 series* (Doc ID MPX5100/D), 1997.
- [3] Dirk Fox: „Einmessen“ eines digitalen Messgeräts. [ft:pedia 1/2013](#), S. 39-48.
- [4] Georg Stiegler: *I²C mit dem TX – Teil 3: Luftdruckmessung*. [ft:pedia 1/2013](#), S. 32-38.

Mechanisches Rechnen

Binäraddierer

Thomas Püttmann

Binäres Rechnen bildet eine der Grundlagen der heutigen Computertechnologie. In diesem Artikel wird gezeigt, wie aus wenigen fischertechnik-Bauteilen ein fehlerfrei funktionierender, unterrichtstauglicher 4-Bit-Kugel-Binäraddierer konstruiert werden kann. Mit ihm lässt sich die Welt des binären Rechnens spielerisch erforschen.

Geschichte

Zahlen kann man auf verschiedene Art und Weise bezeichnen und benennen. Stellenwertsysteme wie unser heute gebräuchliches Dezimalsystem haben zwei vorteilhafte Eigenschaften: Erstens ist das Hinschreiben und Lesen selbst großer Zahlen mit wenigen Symbolen (Ziffern) möglich. Dadurch ist der Schreib-/Lesevorgang wenig fehleranfällig. Zweitens ergibt sich aus dem System heraus eine systematische Benennung der Zahlen, die zur effizienten mündlichen Kommunikation benutzt werden kann. So kann die Zahl 3421 trotz ihrer Größe (man stelle sich z. B. 3421 Metallachsen vor) im Dezimalsystem mit nur vier Ziffern dargestellt werden, und sie besitzt den systematischen Namen Dreitausendvierhunderteinundzwanzig.

Mechanische Rechenmaschinen benutzten fast ausschließlich die Darstellung der Zahlen im Dezimalsystem. Die ersten solchen Maschinen wurden unabhängig voneinander von [Wilhelm Schickard](#) (1592-1635) und [Blaise Pascal](#) (1623-1662) in der ersten Hälfte des 17. Jahrhunderts erfunden und funktionierten nur unzuverlässig und mit menschlichem Nachhelfen. Einen Durchbruch stellte die vom Universalgelehrten [Gottfried Wilhelm Leibniz](#) (1646-1716) im Jahr 1673 erfundene Staffelwalze dar.

Leibniz war es auch, der als erster die Binärdarstellung von Zahlen entdeckte. Diese Darstellung verwendet nur die zwei Symbole 0 und 1 als Ziffern. Die Zahlen werden damit systematisch 0, 1, 10, 11, 100, 101, usw. benannt. Unter allen Stellenwertsystemen sind die Rechenoperationen im Binärsystem am einfachsten durchzuführen. Der Preis dafür ist, dass man für eine gegebene Zahl im Binärsystem die meisten Stellen benötigt. So hat die Binärdarstellung 110101011101 der Zahl 3421 zwölf Stellen statt vier. Man sagt, es handelt sich um eine 12-Bit-Zahl. Bit steht dabei für *binary digit*.

Weniger bekannt als die Tatsache, dass Leibniz das Binärsystem entdeckte, ist, dass Leibniz auch als erster eine Rechenmaschine skizzierte, die das Binärsystem verwendet. In seiner dreiseitigen Aufzeichnung *De progressionem Dyadica* aus dem Jahr 1679 beschreibt er einen Binäraddierer und -multiplizierer mit rollenden Kugeln, siehe [1, 2]. Nach dieser Skizze entwarf [Ludolf von Mackensen](#) (*1938) ein Modell, das im Jahr 1971 am Deutschen Museum gebaut wurde, aber nicht zuverlässig funktionierte. Ein fehlerfrei rechnendes Modell wurde im Jahr 2004 von [Erwin Stein](#) (*1931) und [Gerhard Weber](#) entworfen und gebaut und ist in der Leibniz-Dauerausstellung der Universität Hannover zu sehen [2].

Leibniz Idee, rollende Kugeln zum binären Rechnen zu benutzen, wurde in veränderter Form in den 60er Jahren des vorigen Jahrhunderts neu entdeckt: Im Jahr 1965 meldete *John Thomas Godfrey* ein Patent unter dem Titel *Binary Digital Computer* an [3]. Neben der allgemeinen Beschreibung von Kugel-Flip-Flops (Abb. 1) enthielt die Patentschrift zwei geniale konkrete Lernspielzeuge, die einige Jahre lang unter den Namen *Digicomp II* und *Dr. Nim* von der Firma E.S.R., Inc. verkauft wurden.

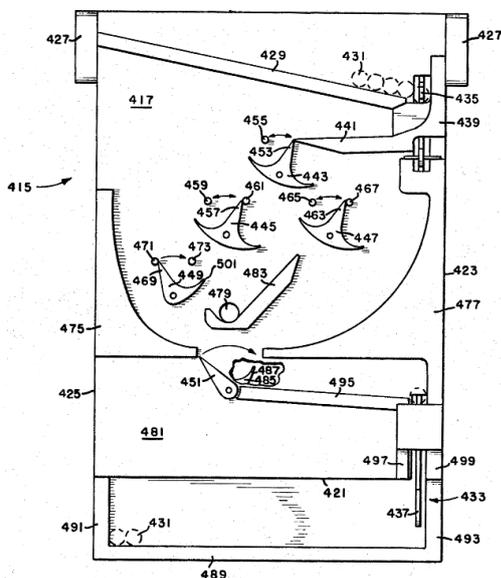


Abb. 1: Kugel-Flip-Flops aus der Patentschrift von J. T. Godfrey [3]

Der Binäraddierer

Für meine eigenen Lehrtätigkeiten entwickelte ich den hier vorgestellten fischertechnik-Binäraddierer (Abb. 2), der auf Godfreys Prinzipien basiert. Das oberste Ziel war dabei die Unterrichtstauglichkeit, d. h. ein einfacher Aufbau aus wenigen Teilen im Raster und eine fehlerfreie Funktion ohne Justierung. Der Nachbau sollte jedem Interessierten anhand der Abbildungen und der Teileliste möglich sein. Die selteneren Winkelsteine 60° mit drei Nuten (31918), die die Spitzen der Wippen bilden, kann man problemlos durch normale Winkelsteine 60° (31010) ersetzen, bei denen man jeweils die zwei Zapfen abtrennt.

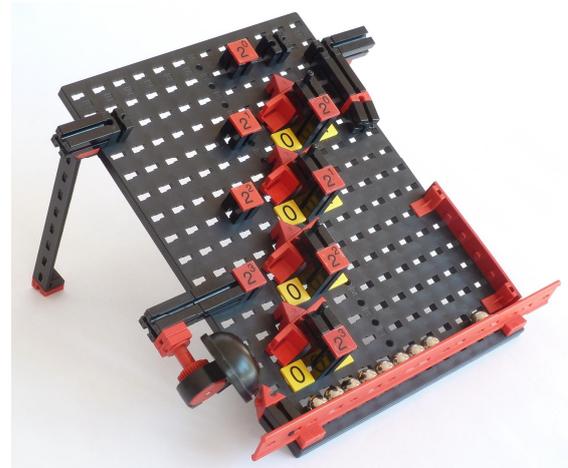


Abb. 2: Der Binäraddierer

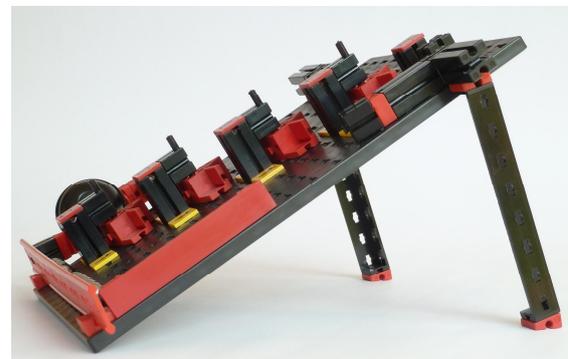


Abb. 3: Ansicht von rechts

Zur Meldung eines Überlaufs verwende ich in der abgebildeten Version eine Fahrradklingel. Fahrradklingeln lassen sich perfekt auf Räder 23 schrauben und können so jede fischertechnik-Kugelbahn akustisch bereichern (Abb. 4).

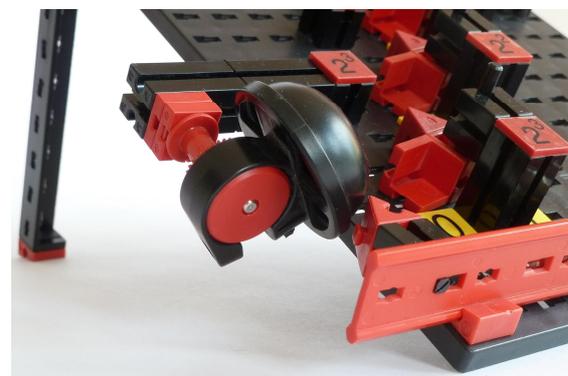


Abb. 4: Klingelbefestigung

Natürlich können Puristen beim Nachbau auch auf eine Klingel verzichten und die linke Kante unten symmetrisch zur rechten Kante mit einer Bauplatte 6 x 1 abschließen.

Vor mir hat übrigens schon Holger Howey einen [fischertechnik-Binäraddierer](#) gebaut. Er war 2010 auf der Modellausstellung in Münster zu sehen.

Teilleiste

Anz.	ft-Nr.	Bezeichnung
1	31010	Winkelstein 60°
1	31016	Seiltrommel rot
1	31034	Metallachse 30
4	31918	Winkelstein 60° (drei Nuten)
7	31981	Winkelstein 15°
4	32064	BS 15 m. Bohrung
9	32879	BS 30 schwarz
13	32881	BS 15 schwarz
1	32985	Grundplatte 500
1	36581	Rad 23 rot
2	37237	BS 5 rot
2	36293	Winkelträger 120
8	38236	Bauplatte 1x1 gelb
8	38240	Eckstein 15
1	38245	Bauplatte 6x1 rot
8	38246	Bauplatte 1x1 rot
4	38414	Kunststoffachse 40
2	38423	Winkelstein 10 rot
10	144262	Kugeln

Eine pdf-Datei mit den Zahlen, die auf die Bauplatten 1x1 geklebt werden, habe ich zum [Download](#) bereitgestellt. Und schließlich benötigt ihr noch eine Fahrradklingel.

Aufbau

Hauptbestandteil des Addierers sind die vier Toggle-Flip-Flops. Jedes dieser Flip-Flops besteht aus einer Wippe, die zwei stabile Zustände einnehmen kann. Ist eine solche Wippe nach links gekippt, so nennen wir den Zustand des Flip-Flops 0. Das Flip-Flop befindet sich im Zustand 1, wenn die Wippe nach rechts gekippt ist. Die vier Flip-Flops zusammen bilden das Addierwerk. Der Gesamtzustand des Addierwerks ergibt sich aus den Zuständen der einzelnen Flip-Flops, indem wir sie von links unten nach rechts oben auslesen und die ausgelesenen Werte von links nach rechts nebeneinander schreiben. Der Gesamtzustand des Addierwerks wird also durch eine 4-Bit-Binärzahl beschrieben.

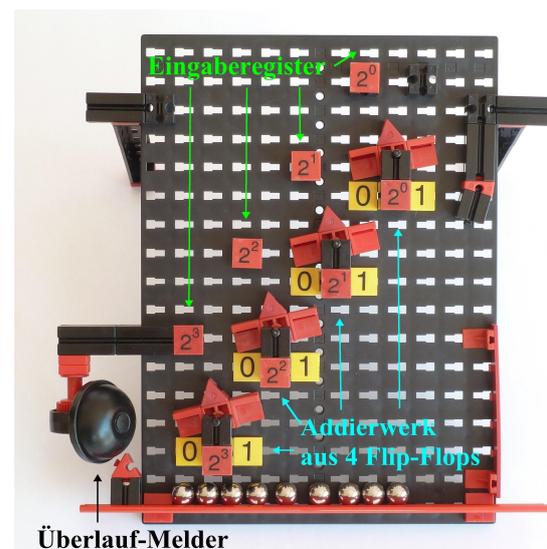


Abb. 5: Aufbau des Binäraddierers

Oberhalb der Flip-Flops befinden sich vier Bausteine 15 mit roten Platten, die zusammen das Eingaberegister bilden. Ein Bit des Eingaberegisters ist 1, wenn auf dem Baustein 15 eine Kugel liegt, ansonsten 0.

Zählen

Zunächst führen wir einen Reset aus, d. h. wir setzen mit der Hand alle vier Flip-Flops des Binäraddierers in den Zustand 0. Der Gesamtzustand des Addierwerks ist also 0000. Lassen wir jetzt Kugeln durch den oberen Einlass fallen, so ändert sich der

Zustand des Binäraddierers. Nach der ersten Kugel befindet er sich im Zustand 0001 (Abb. 6), nach der zweiten im Zustand 0010, danach folgen 0011, 0100, usw. bis zur fünfzehnten Kugel, nach der der Zustand 1111 vorliegt.

Unsere Maschine ordnet somit jeder Zahl von Null bis 15 systematisch eine Bezeichnung zu, die sogenannte Binärdarstellung dieser Zahl. Sie zählt also, denn Zählen ist nichts anderes als das reproduzierbare Benennen einer Anzahl von Objekten.

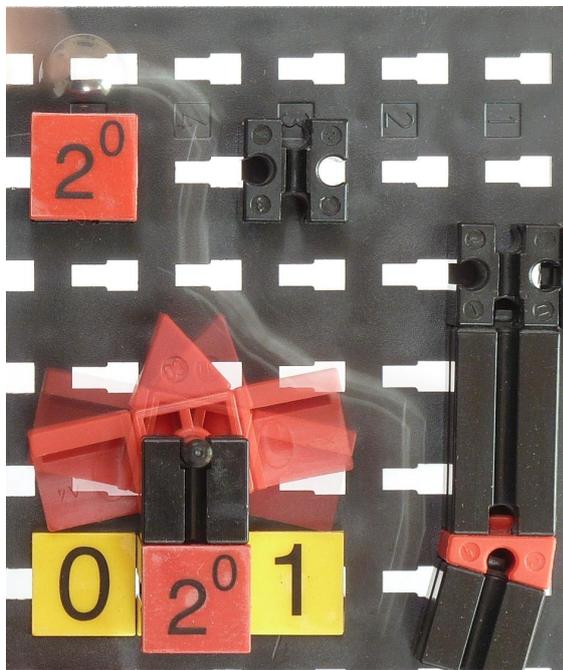


Abb. 6: Umschalten eines Flip-Flops

Addieren

Wenn man noch keine Erfahrungen mit dem Binärsystem gemacht hat, ist es am besten, sich für einen Moment gedanklich vom Rechnen im Dezimalsystem und der Dezimaldarstellung vollkommen zu befreien.

Um das zu verdeutlichen, verwenden wir im Folgenden zwar die bekannten Namen Null, Eins, Zwei, usw. in ihrer ausgeschriebenen Form, symbolisch aber nur die Binärziffern 0 und 1.

Addition ist letztendlich nichts anderes als Weiterzählen. Angenommen, wir wollen Fünf + Sechs mit dem Binäraddierer

berechnen. Dazu könnten wir zunächst fünf Kugeln durch den oberen Einlass fallen lassen und danach sechs. Dabei müssen wir aber die Kugeln selbst abzählen. Es ist bedeutend schneller, die eben beim Zählen gewonnenen Binärdarstellungen Fünf = 0101 und Sechs = 0110 zu verwenden.

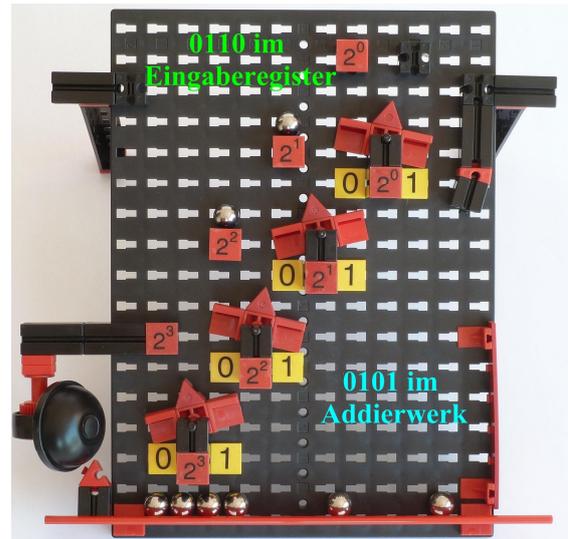


Abb. 7: Vor der Addition

Nach einem Reset wird die 0101 zunächst ins Eingaberegister gegeben und dann durch vorsichtiges Hinunterstoßen von den Sockeln ins Addierwerk transferiert. Anschließend wird 0110 ins Eingaberegister gegeben (Abb. 7) und dann durch Hinunterstoßen zur bereits vorhandenen Zahl 0101 im Addierwerk addiert (Abb. 8).

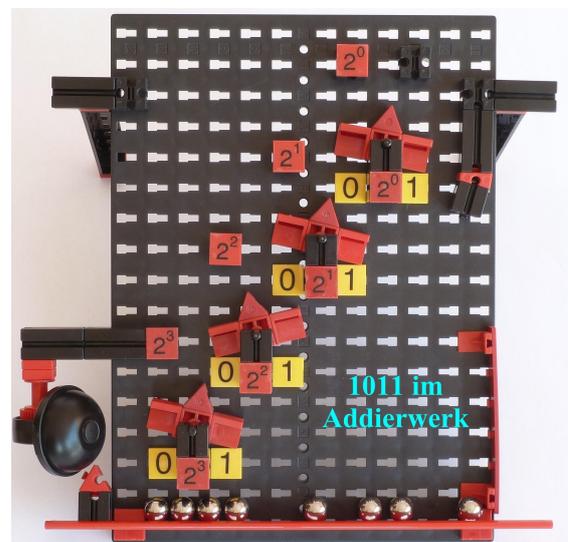


Abb. 8: Nach der Addition

Man erhält als Ergebnis 1011, d. h. die Binärdarstellung der Zahl Elf, die wir beim Zählen oben gewonnen haben.

Funktion

Um genau zu verstehen, wie und warum die Maschine funktioniert, schauen wir uns einmal an, wie die Binärdarstellung von Zahlen mathematisch definiert ist. Eine ganze Zahl N zwischen Null und Fünfzehn wird dabei geschrieben als

$$N = a \cdot \text{Zwei} \cdot \text{Zwei} \cdot \text{Zwei} + b \cdot \text{Zwei} \cdot \text{Zwei} + c \cdot \text{Zwei} + d = ((a \cdot \text{Zwei} + b) \cdot \text{Zwei} + c) \cdot \text{Zwei} + d$$

wobei die vier Stellen a , b , c und d entweder 0 oder 1 sind. Die Zahl Zwei selbst erhält also die Binärdarstellung 0010, die Zahl Vier = Zwei \cdot Zwei die Binärdarstellung 0100, und so weiter.

Die erste Frage, die wir beantworten wollen, ist: Warum sind das genau die Darstellungen, die unsere Maschine beim Zählen liefert?

Fällt eine Kugel in ein Flip-Flop, so ändert sich dessen Zustand (siehe Abb. 6). Ist der Zustand vorher 0, so ist er nachher 1 und die Kugel fällt ohne weitere Wirkung nach unten ins Ablagefach. Ist der Zustand vorher 1, so ist er nachher 0, die Kugel wird aber nach links unten ans nächste Flip-Flop weitergeleitet.

Wenn N Kugeln ein anfänglich zurückgesetztes Flip-Flop durchlaufen, wird die abgerundete Hälfte der Kugeln ans nächste Flip-Flop weitergegeben. Der Zustand des ersten Flip-Flops ist zum Schluss 0, wenn N gerade ist, und 1, wenn N ungerade ist.

Benutzen wir die mathematische Binärdarstellung von N von oben, so ist die abgerundete Hälfte von N gerade $(a \cdot \text{Zwei} + b) \cdot \text{Zwei} + c$ und d ist der Zustand des ersten Flip-Flops zum Schluss. Das erste Flip-Flop hat also das d aus der Binärdarstellung extrahiert. Das nächste Flip-Flop extrahiert

auf die gleiche Weise das c aus der übergebenen Kugelanzahl $(a \cdot \text{Zwei} + b) \cdot \text{Zwei} + c$ und leitet $a \cdot \text{Zwei} + b$ an das folgende Flip-Flop weiter. Dieses extrahiert daraus b und gibt a Kugeln an das vierte Flip-Flop weiter. Insgesamt zeigen also die Flip-Flops die Binärdarstellung $abcd$ von N an.

Ein Beispiel: Werden elf Kugeln durch den oberen Einlass in die zurückgesetzte Maschine gegeben, so befindet sich das oberste Flip-Flop zum Schluss im Zustand 1, da Elf ungerade ist. Fünf Kugeln werden an das zweite Flip-Flop übergegeben. Dieses hat zum Schluss den Zustand 1, weil Fünf ungerade ist. Zwei Kugeln werden an das dritte Flip-Flop weitergeleitet. Das befindet sich zum Schluss im Zustand 0, weil Zwei gerade ist. Eine Kugel wird an das vierte Flip-Flop übergeben, die es in den Zustand 1 versetzt. Die Binärdarstellung von Elf ist somit 1011.

Die zweite Frage, die wir beantworten wollen, ist: Warum kann man mit der Maschine die Binärdarstellungen zweier Zahlen addieren?

Hat man zwei Zahlen und ihre Binärdarstellungen

$$N_1 = a_1 \cdot \text{Zwei} \cdot \text{Zwei} \cdot \text{Zwei} + b_1 \cdot \text{Zwei} \cdot \text{Zwei} + c_1 \cdot \text{Zwei} + d_1, \\ N_2 = a_2 \cdot \text{Zwei} \cdot \text{Zwei} \cdot \text{Zwei} + b_2 \cdot \text{Zwei} \cdot \text{Zwei} + c_2 \cdot \text{Zwei} + d_2,$$

so erhält man deren Summe, indem man die entsprechenden Stellen addiert und die Überträge berücksichtigt, die entstehen, wenn beide Stellen den Wert 1 haben.

Zwei Beispiele:

$$\begin{array}{r} 0101 \\ + 0110 \\ = 1011 \end{array}$$

Hier entsteht nur ein Übertrag in der dritten Stelle.

$$\begin{array}{r} 0011 \\ + 0111 \\ = 1010 \end{array}$$

Hier entsteht ein Übertrag in der ersten, zweiten und dritten Stelle. An der zweiten

Stelle werden sogar drei Einsen addiert. Jedes Flip-Flop stellt nun einen 1-Bit-Addierer dar:

0 im Flip-Flop + 1 Kugel von oben
= 1 im Flip-Flop,

1 im Flip-Flop + 1 Kugel von oben
= 0 im Flip-Flop + Übertrag nach links.

Wenn die Kugeln alle gleichzeitig vom Eingaberegister ins Addierwerk gestoßen werden, erfolgt die Addition der entstehenden Überträge zeitversetzt. Daher bereitet die Addition der drei Einsen im zweiten Beispiel kein Problem. Man muss hier nicht wie bei elektronischen Addierern zwischen Halb- und Volladdierern unterscheiden.

Subtrahieren, Multiplizieren und Dividieren

Mit der Maschine lassen sich halbautomatisch auch Subtraktionen, Multiplikationen und Divisionen durchführen. Praktisch führt man die Subtraktion Elf minus Sechs durch, indem man zunächst Elf = 1011 ins Addierwerk eingibt. Die Sechs wird dann komplementär ins Eingaberegister eingegeben, d. h. man legt die Kugeln genau an den Stellen ins Eingaberegister, an denen die Binärdarstellung 0110 von Sechs Nullen hat, also links unten und rechts oben. Das Eingaberegister wird nun durch Hinunterstoßen addiert. Abschließend wird noch 0001 addiert. Man erhält 0101 + Überlauf. Der Überlauf wird ignoriert. Das Ergebnis ist also Fünf.

Dieses auf den ersten Blick etwas seltsam anmutende Vorgehen lässt sich mit den Begriffen Einerkomplement und Zweierkomplement erklären. Das Zweierkomplement einer Zahl N zwischen Null und Fünfzehn ist die Zahl N' , so dass $N + N' =$ Sechzehn ist. Zu Sechs = 0110 ist also zum Beispiel das Zweierkomplement Zehn = 1010. Wenn man nun zu einer Zahl M das Zweierkomplement N' von N addiert, addiert man Sechzehn $- N$ und erhält auf unserer Maschine dabei $M - N +$ Überlauf. Wenn man den Überlauf ignoriert, ist also

Subtraktion das gleiche wie Addition mit dem Zweierkomplement.

Schneller als das Zweierkomplement einer Zahl kann man im Kopf das Einerkomplement bilden, bei dem jede Binärstelle einzeln komplementiert wird. Die Summe einer Zahl und ihres Einerkomplements ergibt offensichtlich immer 1111 = Fünfzehn, da an jeder Stelle eine 1 und eine 0 addiert werden. Das Einerkomplement plus 1 ist also das Zweierkomplement. Damit ist das obige Vorgehen vollständig erklärt.

Die Multiplikation Fünf = 0101 mal Drei = 0011 führt man praktisch durch, indem man die Drei einmal unverändert eingibt und dann um zwei Stellen nach links versetzt (1100) dazu addiert. Man erhält 1111 = Fünfzehn. Mathematisch liegt das am Distributivgesetz: $0101 \cdot 0011 = (0100 + 0001) \cdot 0011 = 1100 + 0011$. Mit dieser Überlegung ist klar, dass man bei jedem beliebigen Produkt nur verschobene Kopien des zweiten Faktors addieren muss.

Die halbautomatische Division zweier Zahlen orientiert sich wie die Multiplikation an der schriftlichen Rechnung und nutzt damit ebenfalls das Distributivgesetz. Sie wird hier nicht weiter erklärt.

Schlussbemerkungen

In einer der nächsten Ausgaben werden weitere fischertechnik-Kugellogik-Elemente und Anwendungen vorgestellt, insbesondere ein R-S-Flip-Flop. Ein Video des hier vorgestellten Binäraddierers findet ihr in [youtube](#).

Literatur und Links

- [1] Georgi Dalakov: [Gottfried Leibniz](http://history-computer.com/).
- [2] Erwin Stein: [Die Leibniz-Dauerausstellung der Gottfried Wilhelm Leibniz Universität in Hannover](#). 2008.
- [3] John Thomas Godfrey: [Binary Digital Computer, US Patent 3,390,471](#)

Computing

Endliche Automaten in Robo Pro

Dirk Fox

Endliche Automaten sind ein zentrales Konzept der Informatik. Sie eignen sich besonders gut zur Modellierung autonomer IT-Systeme – wie z. B. Roboter oder Steuerungen. Die Programmierung Endlicher Automaten wird von Robo Pro hervorragend unterstützt – eine Einführung in Endliche Automaten sucht man in fischertechnik-Anleitungen jedoch vergebens. Das holen wir hiermit nach.

Hintergrund

Wer mit der Programmierung in Robo Pro beginnt, wird – auch durch die einführenden Beispiele in der ausführlichen Online-Hilfe – leicht dazu verleitet, Robo Pro-Programme grundsätzlich wie einen linearen „Programmablaufplan“ zu schreiben. Das ist zwar für viele einfache Aufgabenstellungen ein geeignetes Vorgehen, aber komplexere Steuerungsaufgaben lassen sich so nur sehr holprig verwirklichen. Das Ergebnis sind unübersichtliche, oft gar chaotische Programme. Sie stellen häufig wenig elegante Problemlösungen dar, bei der das Lösungskonzept schwer erkennbar und die Fehlersuche aufwändig ist. Vor allem aber bleiben sie weit hinter den Möglichkeiten von Robo Pro zurück – einer sehr mächtigen Programmierumgebung, die gerade für die Lösung von Steuerungsaufgaben besonders geeignet ist.

Diese „heimlichen“ Stärken von Robo Pro lassen sich besonders gut mit dem Konzept des ‚Endlichen Automaten‘ verstehen und erläutern. Daher starten wir mit einem kleinen Ausflug in die Welt der Endlichen Automaten – wem das bereits ein Begriff ist, der darf die beiden folgenden Abschnitte auch gerne überspringen. Anschließend werden wir die Modellierung und Programmierung in Robo Pro an Beispielen vorstellen.

Endliche Automaten

Als ‚Endlicher Automat‘ wird ein Modell bezeichnet, in dem das Verhalten eines Systems als eine endliche Menge von *Zuständen* beschrieben wird. Der Übergang von einem Zustand in einen anderen wird dabei von *Bedingungen* (wie z. B. Ereignissen) ausgelöst, auf die jeweils eine *Aktion* folgt. Endliche Automaten werden daher auch „Zustandsmaschinen“ genannt.

Endliche Automaten haben in der Informatik bei der Entwicklung von Compilern (‚Übersetzern‘) für Computersprachen eine wichtige Rolle gespielt. Aber auch eingebettete Systeme wie z. B. eine Aufzugsteuerung oder ein Roboter lassen sich hervorragend als Endlicher Automat modellieren (mehr dazu im nächsten Abschnitt).

In der Theoretischen Informatik wird das Konzept des Endlichen Automaten verwendet, um einige fundamentale Aussagen über Algorithmen zu formulieren und zu beweisen – damit wollen wir uns hier aber nicht beschäftigen. Wer mehr über die theoretischen Hintergründe wissen möchte, dem empfehle ich als Lektüre *das* Grundlagenwerk der Automatentheorie von Hopcroft und Ullman [1].

Für diesen Beitrag ist wichtig, dass sich prinzipiell jedes informationstechnische System – ein MP3-Spieler, ein Smartphone,

ein Computer – als Endlicher Automat beschreiben bzw. ‚modellieren‘ lässt. Dabei entsteht ein sehr klarer Überblick über das (gewünschte) Systemverhalten. Wird eine solche Modellierung allerdings zu komplex, stellt sie beim Systementwurf keine große Hilfe mehr dar; daher versucht man, komplexe Systeme zunächst auf einer höheren Abstraktionsebene als Endlichen Automaten zu modellieren, damit das Systemverhalten als Ganzes noch überschaubar bleibt.

fischertechnik-Automaten

Vor allem für die Aufgabenstellungen, die sich mit einem fischertechnik-Interface oder dem TX(T)-Controller lösen lassen, stellt die Modellierung als Endlicher Automat eine große Hilfe bei der Programmierung dar. Warum ist das so?

Ein Endlicher Automat – hier vielleicht anschaulicher Zustandsmaschine genannt – kennt nicht nur die Zustände, in die er, ausgelöst durch ein Ereignis wie ein bestimmter Sensorwert oder eine Tasterbetätigung, als nächstes übergeht, sondern auch die Zustände, die er zuvor eingenommen hat. Er verfügt also gewissermaßen über ein „Gedächtnis“. Damit können dieselben Sensoren (Fototransistoren, Reed-Kontakte etc.) abhängig vom Zustand des Systems unterschiedliche Aktionen veranlassen.

Bei Systemen, die komplett durch externe Ereignisse gesteuert werden, ist genau diese Eigenschaft wesentlich – denn in der Regel soll sich ein solches System bei Eintritt eines bestimmten Ereignisses nicht in jeder Situation (= Zustand) gleich verhalten. Beispielsweise soll die Schrankensteuerung einer Parkgarage einem neu eintreffenden Fahrzeug das Ziehen eines Parktickets nur dann ermöglichen, wenn es noch freie Parkplätze gibt. Und das Unterbrechen der Lichtschranke im Fahrstuhl soll nur dann zum Öffnen der Türe führen, wenn sie gerade schließt – aber natürlich nicht während der Fahrt.

Modellierung

Will man ein Programm zur Lösung einer bestimmten Aufgabenstellung entwickeln, sollte man sich immer zunächst (mindestens im Kopf, besser auf Papier) ein *Modell* von der Lösung machen, anstatt gleich mit der Programmierung zu beginnen. Erst, wenn das Modell auf Korrektheit geprüft ist, sollte man in die Tasten greifen. Wer das nicht tut, läuft leicht Gefahr, ein chaotisches Programm mit viel Flickwerk zu erzeugen – das er möglicherweise schon wenig später selbst nicht mehr so recht durchschaut.

In einem solchen Modell versucht man, die Aufgabenstellung zunächst in Teilprobleme zu zerlegen, die sich unabhängig voneinander lösen lassen. Dann zerlegt man diese Teilprobleme wiederum in unabhängige Teilprobleme – so lange, bis nur noch solche Teilprobleme übrig bleiben, die sich nicht mehr sinnvoll weiter zerlegen lassen. Anschließend löst man jedes Teilproblem Schritt für Schritt, indem man den Lösungsweg konkret beschreibt. Das Verfahren ähnelt der römischen Herrschaftsstrategie des *divide et impera* („teile und herrsche“) – und funktioniert ebenso erfolgreich.

Will man eine Problemlösung als Endlichen Automaten modellieren, muss man zunächst alle Zustände definieren, die das zu entwickelnde System annehmen soll, und in denen es einen ‚Auslöser‘ (= Bedingung) erwartet, um in einen anderen Zustand überzugehen und eine Aktion zu veranlassen. Dann wird einer dieser Zustände als *Startzustand S* des Endlichen Automaten gewählt. In diesem Zustand befindet sich das System direkt nach der Aktivierung.

Damit die Lösung auch terminiert (d. h. irgendwann endet) muss der Automat mindestens einen *Endzustand E* besitzen. Anders als beim Startzustand kann der Automat über beliebig viele Endzustände verfügen, aus denen er in keinen anderen Zustand mehr übergeht.

Dargestellt wird ein Endlicher Automat entweder grafisch als Zustandsübergangsdiagramm oder als –tabelle. Im *Zustandsübergangsdiagramm* werden Zustände als Kreise und Übergänge als Pfeile (mit Bedingung) zum jeweiligen Folgezustand dargestellt. In einer *Zustandsübergangstabelle* ordnet man allen Zuständen für jede Bedingung einen Folgezustand zu.

Außerdem ist noch festzulegen, ob eine Aktion vor, während oder nach einem Zustandsübergang ausgeführt wird. Je nach Aufgabenstellung kann die eine oder die andere Variante angemessener sein, auch wenn grundsätzlich eine Modellierung mit jeder der drei Varianten möglich ist. Die Unterscheidung hat aber Auswirkungen auf die Implementierung, wie wir noch sehen werden [3].

Im Zustandsübergangsdiagramm lässt sich diese Unterscheidung darstellen, indem die Aktion oben oder unten im Zustands-Kreis oder am Übergangs-Pfeil eingetragen wird. In einer Zustandsübergangstabelle notieren wir die Aktion für alle drei Varianten in Klammern hinter dem Eintrag des Folgezustands, in oder vor dem die Aktion ausgeführt werden soll.

Am besten lässt sich die Modellierung einer Problemlösung als Endlicher Automat an einem einfachen Beispiel illustrieren: einem Lichtschalter. Er besitzt zwei Zustände (Z_0 : „Aus“ und Z_1 : „An“) und geht bei Betätigung (= Bedingung) in den jeweils anderen Zustand über – und schaltet dabei den Stromfluss an bzw. aus (= Aktion).

Die Zustandsübergangstabelle ist einfach. Da man den Lichtschalter im „An“-Zustand nicht erneut an- und im „Aus“-Zustand nicht ausschalten kann, werden die entsprechenden Einträge in der Tabelle mit

„./.“ gekennzeichnet: Hier verbleibt der Automat im jeweiligen Zustand.

	anschalten	ausschalten
Z_0	Z_1 (an)	./.
Z_1	./.	Z_0 (aus)

Tab. 1: Zustandsübergangstabelle
„Lichtschalter“

Auch die Darstellung als Zustandsübergangsdiagramm ist simpel. Die beiden Fälle, in denen der Schalter seinen Zustand behält, werden im Diagramm als Übergang in denselben Zustand gekennzeichnet. Natürlich könnte man die beiden Pfeile auch weglassen; allerdings hilft diese Darstellung dabei, zu prüfen, ob man im Diagramm keinen Übergang vergessen hat: Denn aus jedem Zustand muss für jedes Ereignis ein Pfeil (= Übergang) abgehen.

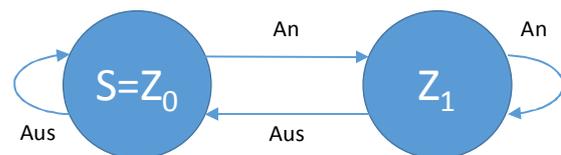


Abb. 1: Zustandsübergangsdiagramm
„Lichtschalter“

In diesem einfachen Beispiel ist es irrelevant, ob die Aktion (Licht ein, Licht aus) vor, beim oder nach dem Übergang in den Folgezustand ausgeführt wird. Für unser Zustandsübergangsdiagramm haben wir die Variante „während des Übergangs“ gewählt. Bei diesem Endlichen Automaten kann jeder der beiden Zustände als Startzustand gewählt werden; wir haben uns für den „Aus“-Zustand Z_0 entschieden. Einen Endzustand besitzt der Automat nicht – das bedeutet, dass die Lösung nicht ‚terminiert‘.²

² Bei einem strengen Begriffsverständnis ist die Lösung damit gar kein Algorithmus – denn der ist eine *terminierende* Lösungsvorschrift. Viele Systemsteuerungen („eingebettete Systeme“)

sollen jedoch ohne Unterbrechung funktionieren und besitzen daher keinen Endzustand; sie können höchstens abgebrochen werden.

Ist die Aufgabenstellung komplexer, kann es sinnvoll sein, die Modellierung auf unterschiedlichen Abstraktionsebenen vorzunehmen. So kann man zunächst mit einer groben Unterscheidung von Zuständen beginnen, die sich später jeweils wiederum als eigene Endliche Automaten darstellen lassen. Auf diese Weise kann durch eine schrittweise Konkretisierung ein Teilproblem nach dem anderen präzisiert werden, ohne dass dabei die Systematik der Gesamtlösung (der ursprüngliche, abstrakte Endliche Automat) verändert werden muss.

Programmierung in Robo Pro

Die Unterprogramme in Robo Pro bieten eine hervorragende Möglichkeit, ein als Endlicher Automat modelliertes System übersichtlich und anschaulich zu programmieren: sie können nämlich mehrere Ausgänge besitzen.

Damit lässt sich in Robo Pro jeder Zustand des Zustandsübergangsdiagramms als Unterprogramm realisieren, das – abhängig von unterschiedlichen Ereignissen – über einen anderen Ausgang verlassen wird und von dort in den Folgezustand übergeht, sprich: ein anderes Unterprogramm aufruft.

Das Hauptprogramm entspricht damit dem Zustandsübergangsdiagramm, die Unterprogramme bilden die Zustände, die Aufrufe der Unterprogramme stellen die Zustandsübergänge dar. Die Endzustände werden durch das Programmende-Symbol (rotes Ampelmännchen) realisiert.

Beispiele

Wie hilfreich die Modellierung und Programmierung einer Aufgabenstellung als Endlicher Automat in der Praxis ist, möchten wir im Folgenden an drei Beispielen verdeutlichen: der Steuerung einer Treppenhausbeleuchtung, der Programmierung eines Spurfolger-Roboters und der Realisierung eines Morsezeichen-Dekodierers.

Treppenhausbeleuchtung

Wir beginnen mit einem einfachen Beispiel. Eine Treppenhausbeleuchtung soll sowohl durch einen Bewegungsmelder als auch durch einen Taster für eine festgelegte Zeit (eine Minute) aktiviert werden können. Der Bewegungsmelder soll allerdings nur funktionieren, wenn es im Treppenhaus hinreichend dunkel ist. Für diese Steuerungsaufgabe können wir die folgenden Zustände unterscheiden:

- *Zustand 0*: Die Treppenhausbeleuchtung und der Bewegungsmelder sind inaktiv (Startzustand).
- *Zustand 1*: Die Treppenhausbeleuchtung ist inaktiv, aber der Bewegungsmelder ist aktiviert.
- *Zustand 2*: Die Treppenhausbeleuchtung ist aktiv.

Die Übergänge zwischen diesen drei Zuständen werden durch vier Ereignisse gesteuert:

- einen Dämmerungssensor (Fototransistor), der einen festgelegten Wert über- oder unterschreitet,
- einen Taster, mit dem die Treppenhausbeleuchtung aktiviert werden kann und
- einen Infrarot-Bewegungsmelder, der bei einer Bewegung im Raum schaltet.

Die eigentliche Aktion – das Aktivieren der Treppenhausbeleuchtung für eine vorgegebene Zeit – findet im Zustand Z_2 statt. Die beiden anderen Zustände warten nur auf das Eintreten der Ereignisse. Damit lässt sich die folgende Zustandsübergangstabelle bilden:

	hell	dunkel	Taster	Bewegung
Z_0	./.	Z_1	Z_2	./.
Z_1	Z_0	./.	Z_2	Z_2
Z_2	Z_0	Z_1	./.	./.

Tab. 2: Zustandsübergangstabelle
Treppenhausbeleuchtung

Im nachfolgenden Zustandsübergangsdiagramm ist die Aktion im Zustandskreis Z_2 eingetragen (Abb. 2).

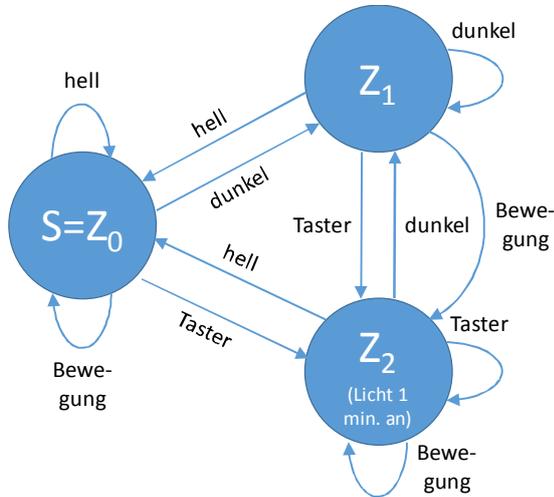


Abb. 2: Treppenhausbeleuchtung als Endlicher Automat

Die Robo Pro-Realisierung der Zustände Z_0 und Z_1 ist einfach – hier werden nur die Werte der relevanten Sensoren abgefragt (I1: Taster, I2: Bewegungssensor, I3: Helligkeitssensor).

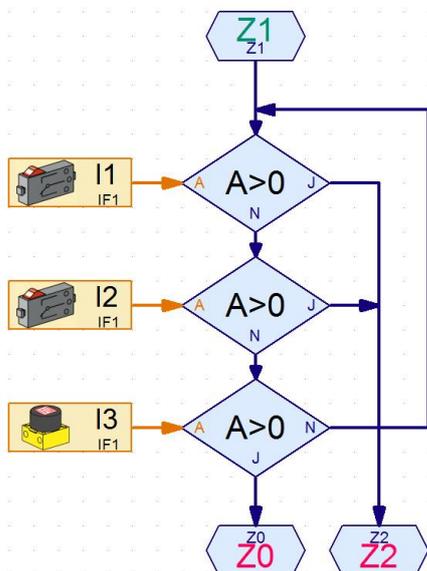


Abb. 3: Robo Pro-Realisierung von Z_1

Der Zustand Z_2 enthält zusätzlich die Aktivierung der Treppenhausbeleuchtung für die Dauer von einer Minute, bevor der Dämmerungssensor ausgewertet wird, der

einzig für die Übergänge aus Z_2 relevante Sensor (Abb. 4).

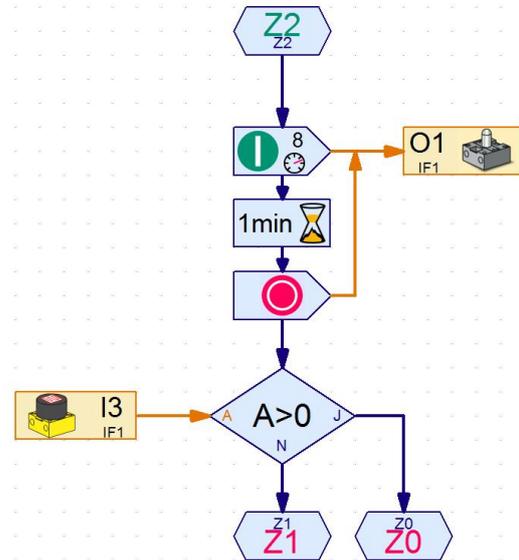


Abb. 4: Robo Pro-Realisierung von Z_2

Das Hauptprogramm bildet den Zustandsgraphen aus Abb. 2 ab – bis auf die Übergänge ohne Zustandswechsel, die wir im Programm weglassen können (Abb. 5).

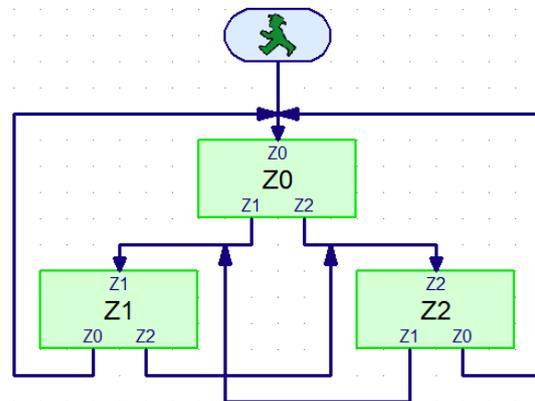


Abb. 5: Hauptprogramm Treppenhausbeleuchtung

Wer diese kleine Steuerung in einem Modell verwenden möchte, wird sich fragen, wie man mit fischertechnik den Bewegungsmelder realisiert. Denn es gibt keinen solchen fischertechnik-Sensor. Also muss man auf einen Fremdsensor zurückgreifen.

Tatsächlich gibt es zahlreiche PIR (Passive Infrared) Sensoren, die auf Bewegungen in vier bis zehn Meter Entfernung reagieren

und einen Umgebungswinkel von 90-180° abdecken – und die direkt an einen TX(T)-Controller angeschlossen werden können. Ein solcher Sensor ist der [GH-718C](#), der bei Bewegungserkennung (7 m, 110°) einen 3,3 V-Pegel liefert und im Elektronikhandel für unter 8 € erhältlich ist (Abb. 6).

Beim Anschluss an den TX(T)-Controller wird in Robo Pro als ‚Eingangsart‘ 10 V digital gewählt; dann liegt am Eingang eine ‚1‘ an, wenn der Sensor eine Bewegung detektiert hat.



Abb. 6: PIR-Sensor GH-718C

Wer keinen Fremdsensor einsetzen möchte, kann stattdessen auch eine Lichtschranke (einen Fototransistor mit Linsenlampe) oder den fischertechnik-Ultraschall-Abstandssensor verwenden.

Spurfolger (Line Follower)

Eine beliebte Aufgabe für autonome Roboter ist die Programmierung eines „Spurfolgers“ (oder „Line Followers“) – eines Roboters, der mittels Infrarotsensoren einer durchgezogenen Linie auf dem Boden folgt. Diese Aufgabe ist geradezu prädestiniert für eine Modellierung als Endlicher Automat. Die benötigten Zustände sind offensichtlich:

- *Zustand 0*: Roboter befindet sich auf der Linie
- *Zustand 1*: Roboter befindet sich rechts neben der Linie
- *Zustand 2*: Roboter befindet sich links neben der Linie

- *Zustand 3*: Roboter hat die Linie verloren

Auch die Aktionen in den jeweiligen Zuständen sind unmittelbar klar: im Zustand Z_0 muss der Roboter geradeaus fahren, im Zustand Z_1 muss er nach links drehen, im Zustand Z_2 nach rechts. Hat der die Linie verloren (Zustand Z_3), benötigt er einen geeigneten Suchalgorithmus, der ihn wieder auf die Linie zurückbringt.

Gesteuert wird der Spurfolger von den Werten zweier Infrarot-Sensoren. Damit der Roboter die Zustände Z_1 bis Z_3 unterscheiden kann, benötigen wir zwei relativ dicht nebeneinander liegende Sensoren, wie sie im fischertechnik-Spursensor verbaut sind. Damit können wir nun auch die Bedingungen formulieren, die zu einem Zustandsübergang führen:

- beide Sensoren liefern den Wert 0: der Sensor befindet sich direkt über der Linie
- der rechte Sensor liefert den Wert 1, der linke den Wert 0: der Sensor liegt leicht rechts von der Linie
- der rechte Sensor liefert den Wert 0, der linke den Wert 1: der Sensor liegt leicht links von der Linie
- beide Sensoren liefern den Wert 1: der Sensor hat die Linie verloren

Die Zustandsübergangstabelle ist nun ebenfalls schnell erstellt:

	00	01	10	11
Z_0	./.	Z_1 (links)	Z_2 (rechts)	Z_3 (Suche)
Z_1	Z_0 (gerade)	./.	Z_2 (rechts)	Z_3 (Suche)
Z_2	Z_0 (gerade)	Z_1 (links)	./.	Z_3 (Suche)
Z_3	Z_0 (gerade)	Z_1 (links)	Z_2 (rechts)	./.

Tab. 3: Zustandsübergangstabelle Spurfolger (Line Follower)

Die angegebenen Aktionen steuern die Richtung der Motoren. Zusätzlich könnte man das „Gedächtnis“ des Endlichen Automaten nutzen, um die Geschwindigkeit zu steuern: So ließe sich beispielsweise bei jedem Übergang von Zustand Z_0 in Z_0 (der Roboter fährt exakt auf der Linie) die Geschwindigkeit Schritt für Schritt bis zu einem Maximalwert erhöhen – und bei einem Abweichen von der Linie sofort auf die Minimalgeschwindigkeit setzen.

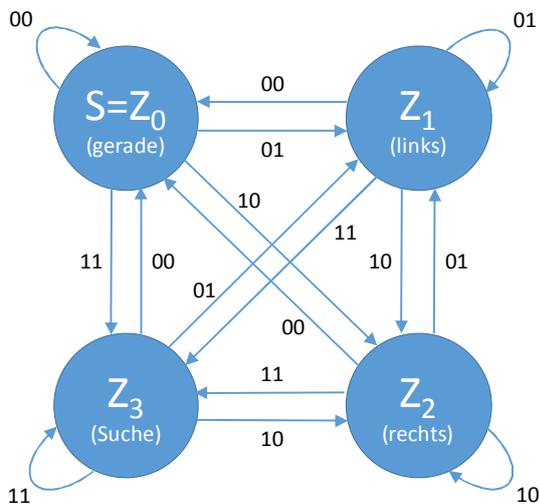


Abb. 7: Spurfolger (Line Follower) als Endlicher Automat

Aus der Zustandsübergangstabelle lässt sich das Zustandsübergangsdiagramm ableiten (Abb. 7). Die Aktionen werden während des Zustandsübergangs ausgeführt. Den Zustand Z_0 haben wir als Startzustand festgelegt.

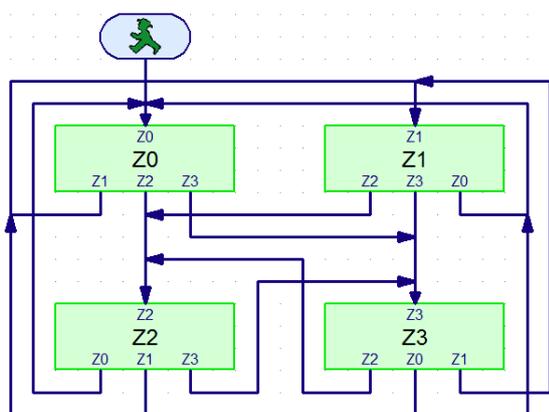


Abb. 8: Hauptprogramm des Spurfolgers

Die Umsetzung dieses Automaten in ein Robo Pro-Programm ist sehr einfach. Das Hauptprogramm (Abb. 8) ist ein ziemlich exaktes Abbild des Zustandsübergangsdiagramms aus Abb. 7.

Die Realisierung der vier Zustände ist ebenfalls sehr einfach: Zu Beginn wird die Steuerungs-Aktion (links, rechts, gerade) ausgeführt; anschließend werden die Werte der beiden Spursensoren für die richtigen Zustandsübergänge ausgewertet (Abb. 9).

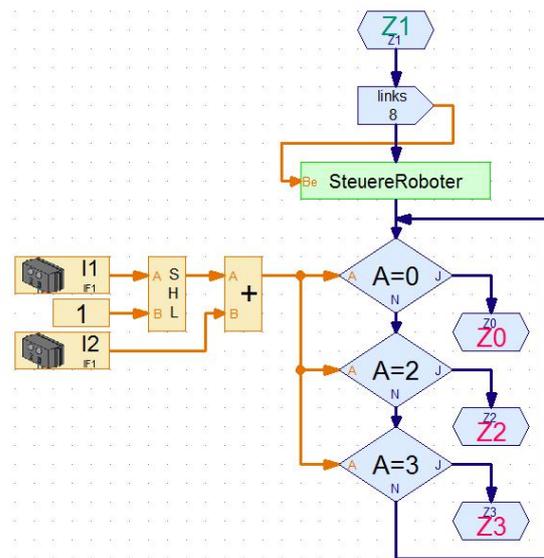


Abb. 9: Zustand Z_1 des Spurfolgers

Das Unterprogramm „SteuereRoboter“ ist nichts anderes als das in der Online-Hilfe von Robo Pro im Kapitel 6.1 beschriebene Beispiel für die Steuerung eines Roboters mit zwei Antriebsmotoren (ohne „Stopp“ und „zurück“). Natürlich kann hier auch ein eigenes Programm für die Steuerung des Roboters eingesetzt werden.

Im Zustand Z_3 wird keine Motorsteuerung ausgelöst, sondern parallel ein Linien-Suchprogramm gestartet. Mit Spurfolger-Robotern werden wir uns in einer der folgenden ft:pedia-Ausgaben noch ausführlicher beschäftigen, daher betrachten wir den Suchalgorithmus bei verlorener Linie hier nicht genauer. Jedoch kann man sich leicht vorstellen, dass sich dieser Algorithmus wieder-

rum als eigener Endlicher Automat modellieren lässt – ein schönes Beispiel für eine schrittweise Verfeinerung.

In vielen Spurfolger-Aufgabenstellungen gibt es auch eine Ende-Bedingung (Ziel erreicht), für die ein (zusätzlicher) Endzustand definiert werden kann.

Morsezeichen-Dekodierer

Als drittes Beispiel möchten wir einen Morsezeichen-Dekodierer realisieren, der zu empfangenen Morsesignalen die zugehörigen Buchstaben ausgibt. Dabei beschränken wir uns auf die sechs Buchstaben „a“, „e“, „i“, „m“, „n“ und „t“, die sich mit maximal zwei Morsesignalen darstellen lassen [2] (Abb. 10).

a	● —
e	●
i	● ●
m	— —
n	— ●
t	—

Abb. 10: Morse-Codes der mit max. zwei Signalen darstellbaren Zeichen

Dazu nehmen wir zunächst an, dass ein gegebener Empfänger lange und kurze Signale unterscheiden kann und ein Zeichenende (lange Pause) erkennt. Damit haben wir drei verschiedene Bedingungen, die wir unterscheiden müssen, um unseren Dekodierer zu steuern:

- ein kurzes Zeichen wurde empfangen
- ein langes Zeichen wurde empfangen
- ein Zeichenende (lange Pause) wurde empfangen.

Weiter kann unser Morsezeichenempfänger vier verschiedene Zustände annehmen:

- *Zustand 0 (Startzustand)*: Warte auf Zeichenende (lange Pause)
- *Zustand 1*: Warte auf nächstes Zeichen
- *Zustand 2*: Warte auf nächstes Signal, nachdem ein kurzes empfangen wurde

- *Zustand 3*: Warte auf nächstes Signal, nachdem ein langes empfangen wurde

Der Automat startet im *Startzustand*. Empfängt er eine lange Pause, geht er in *Zustand 1* über und wartet auf das nächste Zeichen. Empfängt er ein kurzes Signal, geht er in *Zustand 2* über, empfängt er ein langes in *Zustand 3*. Bei einem Zeichenende-Signal bleibt er im *Zustand 1*.

Aus dem *Zustand 2* geht er beim nächsten Signal wieder in den Startzustand über und gibt bei einem kurzen Zeichen „i“ aus, bei einem langen „a“. Bei einem Zeichenende-Signal gibt er „e“ aus und geht in *Zustand 1* über. Aus dem *Zustand 3* geht er beim nächsten Signal wieder in den Startzustand über und gibt bei einem kurzen Zeichen „n“ aus, bei einem langen „m“. Bei einem Zeichenende-Signal gibt er „t“ aus und geht in *Zustand 1* über.

	kurz	lang	Pause
Z ₀	.i.	.l.	Z ₁
Z ₁	Z ₂	Z ₃	.l.
Z ₂	Z ₀ („i“)	Z ₀ („a“)	Z ₁ („e“)
Z ₃	Z ₀ („n“)	Z ₀ („m“)	Z ₁ („t“)

Tab. 4: Zustandsübergangstabelle Morsezeichen-Dekodierer

Notieren wir die Zustände und die Bedingungen für die Zustandsübergänge zunächst in einer Zustandsübergangstabelle (Tab. 4).

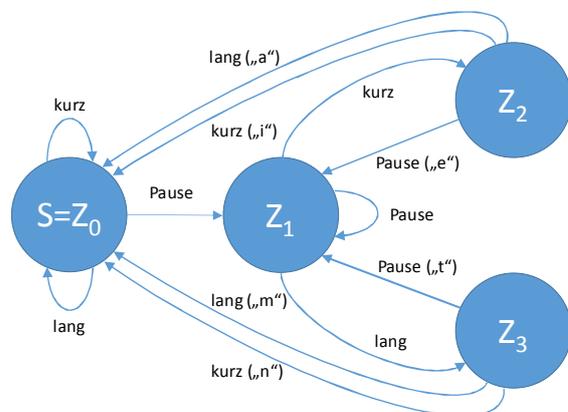


Abb. 11: Morsezeichen-Dekodierer als Endlicher Automat

Dazu lässt sich leicht ein Zustandsübergangsdiagramm zeichnen, das den Zustandsautomaten sehr übersichtlich darstellt (Abb. 11). Die Zustände lassen sich sehr einfach als Robo Pro-Unterprogramm realisieren. Abb. 12 zeigt den Zustand Z₃, in dem die Signale ausgewertet werden.

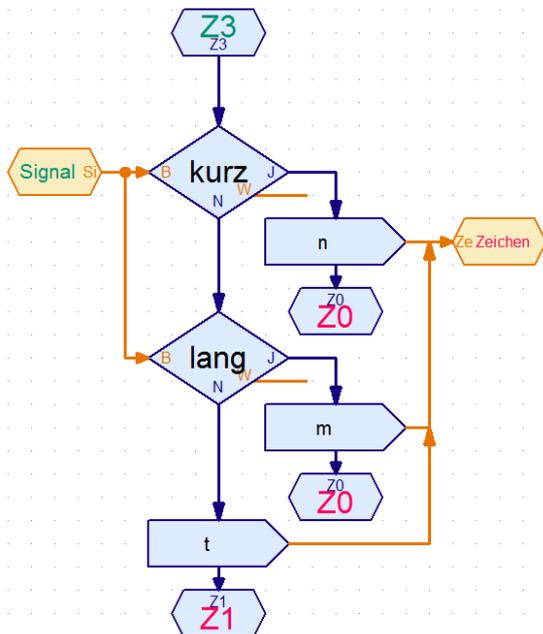


Abb. 12: Zustand Z₃ des Morsezeichen-Dekodierers in Robo Pro

Im Hauptprogramm erkennt man den Endlichen Automaten gut wieder. Das empfangene Morsesignal (,kurz‘, ,lang‘ oder ,Pause‘) wird an die gerade aktiven Zustände übergeben und dort ausgewertet; aus den Zuständen Z₂ und Z₃ wird mit dem nächsten Zustandsübergang jeweils das dekodierte Zeichen ausgegeben (Abb. 13).

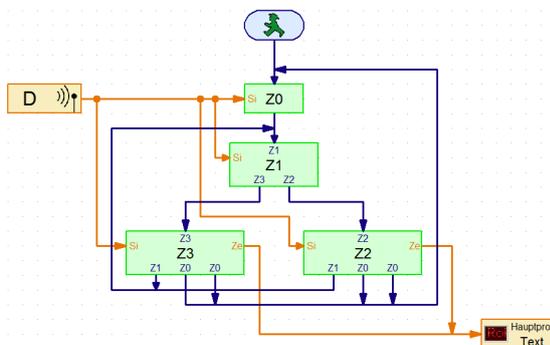


Abb. 13: Morsezeichen-Dekodierer; Hauptprogramm in Robo Pro

Diesen noch recht einfachen Endlichen Automaten kann man leicht zu einem vollwertigen Morsezeichen-Decoder weiterentwickeln – das sei euch als kleine Fingerübung überlassen. Beachtet dabei, dass man für eine saubere Lösung auch noch einen Fehlerzustand benötigt; schließlich kann es zu Übertragungsfehlern kommen.

Fazit

Die Modellierung einer Steuerungsaufgabe als Endlicher Automat hat einige wichtige Vorzüge: Die gewünschten Abläufe werden durch die Definition von Zuständen und Übergängen sauber und unzweideutig definiert. Auch lässt sich anhand eines Zustandsübergangsdiagramms sehr leicht überprüfen, dass keine Bedingungen übersehen wurden, denn aus jedem Zustand muss für jedes mögliche Ereignis genau ein Übergangspfeil abgehen. Allerdings kann es bei einer komplexen Aufgabenstellung herausfordernd sein, die richtige Abstraktionsebene für das Zustandsübergangsmodell zu finden.

Die Implementierung solcher Endlicher Automaten wird in Robo Pro hervorragend unterstützt – die Zustände lassen sich als Unterprogramm realisieren, die Übergänge durch Verbindungslinien im Hauptprogramm. Wer ein sauber konzipiertes und übersichtliches Steuerungsprogramm anstrebt, für den ist eine Modellierung als Endlicher Automat eine große Hilfe.

Quellen

- [1] John E. Hopcroft, Jeffrey Ullman: *Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie*. Addison-Wesley, 1990.
- [2] Dirk Fox: *Morsetelegraf*. ft.pedia/1/2014, S. 4-10.
- [3] Ferdinand Wagner, Ruedi Schmuki, Thomas Wagner, Peter Wolstenholme: *Modeling Software with Finite State Machines. A Practical Approach*. CRC Press, 2006.

Modell

Detail Engineering R2D3 (3) – IR-Fernbedienung am Robo TX Controller

Andreas Gail

Im Rahmen des Baus des [Robotermodells R2D3](#) wurde eine Reihe von ganz unterschiedlichen Detaillösungen erarbeitet, die durchaus Lösungsansätze bei diversen anderen Bauprojekten sein könnten. Teil 3 der Beitragsreihe stellt vor, wie die IR-Fernbedienung als Eingangsgröße am Robo TX Controller betrieben werden kann.

Aufgabenstellung

Das Robotermodell R2D3 (Abb. 1) sollte neben der Bedienung über PC gleichzeitig auch über die IR-Fernbedienung von fischertechnik steuerbar sein.

Bei Anwendung des neuen angekündigten Controllers von fischertechnik ist nachfolgendes Vorgehen hinsichtlich der IR-Fernbedienung nicht mehr erforderlich, wohl aber kann es durchaus weiterhin ähnlich gelagerte grundsätzliche Problemstellungen geben.

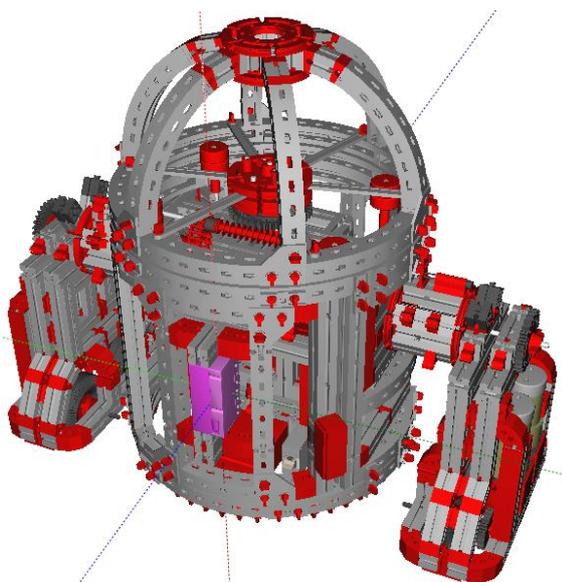


Abb. 1: Roboter mit IR-Empfängermodul
(violett dargestellt)

Lösung

Beispielhaft genutzt wird der linke Bedienhebel der Fernbedienung.

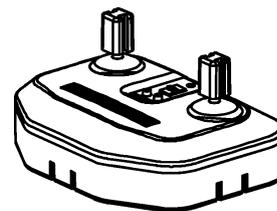


Abb. 2: IR Sender

Betrachtet man den Fernbedienhebel genauer, sieht man, dass mit diesem einen Hebel acht verschiedene Grundstellungen möglich sind, die individuell als Eingangsgröße für den Robo TX Controller („TX“) genutzt werden können. Zwischenstellungen sind dabei nicht relevant, es wird jeweils nur die prinzipielle Position ausgewertet.

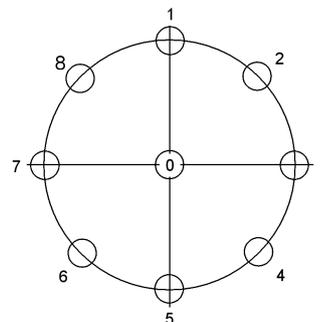


Abb. 3: Grundstellungen eines Bedienhebels
der IR-Fernbedienung

Entsprechend der Einstellung am linken Bedienhebel werden zugehörige Signale zum IR-Empfängermodul gesendet und über die Anschlusspaare M1 und M2 ausgegeben.

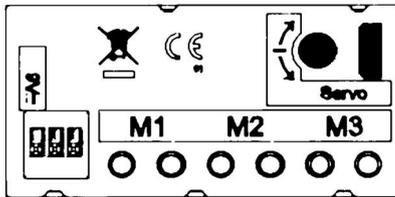


Abb. 4: IR Empfänger Modul

Bei der Betrachtung des IR-Empfängermoduls fällt auf, dass es für den rechten Bedienhebel nur noch das Anschlusspaar M3 gibt und einen Steckanschluss für einen Servo. Somit ist der rechte Bedienhebel nur sehr begrenzt in der hier gezeigten Weise nutzbar. Die Umwandlung des Servo-Signals in ein Signal analog M1-M3 ist nur mit hohem Aufwand möglich und wird daher hier nicht weiterverfolgt.

Verschaltung

Mit diesen Bausteinen ist es möglich, bis zu sechs digitale Eingänge des TX kommend von der IR-Fernbedienung anzusteuern. Im nachfolgenden Fall werden vier Eingänge genutzt. Am IR-Empfänger können dabei jeweils beide Ausgänge von M1 und M2 genutzt werden.

Zur Ansteuerung werden zusätzlich Widerstände mit $330\ \Omega$ ($1/4\ W$) benötigt. Diese Widerstände haben den Farbcode orange-orange-braun. Zur Verschaltung werden die Eingänge I1-I4 des TX über die Widerstände mit Masse verbunden; dadurch werden die Eingänge permanent gesetzt. Die vier Eingänge I1-I4 werden gleichzeitig mit dem IR-Empfänger M1a, M1b, M2a und M2b verbunden. Im Fall der Ansteuerung eines Ausgangs am IR-Empfänger wird dieser mit $+9\ V$ belegt, was zur Rücksetzung des zugehörigen Eingangs am TX führt.

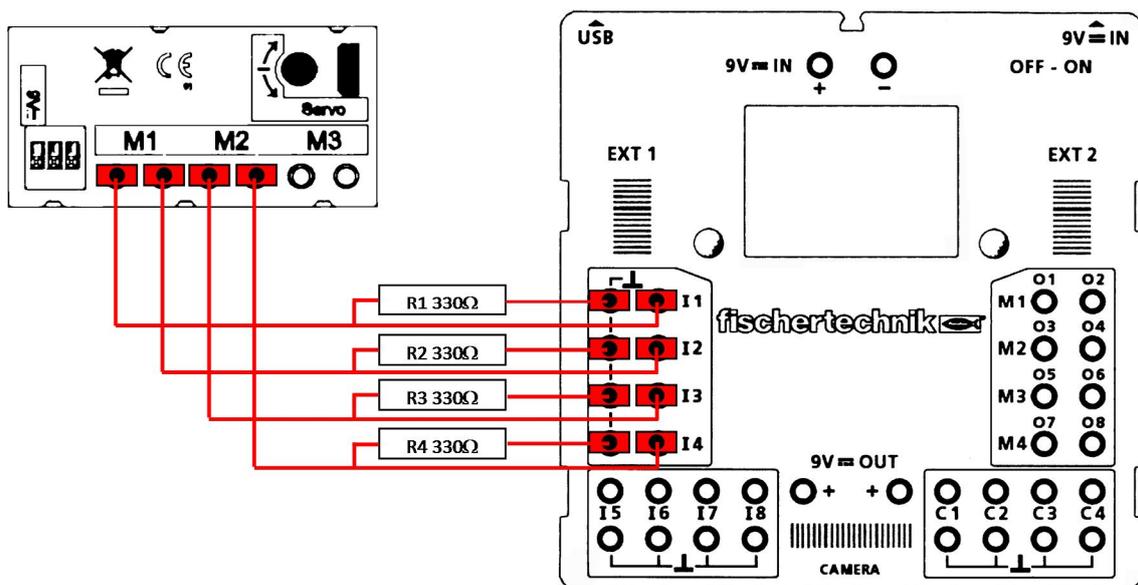


Abb. 5: Verschaltung der IR Fernbedienung mit dem TX

Signale an M1 und M2

Entsprechend der Hebelstellung an der IR-Fernbedienung ergeben sich am IR-Empfängermodul folgende Signale (invers dargestellt):

M1b An2 In1	M2a An3 In2	M2b An4 In3	M1a An1 In4	Beschreibung
0	0	0	0	Pos = 0 Hebel Grundstellung mittig
1	0	0	0	Pos = 1 Hebel oben
1	1	0	0	Pos = 8 Hebel oben-links
1	0	1	0	Pos = 2 Hebel oben-rechts
0	1	0	0	Pos = 7 Hebel links
0	0	1	0	Pos = 3 Hebel rechts
0	0	0	1	Pos = 5 Hebel unten
0	1	0	1	Pos = 6 Hebel unten-links
0	0	1	1	Pos = 4 Hebel unten-rechts

Tab. 1: IR-Empfängermodul-Signale in Abhängigkeit der Hebelstellung der IR-Fernbedienung

Software

Da die vom IR-Empfänger abgegebenen Signale ungenau sind und teilweise flackern (auch wenn diese gar nicht angesteuert werden) ist eine Kompensation mithilfe der vorliegenden Software erforderlich. Hierbei werden bei 3.000 Durchläufen die Eingänge des TX ausgelesen.

Nur wenn bei diesen 3.000 Durchgängen 55% (bzw. 60%) der Durchgänge eine Ansteuerung angezeigt haben, wird davon ausgegangen, dass tatsächlich eine gewollte Bedienung vorliegt.

Der Eingang I1 (M1a-An1-In2) zeigte ein etwas abweichendes Verhalten von den anderen Eingängen, so dass in diesem Fall 60% der Ansteuerungen vorgelegen haben müssen, um diese als ‚gewollt‘ zu werten. Im vorliegenden Fall läuft die Software auf einem PC und wurde nicht in den TX geladen. Die 3.000 Durchläufe führen dabei zu einer Ansprechverzögerung von etwa 0,5 s.

Zu Beginn der Schrittkette wird zunächst die Variable *Zaehler* = 0 gesetzt sowie anschließend *An1-An4* = 0. Nachfolgend wird geprüft, ob im Hauptprogramm der Schalter *IR* aktiviert wurde; wenn nein werden die globalen Variablen *In1-In4* = 0 gesetzt und das Unterprogramm wird verlassen. Dann ist die Bedienung über die IR-Fernbedienung abgeschaltet.

Wenn der Schalter *IR* aktiviert wurde, beginnt ein 3.000-facher Schleifendurchlauf; Schleifenzähler ist die Variable *Zaehler*. Bei jedem Schleifendurchlauf wird geprüft, wie die Einstellung der Eingänge *I1-I4* beschaltet sind, entsprechend wird die Variable *An1-An4* bei jedem Durchlauf um eins erhöht bzw. nicht erhöht. Nach 3.000 Schleifendurchläufen beginnt die Auswertung und Anpassung der globalen Variablen *In1-In4*. Wenn z. B. bei mehr als 1.650 von 3.000 (55%) Schleifendurchläufen eine Signalaufschaltung erkannt wurde, so wird die zugehörige globale Variable *In1-In4* = 1 gesetzt. Wenn die Einstellung *In1-In4* geändert wurde, wird dies im Bedienfeld des Hauptprogramms visualisiert. Die Auswertung von *In1-In4* erfolgt in anderen Programmteilen.

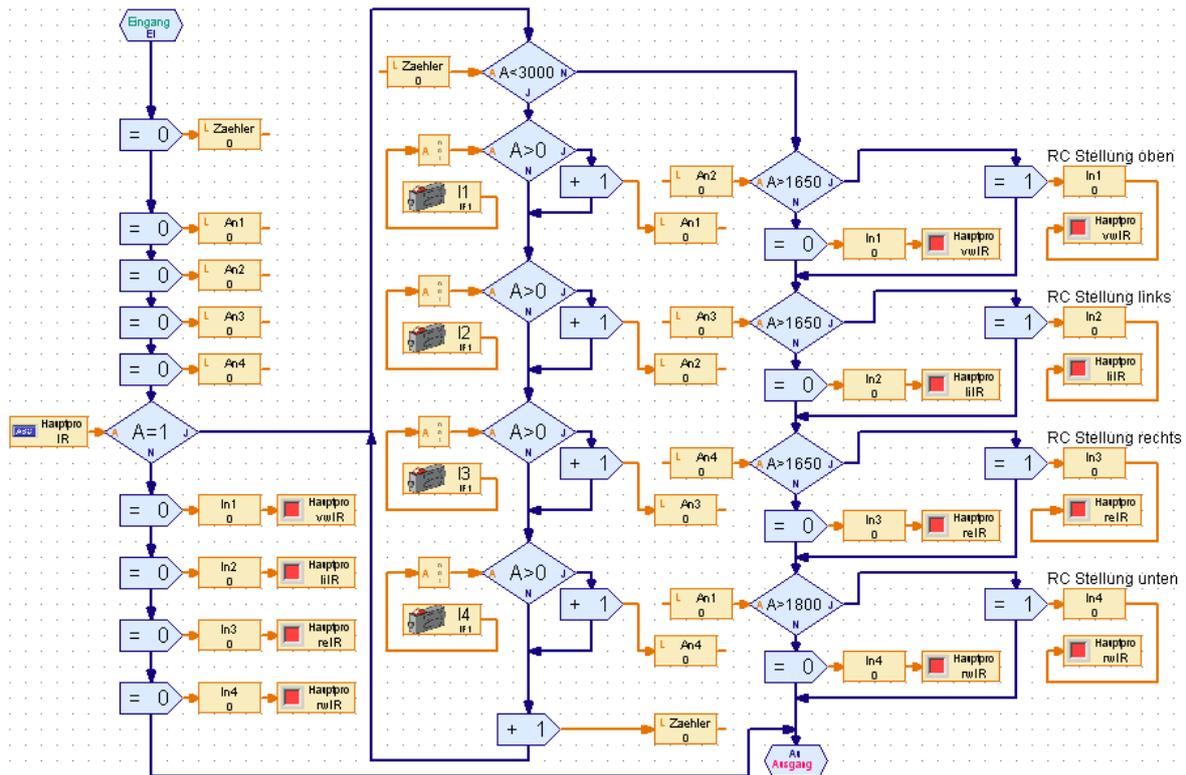


Abb. 6: TX-Software, Unterprogramm zur Auswertung der IR-Eingangssignale mit individualisierter Signalaufbereitung

Elektronik

LEDs in Leuchtsteinen

Gerhard Birkenstock

Wenn man heute eine neue Lampe kauft, ist mit hoher Wahrscheinlichkeit LED-Technik eingebaut. Dank des guten Wirkungsgrades und ihrer langen Lebensdauer stoßen LEDs in alle Bereiche des täglichen Lebens vor. Dieser Beitrag zeigt, wie man auf elegante Weise von der ft-Glühlampe zur ft-LED in den Leuchtsteinen umstellen kann.

Ärger mit den ft-Glühlampen

Kennst du das? Man hat ein ft-Modell gebaut. Ein selbst steuerndes Fahrzeug mit einem Lichtsensor in Richtung Fahrbahn, welches einer Bodenmarkierung folgt. Oder eine Fertigungsstraße mit Lichtschranken und Farbsortierungen. Und dann... auf einmal fährt das Fahrzeug einfach weg, rast unter die Nähmaschine und versucht, mit durchdrehenden Antriebsrädern die Hauswand zu versetzen. Und bei der Fertigungsstraße... fliegen die zu bearbeitenden Teile völlig unkontrolliert vom Transportband. Alles fällt in einen Topf. Von Bearbeiten und Sortieren ist keine Rede mehr.

Nach wenigen Sekunden der Fehlersuche steht fest: es ist mal wieder eine ft-Glühlampe ausgefallen. Der Glaskörper ist einfach schwarz. Durchgebrannt.

Bei mir ging der Frust so weit, dass ich schon keine Modelle mit Lampe mehr gebaut habe. Ich wollte einfach nicht schon wieder eine der ft-Glühlampen verwenden. Es musste eine Lösung her.

Lösungsfindung und Idee

Es sollte etwas Leuchtendes sein, was nicht stetig durchbrennt, sondern eine höhere Lebensdauer hat. Also eine LED!

Auch im Jahre 2007, in dem ich diese Gedanken alle hatte, hatten die LEDs schon eine gute Lichtausbeute. Es war der Anfang

des heute schon allgegenwärtigen Siegeszuges der LED-Technologie. Somit war der Gedanke schnell gefasst. Einziges Problem: LEDs wollen Gleichstrom. Wenn man sie verpolt, sind sie sehr schnell genauso defekt wie die Version mit Glühfaden. Hier musste Abhilfe geschaffen werden. Der LED-Ersatz musste gleiche oder bessere Eigenschaften wie die Lampen haben.

Es musste eine Schaltung her, welche den Strom an der LED immer in der gleichen Richtung anbietet, egal, wie ich außen die Kontakte anschließe: Die [Graetz-Brücke](#) [1]. Zudem wird ein sogenannter Vorwiderstand zur StromEinstellung zwingend benötigt (ft:pedia 2/2011 [2]).

Nach diesen Überlegungen stand fest:

- eine LED,
- ein Widerstand und
- vier Dioden

müssen verwendet werden. Um das nicht fliegend verdrahten zu müssen, wurde auch eine Platine notwendig. Um Bauraum zu sparen, sollte alles in [SMD-Technik](#) (*Surface Mounted Device*, zu Deutsch: Oberflächenmontage [3]) ausgeführt werden.

LED mit Vorwiderstand: Berechnung

Die zu verwendeten Bauteile müssen nun berechnet werden.

Eine LED, welche gelblich-weißes Licht abgibt, benötigt eine Spannung von ca. 1,8 V. Jede der Dioden für den Gleichrichter hat eine Schwellspannung von ca. 0,7 V. Wie in den Bildern zu erkennen, sind immer zwei Dioden in Durchlassrichtung geschaltet.

Weiterhin muss man noch etwas über die viel verwendeten ft-Transformatoren wissen. Es wird immer angegeben, dass mit 9 V gearbeitet wird. Sobald ein Motor als Last angeschlossen wird, ist dies auch richtig. Wird jedoch nur wenig Strom von einem Transformator abgenommen, beträgt die Spannung gut 12 V. Daher muss bei der Berechnung der Komponenten mit der maximalen Spannung gerechnet werden.

Die Spannung am Vorwiderstand ergibt sich nun aus den bekannten Informationen:

$$U_{\text{Widerstand}} = U_{\text{Netzteil}} - U_{\text{LED}} - 2U_{\text{Diode}}$$

$$U_R = 12 \text{ V} - 1,8 \text{ V} - 2 \cdot 0,7 \text{ V} = 8,8 \text{ V}$$

Es ist natürlich energietechnisch nicht gerade toll, mehr als das Vierfache der elektrischen Energie am Widerstand in Wärme umzusetzen, die in der LED für Licht sorgt. Wenn man sich jedoch überlegt, dass die Glühlampen bis zu 0,1 A brauchen, sind die LEDs mit ihren 0,02 A richtige Sparer.

Mittels Ohm'schem Gesetz ergibt sich nun der Vorwiderstand aus:

$$R_v = \frac{U_v}{I} = \frac{8,8 \text{ V}}{0,02 \text{ A}} = 440 \Omega$$

Die elektrische Leistung an den Gleichrichterdioden und an der LED braucht nicht berücksichtigt zu werden. Der verwendete Diodentyp LL4148 kann mehr als 100 mA vertragen und die LED ist ja für die 20 mA vom Hersteller ausgelegt worden. Nur der von mir bestimmte Widerstand muss kontrolliert werden. An ihm fällt folgende Leistung ab:

$$P = U \cdot I = 8,8 \text{ V} \cdot 0,02 \text{ A} = 0,176 \text{ W}$$

Wir verwenden keine bedrahteten, sondern SMD-Widerstände. Erstere vertragen typischerweise 0,25 W. Ein Blick in das Datenblatt zeigt aber, dass die SMD-Widerstände in der angestrebten Baugröße nur bis 0,125 W belastbar sind.

Nach kurzer Überlegung war die Lösung da: Die Verlustleistung wird einfach auf zwei Widerstände aufgeteilt. Wir verwenden zweimal 220 Ω anstatt einmal 440 Ω , und jeder muss nur die halbe Leistung verkraften, da nur die halbe Spannung an ihm abfällt (während der Strom gleich bleibt):

$$P = \frac{U}{2} \cdot I = \frac{8,8 \text{ V}}{2} \cdot 0,02 \text{ A} = 0,088 \text{ W}$$

Super, so geht es tadellos.

Diese Widerstände sind sogar aus der allgemeinen E6-Normreihe direkt erhältlich.

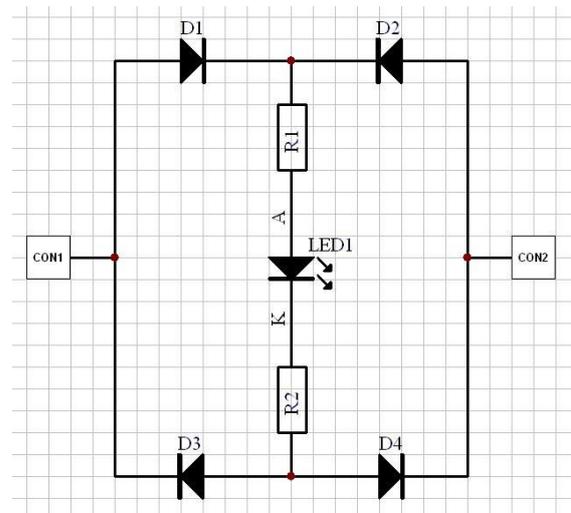


Abb. 1: Gesamtschaltung mit Graetz-Diodenbrücke und den beiden Vorwiderständen

Diese Schaltung versorgt die LED unabhängig von der außen anliegenden Spannung immer in der korrekten Richtung mit Strom. Das zeigen die folgenden Abbildungen, in denen die je nach Stromrichtung ungenutzten Dioden des Gleichrichters der Übersichtlichkeit wegen weggelassen wurden:

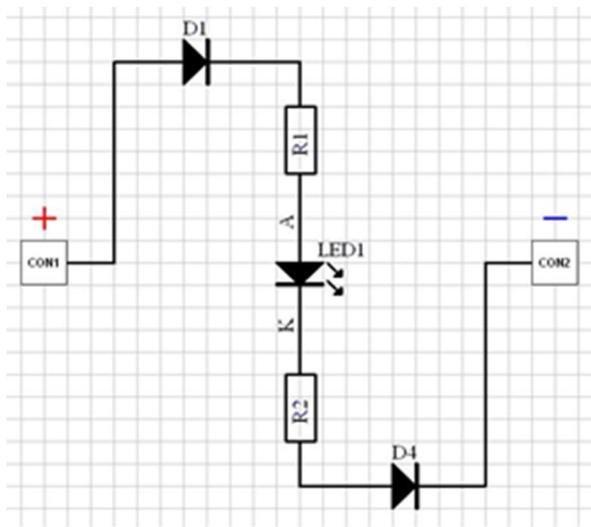


Abb. 2: Graetz-Brücke mit positivem Pol links

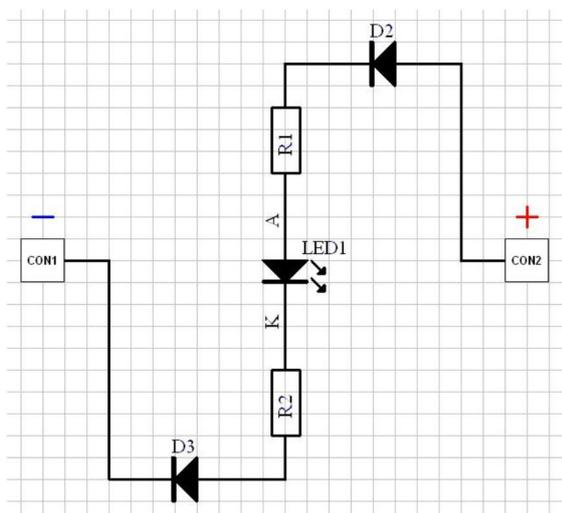


Abb. 3: Graetz-Brücke mit positiven Pol rechts

Eingabe in das PCB-Programm

Die Grundlagen sind berechnet. Die Größe und Art der Bauteile stehen auch fest. Somit kann alles in ein Layout-Programm eingegeben werden. Wir verwenden die folgenden Bauteile:

- D1 bis D4: LL4148
- R1 und R2: 220 Ohm 1206
- LED: KP-3216YC 1206 (gelb)

Ich hatte alles eingegeben, per Transferfunktion im Programm auf die PCB-Seite (*Printed Circuit Board*, zu Deutsch: Leiterplatte) übertragen und die Außenabmessungen der Platine von den ft-Leuchtkappen abgenommen.

Super! Hier habe ich das erste Mal gesehen, dass die Schaltung auf die benötigte Größe gebracht werden kann. Die Schaltung passte somit in die ft-Leuchtkappen. Es konnte also weiter gehen.

Die daraus entstandene Datei habe ich einem Platinen-Layer per E-Mail übergeben. Dieser hat dann innerhalb von wenigen Tagen die Platinen hergestellt. Damit ich den Arbeitsfortschritt beobachten konnte, bekam ich die Bilder der einzelnen Arbeitsschritte zugesendet. Diese Bilder folgen nun hier:



Abb. 4: Bohren der Durchkontaktierungen

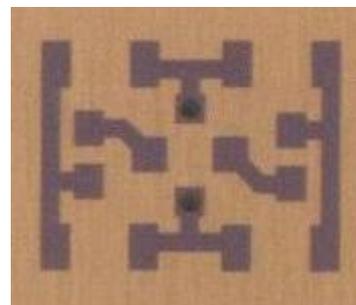


Abb. 5: Resist auftragen und belichten

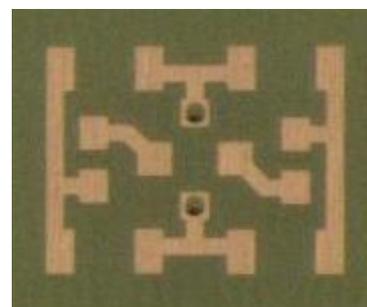


Abb. 6: Platine Ätzen und Resist entfernen

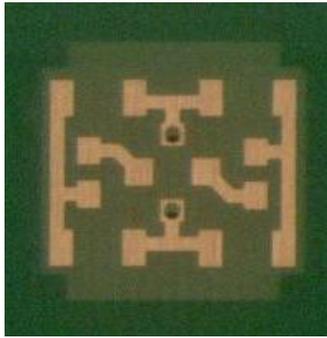


Abb. 7: Lötstopplack aufbringen und aushärten

Lötstopplack brauchte ich bei der kleinen Platine nicht. Daher ist das Auftragen in Abb. 7 nur am Rand erkennbar.

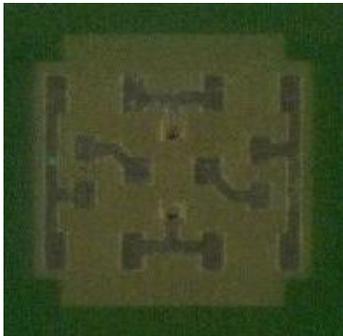


Abb. 8: Letzter Schritt: chemisch Zinn auftragen

Die Platinen sind nun gebrauchsfertig. Die Löt pads für die Bauteile sind gut zu erkennen. Die beiden Löcher für die Anschlussdrähte sind an der richtigen Position, sodass die Anschlussdrähte an genau derselben Stelle heraus kommen, wie die Glühlampen es fischertechnik vorgegeben haben. Die Außenumrandung ist gut am Rand des Lötstopplacks zu erkennen. Die Ecken halten den Platz für die vier Steckstifte der alten ft-Leuchtstein-Kappen frei.

Was ich zu diesem Zeitpunkt noch nicht wusste: Für den nach der Biegung zurück kommenden Draht brauche ich noch zwei Bohrungen. Diese fügte ich nachträglich per Standbohrmaschine ein.

ft-Produktionsstraße

Wenn man eine Platine bestellt, muss man heutzutage immer mindestens 1 dm² abnehmen. Die kleinen Platinen sind nicht wirklich groß. Somit wurden aus einer ganz viele...

Ich hielt nach zwei Wochen Lieferzeit gute 60 Stück in meinen Händen. Das war etwas viel, aber es gibt ja eBay ☺. Alle, die mir zu viel waren, wurden dort angeboten – und sie wurden mir innerhalb weniger Wochen aus den Händen gerissen. Der schöne kleine Nebeneffekt war, dass ich bei jedem Verkauf einen kleinen Gewinn machte. Gerade so groß, dass am Ende der Materialwert von allen abgedeckt war. Die LEDs, welche ich mir zurückgelegt hatte, waren somit bezahlt. Ich hatte nur meine Arbeitszeit selbst zu tragen, und das tat ich mit Freude.

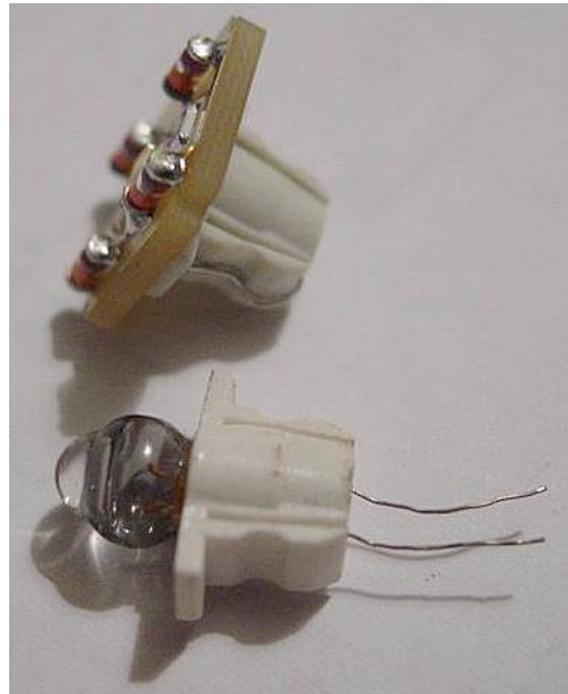


Abb. 9: Das defekte ft Lämpchen und die LED-Lösung

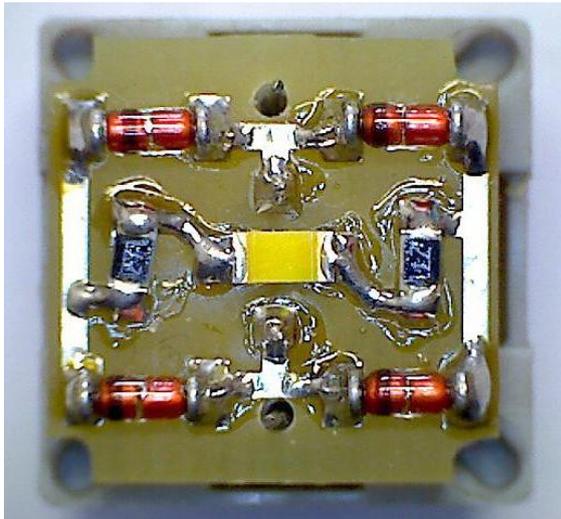


Abb. 10: Platine im aufgesteckten Zustand

Hier sieht man sehr schön die nachträglich eingebrachten Löcher für die zurückkommenden Anschlussdrähte (Abb. 10).

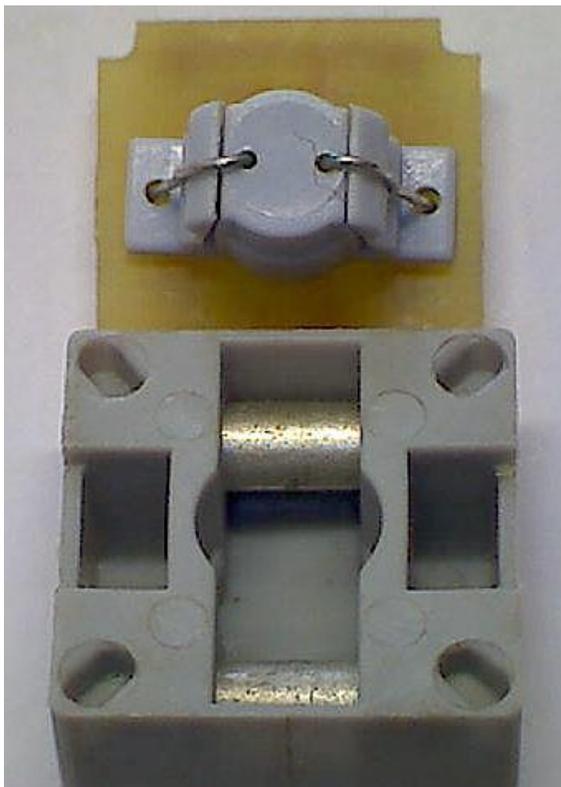


Abb. 11: Rückansicht der Platine

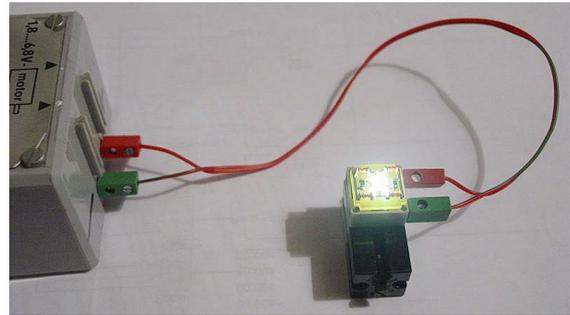


Abb. 12: Der Beweis – es funktioniert!

Produktions-Plattform

Um die vielen Lötungen einfacher zu gestalten, habe ich mir damals einen kleinen Drehtisch mit fischertechnik gebaut. Die Platine wurde aufgesteckt, dann die Anschlussdrähte. Die darunter befindlichen Löcher waren in der richtigen Länge vorbestimmt. Die Drähte wurden abgeschnitten und verlötet. Dann habe ich von innen nach außen bestückt und gelötet. Erst die LED, dann die Widerstände und zum Schluss die Gleichrichter-Dioden. Bei den Dioden ist natürlich immer die Polung wichtig.

Es hat Spaß gemacht und gut funktioniert.

Ausblick: Was geht noch...

Im Text habe ich erwähnt, dass diese von mir gemachten LEDs nur für die alte Version der Leuchtkappen funktionieren. Zum damaligen Zeitpunkt besaß ich keine Leuchtkappen mit Verriegelung – das ist heute anders. Von den Kappen habe ich bereits die Maße abgenommen. Auch hier passt die Schaltung unter die Kappen. Den Beweis hat mein PCP-Programm schon geliefert...

Mit den Platinen kann man natürlich auch etwas spielen. Beispielsweise kann man den Gleichrichter weglassen und setzt nur die Widerstände und eine zweifarbige LED ein. Nun leuchtet die LED in einer Polung in der ersten Farbe und in der anderen Polung in der zweiten. Im Wechselstrom-Betrieb wird daraus die Mischfarbe. Wenn man ROBO Pro verwendet, kann man durch Zeitversatz

(*Impulsbreitensteuerung*, [4]) sogar die Farbe verschieben.

Da LEDs sehr schnell reagieren (der Glühfaden der klassischen Lämpchen verhält sich viel träger) sind auch Stroboskop-Anwendungen denkbar. Hier ist nur die Verarbeitungsgeschwindigkeit von ROBO Pro die Bremse.

Quellen

- [1] Wikipedia: [Gleichrichter](#).
- [2] Thomas Habig: *LEDs mit Vorwiderstand*. [ft:pedia 2/2011](#), S. 14-18.
- [3] Wikipedia: [Surface-mounted device](#).
- [4] Wikipedia: [Pulsweitenmodulation](#).

Computing

Parallel Interface – durch Arduino gesteuert (3)

Jens Lemkamp

Im dritten Teil der Parallel-Interface- und Arduino-Reihe wollen wir unser erstes Modell zum Leben erwecken. Es handelt sich um einen Klassiker der MSR-Technik (Messen-Steuern-Regeln). Ich habe aus dem Ur-Computing-Kasten (30554) des Jahres 1984 das Modell ‚Antennenrotor‘ gewählt, um die Analog-Eingänge des Arduinos für eine typische Regelungsaufgabe zu verwenden, die immer wieder für unterschiedliche Zwecke auftaucht [1].

Aufgabenstellung

Eine drehbar gelagerte Antenne soll mittels eines Drehrades, welches als sogenanntes Stellglied fungiert, ferngesteuert werden. Die Antenne ist auf einem Zahn-Drehkranz montiert. In der Mitte ist ein Potentiometer verbaut, welches bei Rotation der Antenne verstellt wird und somit die Winkelposition misst. Der Drehkranz wird durch einen Minimot mit Getriebe und Schnecke angetrieben (Abb. 1).

Damit gliedert sich der Gesamtaufbau in folgende Komponenten:

1. Drehrad mit Poti (Sollwert-Eingabe)
2. Poti im Drehkranz (Istwert-Messung)
3. Motor, Getriebe, Zahnkranz (Stellglied)

Regelung

Hier haben wir es schon mit den drei wesentlichen Teilen einer einfachen Regelung zu tun. Warum der Aufbau eine echte Regelung ist und keine Steuerung, erklärt sich aus dem geschlossenen Regelkreis (Abb. 2) [2]. Für unser Antennenmodell bedeutet das:

- Geregelt wird die Regelgröße x (Drehwinkel der Antenne). x ist damit auch der sogenannte Istwert.

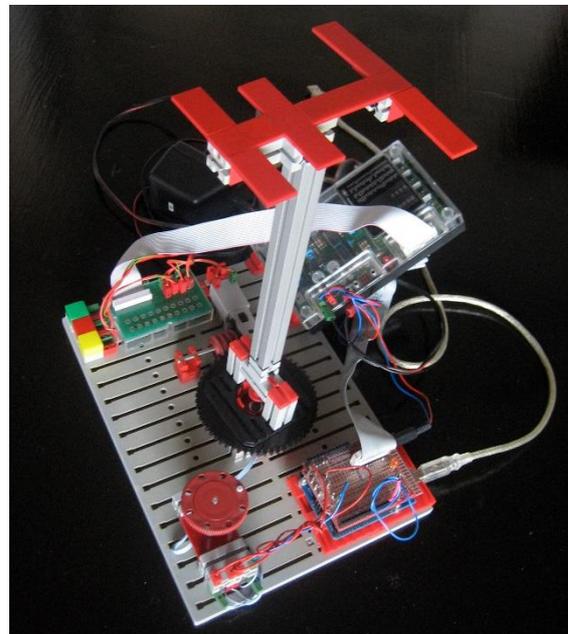


Abb. 1: Antennenrotor

- Die Führungsgröße w ist der Sollwert des Antennenwinkels, einstellbar durch das Handrad.
- e ist die Regelabweichung. Diese ist Null, wenn Soll- und Istwert gleich sind, d. h. die Antenne hat die von uns eingestellte (Soll-) Position angefahren.
- Der Regler (in unserem Modell der Arduino mit Programm) bildet aus dem Soll- und Istwert die Stellgröße y . Das ist der Wert, um den die Antenne verfahren werden muss, wenn sie nicht die Sollposition hat.

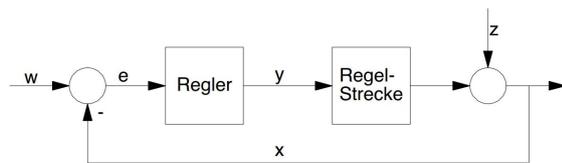


Abb. 2: Regelkreis

- Die Regelstrecke besteht aus Motor, Schnecke und Zahnrad.
- **z** ist die Störgröße. In der Realität könnte z. B. der Wind eine Kraft auf die Antenne bringen, welche zur Veränderung des Istwertes (also des Drehwinkels der Antenne) führt. Der Regler stellt dann automatisch nach.

Somit haben wir einen vollständig geschlossenen Regelkreis.

Schaltung

Ein Potentiometer (kurz: Poti) ist ein verstellbarer Widerstand mit drei Anschlüssen. Ein Schleifer wird durch die Verstellung des Drehwinkels einer Achse über eine Widerstandsbahn geführt. Die beiden äußeren Anschlüsse werden mit 0 V und +5 V verbunden (Abb. 3).

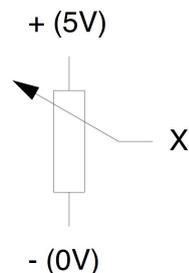


Abb. 3: Schaltzeichen Poti

Der Schleifer ist mit dem Anschluss ‚X‘ verbunden. Wenn das Poti ganz nach rechts bzw. im Schaltbild nach oben gedreht wird, ist der Schleifer direkt mit + bzw. 5 V verbunden, auf Linksanschlag bzw. unten entsprechend mit 0 V.

Bei Einstellungen zwischen 5 V und 0 V stellen sich analog zum Drehwinkel Zwischenwerte ein. Sollte z. B. der Schleifer exakt auf Mittelstellung stehen, erhalten wir 2,5 V am Anschluss ‚X‘.

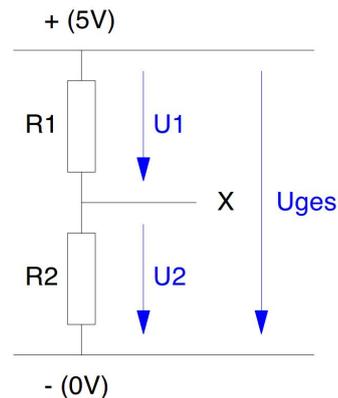


Abb. 4: Spannungsteiler

Das Poti verhält sich wie ein Spannungsteiler (Abb. 4). Widerstände werden mit dem Formelzeichen R bezeichnet, Spannungen mit U . Die Einheiten sind Ω (sprich: Ohm) für den elektrischen Widerstand und V (sprich: Volt) für die Spannung. Dabei gelten die folgenden Gesetzmäßigkeiten:

- Die beiden Einzelwiderstände addieren sich zu einem Gesamtwiderstand:

$$R_1 + R_2 = R_{ges.}$$

- Die Spannungen am Spannungsteiler verhalten sich zueinander im Verhältnis wie die Widerstände zueinander:

$$U_1 + U_2 = U_{ges.}$$

Dabei ist U_1 die Spannung über R_1 und U_2 über R_2 . $U_{ges.}$ ist die Spannung über beide Widerstände, hier also 5 V.

Wenn unser Poti einen Gesamtwiderstand von z. B. 100 k Ω hat (dieser Wert ist meist auf dem Gehäuse aufgedruckt; manchmal wird das Ω -Zeichen weggelassen und man findet nur ‚100 k‘) und das Poti auf Mittelstellung gedreht wird, teilen sich die Widerstandswerte genau auf.

Mit einem Ohm-Meter (ist in den meisten Multimetern bzw. Vielfach-Messgeräten enthalten) kann man dann also zwischen dem Mittelabgriff und jedem der beiden anderen Anschlüssen jeweils 50 k Ω (also genau die Hälfte von 100 k Ω) messen.

Achtung: Wenn man Widerstandswerte mit einem Messgerät messen möchte, muss jede

externe Spannung (also unsere 5 V aus dem Beispiel) abgeschaltet sein, sonst kann das Messgerät zerstört werden. Außerdem sollten alle Anschlüsse frei und nicht mit anderen Schaltungsteilen verbunden sein, weil man sonst fehlerhafte Messwerte erhalten kann.

Die Spannungen verhalten sich wie die Widerstandswerte, demnach stellen sich bei $U_{\text{ges.}} = 5 \text{ V}$ und Poti auf Mittelstellung folgende Werte ein:

$$R_1 = 50 \text{ k}\Omega, R_2 = 50 \text{ k}\Omega$$

$$R_1 + R_2 = R_{\text{ges.}} = 50 \text{ k}\Omega + 50 \text{ k}\Omega = 100 \text{ k}\Omega$$

$$U_1 = 2,5 \text{ V}, U_2 = 2,5 \text{ V}$$

$$U_1 + U_2 = U_{\text{ges.}} = 2,5 \text{ V} + 2,5 \text{ V} = 5 \text{ V}$$

Ein weiteres Beispiel:

Der Schleifer des Potis steht nahe bei dem 5 V- und deutlich vom 0 V-Anschluss entfernt, sagen wir bei ca. 10 %. Dann haben wir folgende Verhältnisse:

$$R_1 = 10 \text{ k}\Omega, R_2 = 90 \text{ k}\Omega$$

$$R_1 + R_2 = R_{\text{ges.}} = 10 \text{ k}\Omega + 90 \text{ k}\Omega = 100 \text{ k}\Omega$$

Und für die Spannungen gilt:

$$U_1 = 0,5 \text{ V (10\%)}, U_2 = 4,5 \text{ V (90\%)}$$

$$U_1 + U_2 = U_{\text{ges.}} = 0,5 \text{ V} + 4,5 \text{ V} = 5 \text{ V}$$

Man kann auch sagen, dass sich die Spannung am Mittelanschluss des Potis proportional zum Drehwinkel verhält. Das setzt ein ideales, lineares Potentiometer voraus. Damit können wir jetzt den Drehwinkel einer Poti-Achse mittels der Spannung messen, die sich mit dem Drehwinkel verändert.

Modell

Der Aufbau des Modells ist trivial: Ich habe es aus der Anleitung des 1984er Computing-Kastens weitestgehend original nachgebaut [1]. Wir treiben einen Drehkranz mittels Mini-Motor und Schnecke an. In der Mitte des Drehkranzes ist ein Poti montiert, welches den Istwert (Drehwinkel der Antenne) misst und an den Arduino weitergibt (Abb. 5).

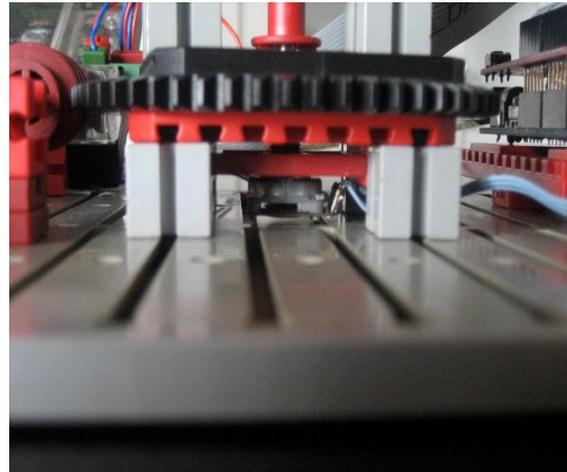


Abb. 5: Istwertgeber

Daher erklären sich auch die 5 V aus den Rechenbeispielen, denn der Arduino wird mit 5 V versorgt und die Analog-Digital-Wandler vertragen ohne zusätzliche Beschaltung maximal 5 V. Ein zweites Poti stellt als Handrad die Istwerteingabe dar.

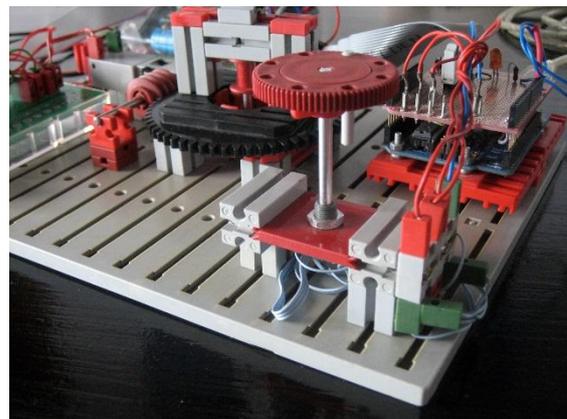


Abb. 6: Sollwertgeber

Der Analog0-Eingang des Arduinos wird mit dem Mittelabgriff des Handrades verbunden, der Mittelabgriff des Potis vom Drehkranz mit dem Analog1-Eingang (Abb. 6). Die beiden anderen Anschlüsse der Potis werden jeweils mit 0 V und 5 V verbunden, und zwar der Anschluss am Linkseinschlag mit 0 V, der Anschluss am Rechtseinschlag mit +5 V (dabei die Spannungsversorgung vom Arduino verwenden!).

Programm

Das Programm ist im Ablauf den alten Basic-Programmen aus dieser Baukastenreihe nachempfunden:

```
void loop(){
  soll = analogRead(analog0); // lese den Sollwert Analog 0
  ist = analogRead(analog1); // lese den Istwert Analog 1
  i = abs(soll - ist);
  Serial.print ("Soll: IST: DIFF: ");
  Serial.print (soll);
  Serial.print (" : ");
  Serial.print(ist);
  Serial.print (" : ");
  Serial.println(i);

  if (soll < ist) {
    outbyte = B00000001;
  }
  if (soll > ist) {
    outbyte = B00000010;
  }

  motout(); //aktuelles Motordatenbyte ausgeben

  if (i < 1) {
    outbyte = B00100000;
    motout();
  }

  if (i < 30) {
    delay(i*3); //milis Pause
    outbyte = B00000000;
    motout(); //aktuelles Motordatenbyte ausgeben
  }
}
```

Abb. 7: Hauptteil des Sketches; void loop()

Die beiden Variablen ‚soll‘ und ‚ist‘ werden mit den Messwerten aus dem Analog/Digitalwandlern des Arduino versorgt. Aus diesen beiden Werten wird die Regelabweichung e gebildet und in der Variablen i hinterlegt. Das ist der Absolutwert der Differenz von Soll- und Istwert.

Diese Werte (Soll, Ist, Differenz) werden über die serielle Schnittstelle an einen evtl. angeschlossenen Computer gesendet und können dort mit einem Terminal-Programm angezeigt werden.

Jetzt wird ausgewertet und der Motor ggf. gefahren:

- Wenn der Sollwert *kleiner* als der Istwert ist, muss der Motor die Antenne *links* herum drehen.
- Ist der Sollwert *größer* als der Istwert, muss die Antenne entsprechend nach *rechts* rotiert werden.

- Das entsprechende Bit wird im Ausgabebyte gesetzt und der Motor durch die Funktion `motout()` gestartet. `motout()` ist eine Funktion, welche die die Bits des `outbyte` zum Interface überträgt (siehe Gesamtprogramm) und damit den Motor startet.
- Wenn $i < 1$ ist die Sollposition erreicht (Istwert = Sollwert), und es werden eine Lampe ein- und der Motor ausgeschaltet.

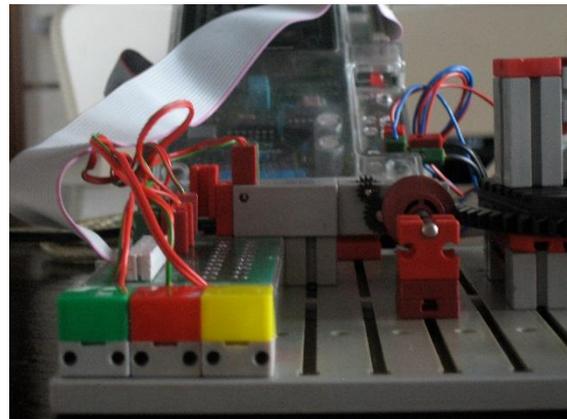


Abb. 8: Antrieb

Jetzt kommt es zu einer Besonderheit, ohne die das Programm bzw. der Regler nicht ordentlich arbeiten würde. Dieser Effekt ist auch in der Programmieranleitung des Ur-Computing-Kastens beschrieben, und die Lösung dieses typischen Regel-Problems habe ich – mit abgewandelten Parametern – übernommen.

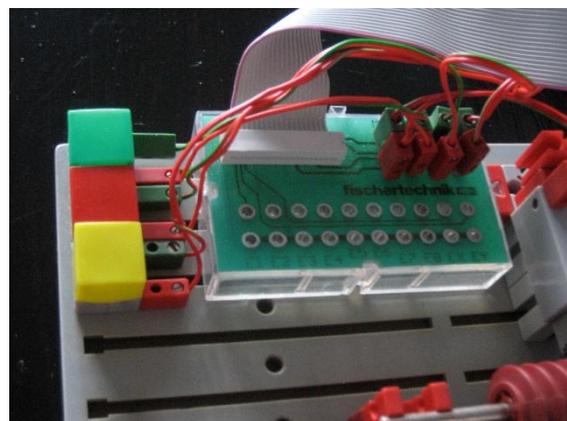


Abb. 9: Anschlussfeld

Bei Erreichen der Soll-Position genügt es nicht, den Motor einfach nur zu stoppen:

durch die Masse des Antennenmastes, des Drehkranzes und des Getriebes läuft die Antenne etwas nach. Das führt dann dazu, dass beim nächsten Durchlauf des Loops der Regler feststellt, dass der Sollwert überschritten wurde und nachregelt, d. h. der Motor fährt zurück, wird beim Ziel (Sollwert = Istwert) wieder gestoppt, und läuft wieder etwas nach, diesmal aber in der anderen Richtung.

Damit zittert die Antenne um ihre eigentlich Soll-Position bzw. fährt je nach Masse hin und her: Der Regler „schwingt“. Eigentlich ist aber gewünscht, dass die Antenne bei Erreichen der Soll-Position stoppt und dort bleibt, bis ein neuer Sollwert kommt oder aber eine Störgröße die Antenne aus der Position bringt und der Regler deshalb eingreifen muss.

Daher erfolgt im letzten Teil der Hauptschleife eine Abfrage, wie weit der Istwert vom Sollwert entfernt ist. Ist die Differenz kleiner 30, wird eine kleine Pause eingelegt und der Motor gestoppt. Die Länge dieser Pause bis zum Abschalten ist abhängig von der Differenz. Je kleiner die Differenz, desto eher wird der Motor abgeschaltet.

Danach geht der Regelkreis weiter im Durchlauf, d. h. wir fragen erneut die Soll- und Istwerte ab und durchlaufen den Regler wie gehabt; der Motor wird eingeschaltet, falls noch eine Differenz zwischen Soll und Istwert besteht.

Durch die Pause (Differenz $i-3$) in Millisekunden wird der Motor schon vor Erreichen des Sollwertes abgebremst, weil wir ihn nicht durchlaufen lassen, sondern abschalten. Damit wird der Motor langsamer, je näher wir dem Ziel kommen, und der Regler schwingt nicht über.

Die Länge der Pause und der Beginn der Bremsphase sollte der Mechanik des Modells angepasst werden, da sie abhängig von der Masse der Konstruktion, dem Getriebe, dem Motor und der genauen Spannung ist, mit der unsere Anlage versorgt wird.

```
int ledpin = 13; // ARDUINO TESTLED
int clock = 5;
int dataout = 4;
int loadout = 2;
int datain = 6;
int loadin = 3;
int outbyte = 0; // M4 M3 M2 M1 Motoroutputbyte auf 0 setzen = alle aus!
int analog0 = 0; // potentiometer wiper (middle terminal) connected to analog pin 0
int analog1 = 1; // potentiometer wiper (middle terminal) connected to analog pin 1
int i = 0; // Hilfsvariable
int soll = 0;
int ist = 0;

// int inbyte = 0; // vordefiniert alle inputs sind 0
int e[] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0}; // E1 = e[1] nicht: e[0]

void setup() {
  Serial.begin(9600);
  pinMode(ledpin, OUTPUT);
  pinMode(clock, OUTPUT);
  pinMode(dataout, OUTPUT);
  pinMode(loadout, OUTPUT);
  pinMode(datain, INPUT);
  pinMode(loadin, OUTPUT);

  digitalWrite(clock, LOW);
  digitalWrite(dataout, LOW);
  digitalWrite(loadout, LOW);
  digitalWrite(datain, HIGH);
  digitalWrite(loadin, LOW);
  shiftOut(dataout, clock, MSBFIRST, 0); // alle Ausgänge auf 0
}

void loop() {
  soll = analogRead(analog0); // lese den Sollwert Analog 0
  ist = analogRead(analog1); // lese den Istwert Analog 1
  i = abs(soll - ist);
  Serial.print("Soll: IST: DIFF: ");
  Serial.print(soll);
  Serial.print(" ");
  Serial.print(ist);
  Serial.print(" ");
  Serial.println(i);

  if (soll < ist) {
    outbyte = 00000001;
  }
  if (soll > ist) {
    outbyte = 00000010;
  }

  motout(); // aktuelles Motordatenbyte ausgeben

  if (i < 1) {
    outbyte = 00100000;
    motout();
  }

  if (i < 30) {
    delay(i*3); // millis Pause
    outbyte = 00000000;
    motout(); // aktuelles Motordatenbyte ausgeben
  }

  void motout() { // Motordatenbyte an Interface senden
    digitalWrite(loadout, LOW);
    shiftOut(dataout, clock, MSBFIRST, outbyte);
    digitalWrite(loadout, HIGH); // einschreiben / übertragen an OutputStufe im Interfa
    digitalWrite(loadout, LOW);
  }

  void getinputs() { // Input Status in e[x] Variablen legen
    digitalWrite(loadin, HIGH);
    digitalWrite(clock, LOW);
    digitalWrite(clock, HIGH);
    digitalWrite(loadin, LOW); // Registereingang geholt
    for (int i = 8; i > 0; i--) {
      digitalWrite(clock, HIGH); // Eingang holen
      e[i] = digitalRead(datain); // Bit lesen & speichern
      digitalWrite(clock, LOW); // 1. Zyklus bzw. 1. Bit ist übertragen (E8)
    }
  }
}
```

Abb. 10: Das Regelungsprogramm

Quellen

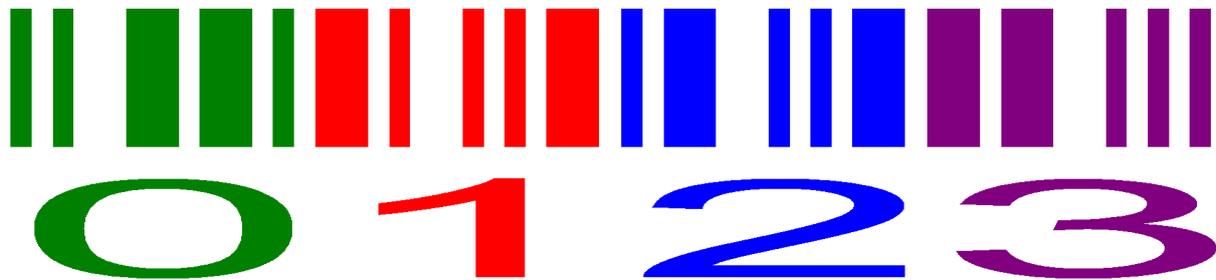
- [1] fischertechnik: [Programmieranleitung](#). fischertechnik Computing, fischerwerke 1984.
- [2] Gerhard Bader: [Grundlagen der digitalen Regelung](#). In: fischertechnik und Computer, CHIP Special 1987, S. 19-25.

Computing

Strichcode-Leser am Robo TX Controller (1): Automatisiert mit RoboPro

Andreas Gail

Strichcodes oder auch Barcodes begegnen uns täglich bei allem, was wir im Supermarkt oder sonstwo kaufen. Der vorliegende Beitrag zeigt, wie ein Strichcode-Leser mit Standard-fischertechnik-Bauteilen aufgebaut werden kann. Weiterhin werden zwei unterschiedliche Automatisierungslösungen vorgestellt: in Teil 1 unter Anwendung der RoboPro Software, in Teil 2 mithilfe von Microsoft Visual Basic 2010.



Der Testaufbau

Der Testaufbau des Strichcode-Lesers (Abb. 1) erfolgt ausschließlich mit Standardteilen. Die [Gliederkette zur Kabelführung](#) ist beim fischertechnik-Kooperationspartner Knobloch GmbH erhältlich.

Zum Start des Lesevorgangs wird der Taster {1} (TX I8) betätigt. Die Transporteinheit {2} (TX M1) setzt sich daraufhin nach links in Bewegung, läuft unter der Leseinheit {5} (TX I1-I2) hindurch und bleibt schließlich links stehen, wenn der Taster {4} (TX I4) von der Transporteinheit {2} ausgehend betätigt wird. Auf dieselbe Art und Weise kann auch ein Lesevorgang von links nach rechts durchgeführt werden. Zum Start ist erneut Taster {1} zu betätigen. Die Transporteinheit {2} bleibt erneut stehen, wenn rechts der Taster {3} (TX I3) betätigt wird. Abb. 5 zeigt den Anschlussplan.

Die Leseinheit {5} ist zum Abhalten von Störlicht mit Platten verblendet. Der eigentliche Lesesensor ist der IR-Spur-Sensor von fischertechnik ([128598](#)). Hierbei handelt es sich vermutlich um ein Bauteil ähnlich dem SHARP IS471F (Conrad Electronic [185094](#)).

Code 39

Es gibt verschiedene standardisierte Strichcodes. Die folgenden Ausführungen beziehen sich auf den Typ *Code 39*. Hiermit sind die Ziffern 0-9 und die Buchstaben A-Z darstellbar. Zur Vereinfachung wird die Decodierung exemplarisch nur für Zahlencodes dargestellt; eine Buchstabenerkennung könnte analog ergänzt werden.

Jeder Zahlenstreifencode besteht aus weißen und schwarzen (zur Verdeutlichung oben farbigen) Streifen. Zwischen den Zahlenstreifencodes ist immer ein dünner

weißer Streifen. Jeder eigentliche Zahlenstreifencode beginnt und endet somit mit einer farbigen Streifenkante. Beispielsweise hat der Zahlenstreifencode für die Ziffer 1 folgende Streifen (in diesem Beispiel von rechts nach links gelesen):

- 1 Streifen rot, breit
- 1 Streifen weiß, schmal
- 1 Streifen rot, schmal
- 1 Streifen weiß, schmal
- 1 Streifen rot, schmal
- 1 Streifen weiß, breit
- 1 Streifen rot, schmal
- 1 Streifen weiß, schmal
- 1 Streifen rot, breit

Zusammen sind das immer neun Streifen. Um die Streifenbreite einzeln erkennen zu können, wechseln sich die Streifenfarben ab: rot, weiß, rot, ... Weiterhin gibt es immer drei breite und sechs schmale Streifen. Für die Verarbeitung innerhalb der Zahlenstreifencodes ist es nicht erheblich, ob die Streifen farbige oder weiß sind, vielmehr ist nur die Breite interessant. Ein schmaler Streifen wird durch 1 und ein breiter Streifen (weil doppelt so breit) als 2 ausgedrückt. So ergibt sich für die Ziffer 1 folgende Zahlenreihe:

2
1
1
1
1
1
2
1
1
2

Summiert man die Zahlenreihe, muss man als Prüfsumme immer 12 erhalten, sonst hat es einen Lesefehler gegeben.

Ein Zahlenstreifencode kann von rechts oder von links gelesen werden; es ergibt sich jeweils eine Zahlenreihe in einander entgegengesetzter Richtung.

Test der Leseeinheit

Wie bereits auf Abb. 1 erkennbar ist die verwendete Leseeinheit für Balkenbreiten wie auf den erwähnten Supermarktprodukten nicht geeignet. Die Mindestbreite eines schmalen Balkens muss beim gewählten Aufbau mindestens 3 mm betragen. Zum Test der Lesegenauigkeit kann ein DIN A4-Blatt erzeugt werden, mit Linienbreiten 0...10 pt, wie nachfolgend gezeigt (Abb. 2).

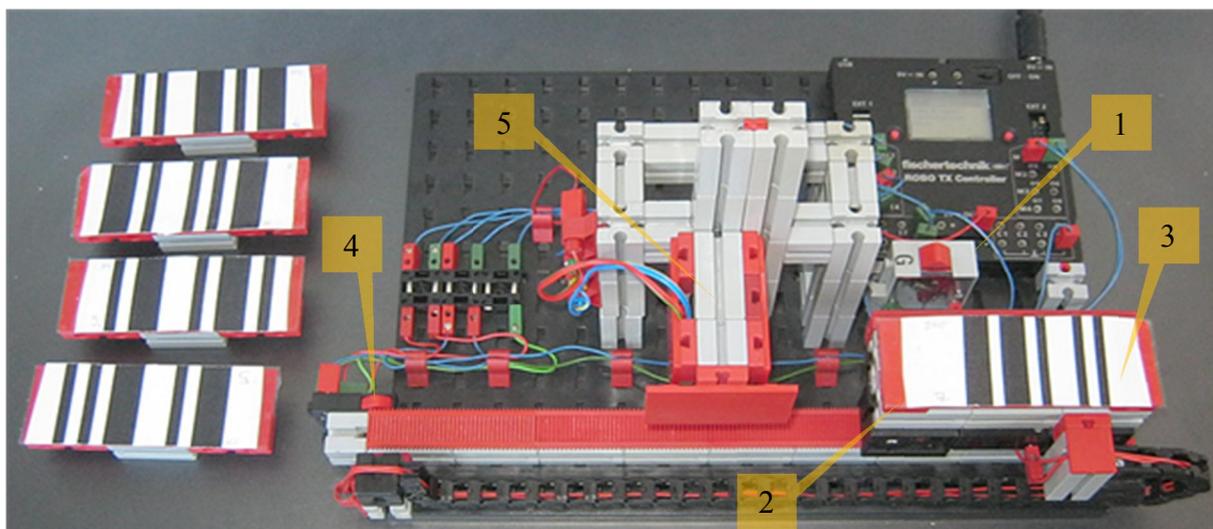


Abb. 1: Strichcode-Leser mit fünf Testcodes zum Austausch

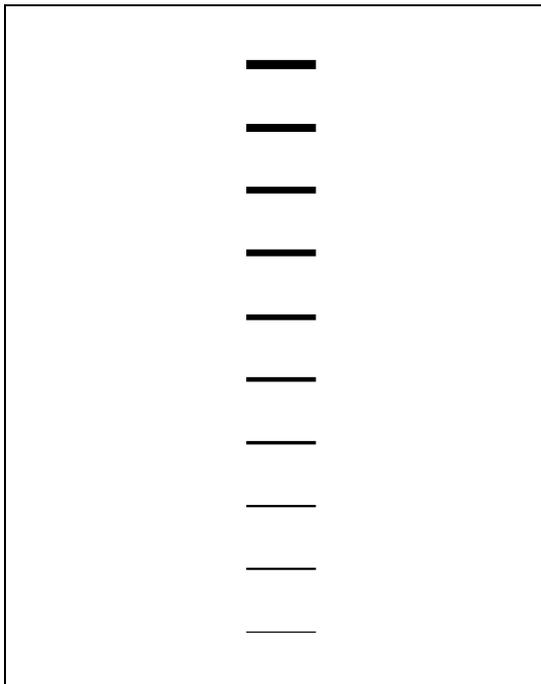


Abb. 2: Testseite (A4) zum Prüfen der Lesegenauigkeit des Lesesensors

Neben der erforderlichen Mindeststreifenbreite ist auch der Abstand des Lesesensors zum Strichcode von entscheidender Bedeutung. Im vorliegenden Fall liegt das Optimum bei 8 mm Abstand. Insgesamt ergab sich als kleinste reproduzierbare Streifenbreite folgender Strichcode:

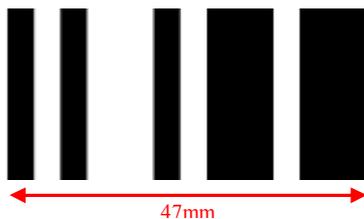


Abb. 3: Kleinster reproduzierbar lesbarer Strichcode, breite im Original 47mm

RoboPro-Konfiguration

Kurz vorweg bemerkt: Die Automatisierung des Projekts mithilfe von RoboPro ist komplex, aber machbar.

Hauptprogramm (siehe Abb. 6): Beim Lesevorgang wird der Zahlenstreifencode unter dem Lesekopf vorbeigeführt. Hierbei wird mit einer für den Controller maximalen Geschwindigkeit (innerhalb von ca. 1 ms) gemessen, wie lange hintereinander

dieselbe Streifenfarbe gelesen wird. Im Idealfall werden im 1 ms-Abstand für ein schwarzes Messergebnis eine 0 und für ein weißes Messergebnis eine 1 in eine Tabelle geschrieben (*ListeRAW*). Maximal 32766 Messungen können so erfasst werden. So werden die Rohdaten erfasst. Im Hauptprogramm wird immer erst mit der Datenerfassung begonnen, wenn weiß gemessen wurde. Schwarz ist der Dauerzustand, in dem der Sensor ‚ins Leere‘ blickt. Nach der Rohdatenerfassung beginnt die Auswertung in den Unterprogrammen 1-4; diese werden hintereinander aufgerufen. Die Funktionen in den Unterprogrammen wurden aufgeteilt, da die RoboPro-Software im Hauptprogramm bezüglich der Konfigurationsfläche sehr limitiert ist.

Unterprogramm Auswertung1: Dieses hat die Aufgabe, die zuvor erzeugte *ListeRAW* auszuwerten und das Auswertungsergebnis in *ListeAW1* zu speichern. In *ListeAW1* wird für das Auslesen des Zahlenstreifencodes der Ziffer 1 folgende Zahlenfolge ermittelt:

1
643
473
142
190
230
157
511
142
190
466
513
673

Die Zahlen bedeuten, dass 1 ms schwarz gesehen wurde, 643 ms weiß, 473 ms schwarz, 142 ms weiß, usw. Nur die rot markierten Zahlen sind dabei relevante Angaben. Die 513 ms am Ende bedeuten wieder einen weißen Streifen am Ende, und 673 ms blickt die Sonde nach der Messung ins Leere, bis die festgelegte Messdauer von 5 s abgelaufen ist.

Unterprogramm Auswertung2: Dieses berechnet die Summe der roten Zahlen und legt das Ergebnis in *ListSum* ab. Um nur die roten Zahlen auszuwerten, werden nur die Index-Nummern 2-10 der Tabelle *ListeAW1* verwendet. Es ergibt sich $ListSum=2501$.

Unterprogramm Auswertung3: Dieses berechnet zunächst aus *ListSum* die Einzelstreifenbreite, analog der Prüfsumme 12 wird $ListDiv = ListSum / 12$ berechnet. Es ergibt sich $ListDiv = 278$. Nachfolgend wird jeder relevante Wert der Tabelle *ListeAW1* durch 278 geteilt und in der Tabelle *ListeAW2FP* als Dezimalzahl gespeichert. Folgende neun Zahlen befinden sich anschließend in der Tabelle *ListeAW2FP*:

- 1,70
- 0,51
- 0,68
- 0,83
- 0,56
- 1,84
- 0,51
- 0,68
- 1,68

Unterprogramm Auswertung4: Dieses arbeitet iterativ und nutzt seinerseits die Unterprogramme *Auswertung5* und *Auswertung6*. Alle Einträge in Tabelle *ListeAW2FP* werden kaufmännisch gerundet und unter vorheriger Verwendung eines Korrekturfaktors *KorrFaktor* in Tabelle *ListeAW2* als ganze Zahl (natürliche Zahl ohne Kommastellen) abgelegt. Bei der ersten Berechnung ist $KorrFaktor = 1$, d. h. es findet keine Korrektur statt. Folgende Zahlen befinden sich dann in *ListeAW2*:

- 2
- 1
- 1
- 1
- 1
- 2
- 2
- 1
- 1
- 2

Diese Zahlenfolge entspricht dem Zahlenstreifencode für die Ziffer 1. Hierbei haben wir Glück, da das richtige Ergebnis nur sehr knapp erzielt wurde: 0,51 ergeben kaufmännisch gerundet 1; wäre eine etwas geringere Spaltenbreite an dieser Stelle gemessen worden, hätte sich ein Lesefehler ergeben.

Die zuletzt erhaltene Zahlenfolge (in *ListeAW2*) ist nur die Beschreibung von Streifen und nicht das eigentlich gewünschte Ergebnis in Form einer einzigen Ziffer. Diese Umwandlung erfolgt in den Unterprogrammen *Auswertung5* und *Auswertung6*. Zurück in Unterprogramm *Auswertung4* wird geprüft, ob in der Variablen *Ergebnis* ein akzeptierter Wert erhalten wurde. Wenn nein, dann steht in der Variablen *Ergebnis* 999. Im Fehlerfall kommen die Korrekturfaktoren in Tabelle *KorrWerte* zum Einsatz. In der Tabelle sind etwa 100 solcher Korrekturfaktoren enthalten (0,999; 1,001; 0,998; 1,002; 0,997; 1,003; ...) mit denen das Unterprogramm *Auswertung4* versucht, doch noch ein stimmiges Ergebnis zu erzielen. Nach Abschluss von Unterprogramm *Auswertung4* wird das Endergebnis ausgegeben.

In den Unterprogrammen *Auswertung5* und *Auswertung6* erfolgt aus den gefundenen Zahlenfolgen die Umrechnung in die gesuchte einzelne Ziffer, in unserem Beispiel eine 1. Im Unterprogramm *Auswertung5* wird aus der Zahlenfolge unter Nutzung des binären Zahlensystems gemäß folgendem Muster eine einzelne Zahl errechnet:

Ziffer	Streifen	Prüfsumme	Streifen binär	Streifen dezimal
0	1 1 1 2 2 1 2 1 1	12	0 0 0 1 1 0 1 0 0	88
1	2 1 1 2 1 1 1 1 2	12	1 0 0 1 0 0 0 0 1	265
2	1 1 2 2 1 1 1 1 2	12	0 0 1 1 0 0 0 0 1	268
3	2 1 2 2 1 1 1 1 1	12	1 0 1 1 0 0 0 0 0	13
4	1 1 1 2 2 1 1 1 2	12	0 0 0 1 1 0 0 0 1	280
5	2 1 1 2 2 1 1 1 1	12	1 0 0 1 1 0 0 0 0	25
6	1 1 2 2 2 1 1 1 1	12	0 0 1 1 1 0 0 0 0	28
7	1 1 1 2 1 1 2 1 2	12	0 0 0 1 0 0 1 0 1	328
8	2 1 1 2 1 1 2 1 1	12	1 0 0 1 0 0 1 0 0	73
9	1 1 2 2 1 1 2 1 1	12	0 0 1 1 0 0 1 0 0	76

Tab. 1: Codierung von Streifenmustern, gültig für vorwärts gelesene Streifen

Ziffer	Streifen	Prüfsumme	Streifen binär	Streifen dezimal
0	1 1 2 1 2 2 1 1 1	12	0 0 1 0 1 1 1 0 0 0	52
1	2 1 1 1 1 2 1 1 2	12	1 0 0 0 0 1 0 0 1	289
2	2 1 1 1 1 2 2 1 1	12	1 0 0 0 0 1 1 0 0	97
3	1 1 1 1 1 2 2 1 2	12	0 0 0 0 0 1 1 0 1	352
4	2 1 1 1 2 2 1 1 1	12	1 0 0 0 1 1 0 0 0	49
5	1 1 1 1 2 2 1 1 2	12	0 0 0 0 1 1 0 0 1	304
6	1 1 1 1 2 2 2 1 1	12	0 0 0 0 1 1 1 0 0	112
7	2 1 2 1 2 1 1 1 1	12	1 0 1 0 0 1 0 0 0	37
8	1 1 2 1 1 2 1 1 2	12	0 0 1 0 0 1 0 0 1	292
9	1 1 2 1 1 2 2 1 1	12	0 0 1 0 0 1 1 0 0	100

Tab. 2: Codierung von Streifenmustern, gültig für rückwärts gelesene Streifen

Unsere gesuchte Ziffer 1 hat also als zugehörige Zahl die 265 oder 289. Das Berechnungsergebnis aus Unterprogramm *Auswertung5* wird in der Variablen *ErgebDec* = 289 abgelegt.

Im Unterprogramm *Auswertung6* wird versucht, ob aus *ErgebDec* = 289 die gesuchte Ziffer ermittelt werden kann. In unserem Fall ist das möglich und es wird *Ergebnis* = 1 ausgegeben.

RoboPro-Visualisierung

Wie in Abb. 4 ersichtlich, sind die Möglichkeiten zur Visualisierung eher als bescheiden anzusehen. Die bei Windows übliche Standardfunktionalität des *drag-and-drop* von und nach RoboPro wird nicht unterstützt. So ist es z. B. nicht möglich, an anderen Stellen erstellte Bitmap-Grafiken auf die Oberfläche von RoboPro zu kopieren. Weiterhin sind die verfügbaren

Bedien- und Anzeigeelemente in RoboPro auf ein Mindestmaß beschränkt.

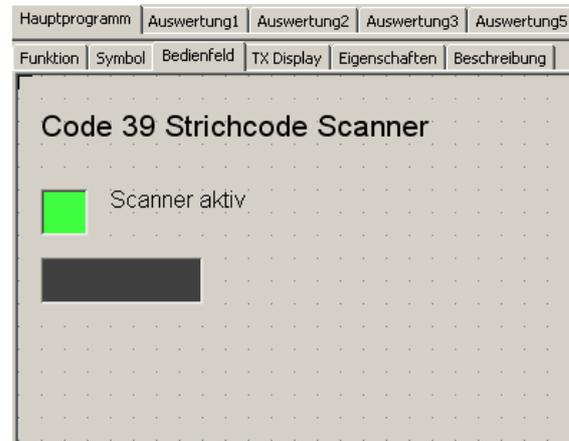


Abb. 4: Ausschnitt der Bedienfeld-Oberfläche mit RoboPro zur Visualisierung der Automatisierungsvorgänge

Ganz anders sieht das aus, wenn man die gestalterischen Möglichkeiten von Microsoft Visual Basic oder Ähnlichem betrachtet. In Teil 2 dieser Beitragsreihe in der nächsten Ausgabe der ft:pedia wird hierzu ein Beispiel gegeben.

Download

Für den interessierten Leser steht im Downloadbereich der ft-Community die komplette [RoboPro-Konfiguration](#) zur Verfügung.

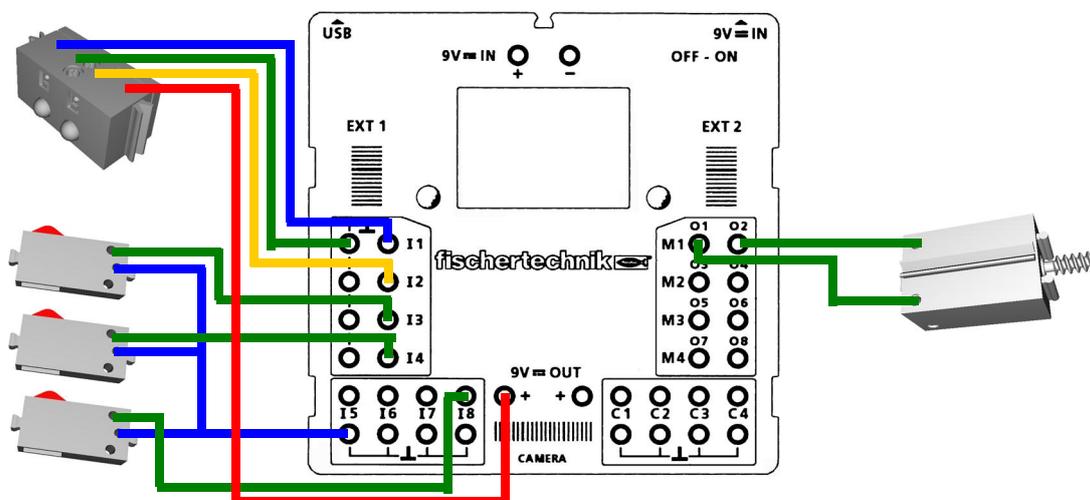


Abb. 5: Anschlussbelegung Testaufbau

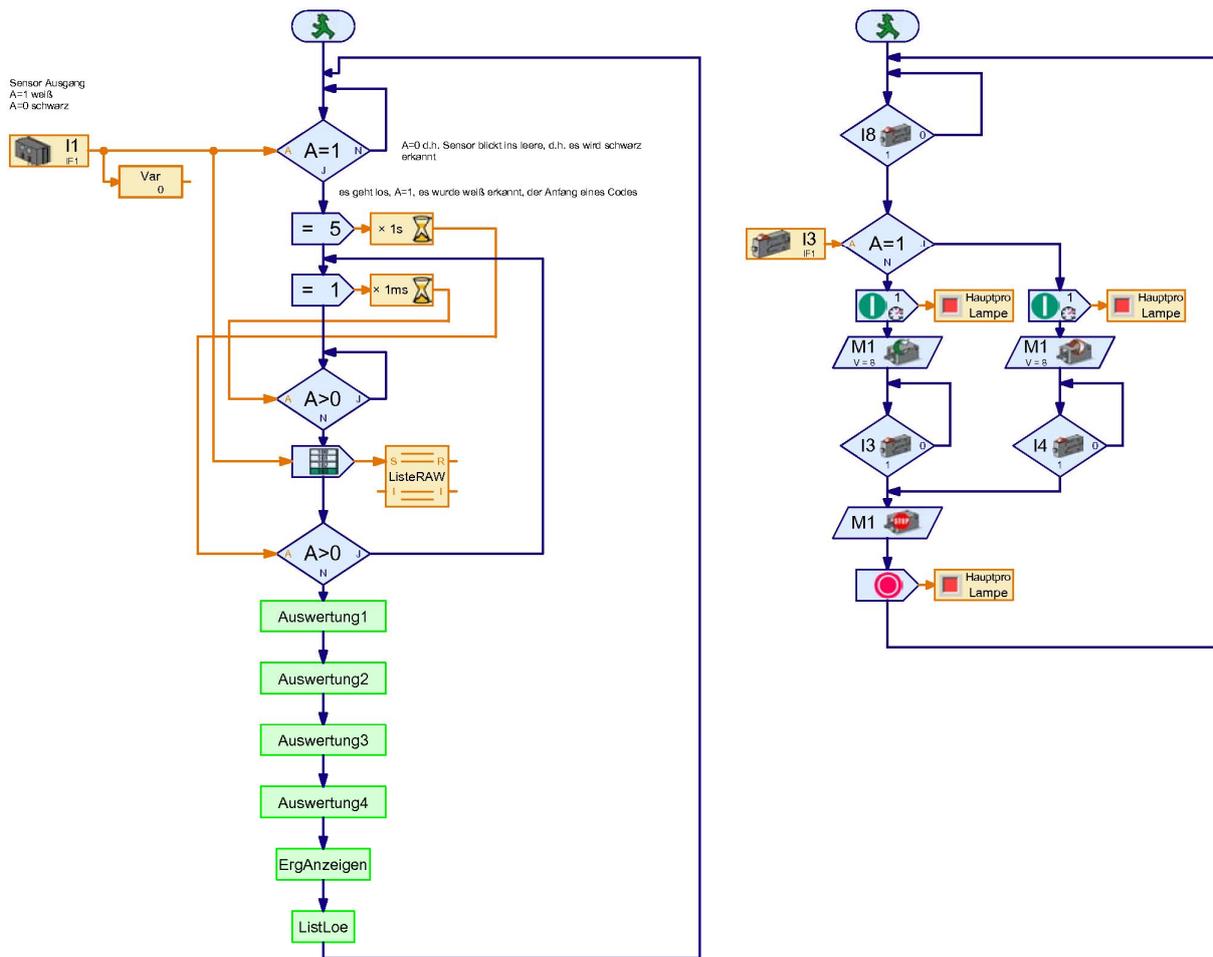


Abb. 6: Strichcode-Leser RoboPro Konfiguration, Hauptprogramm
 (links: Strichcode lesen und auswerten, rechts: Steuerung der Transporteinheit)

Robotik

Navigation

Dirk Fox

Mit einem GPS- [1] und einem Kompass-Sensor [2] haben wir alles, was wir für die automatische Steuerung eines autonomen fischertechnik-Roboters im Freien benötigen. In diesem Beitrag zeigen wir, wie sich aus den Sensor-Daten die für die Steuerung des Roboters erforderlichen Richtungsinformationen berechnen lassen.

Hintergrund

Autonome Roboter, die sich selbstständig im Gelände bewegen, gewinnen ständig an Bedeutung. Die Anwendungen reichen von Fahrerassistenzsystemen bis zu autonomen Robotern in der Landwirtschaft – sehr eindrucksvoll beim jährlichen „[Field Robot Event](#)“ zu bewundern. Eine zentrale Rolle spielt dabei die möglichst exakte Positionsbestimmung – und die war dank des *Global Positioning Systems* (GPS) noch nie so genau und einfach möglich wie heute [1].

GPS-Positionsdaten

Die von einem GPS-Empfänger ermittelten Positionsdaten bestehen aus dem Breiten- und dem Längengrad. Beide werden in der Regel – auch von unserem GPS-Sensor [1] – als Dezimalzahlen angegeben. Die früher übliche Darstellung in Grad, Minuten und Sekunden (sexagesimal) lässt sich leicht aus dem Dezimalwert gewinnen: Eine Minute entspricht einem 60stel Grad, eine Sekunde einer 60stel Minute.

In [Google Maps](#) können wir uns die Koordinaten z. B. des Eiffelturms in Paris in beiden Darstellungen anzeigen lassen (Breitengrad; Längengrad):

- Dezimal: 48,858562°; 2,294481°
- Sexagesimal: 48°51'30,8" N;
2°17'40,1" O

Die Buchstaben „N“ (für nördliche Breite) und „O“ (für östliche Länge, auch „E“ für „East“) werden benötigt, da der Breitengrad ausgehend vom Äquator (0°) nach Norden und Süden, der Längengrad ab dem so genannten „Nullmeridian“ (dem Längengrad durch die Londoner Sternwarte Greenwich) nach Westen und Osten gemessen wird. In der Dezimaldarstellung werden stattdessen Vorzeichen verwendet (Süd und West = -). Die Übereinstimmung beider Darstellungen könnt ihr mit dem Taschenrechner leicht überprüfen.

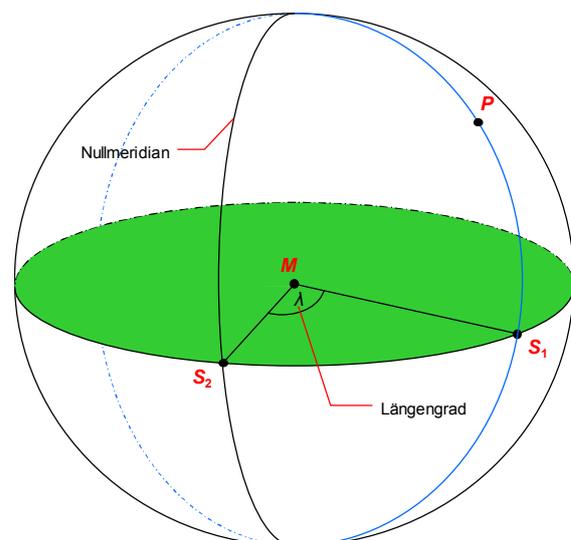


Abb. 1: Längengrad λ

Der Längengrad gibt die Größe des Mittelpunktswinkels λ des Kreissektors auf dem Schnittkreis (grüne Fläche in Abb. 1) durch die Erdkugel am Äquator an. Der entsteht

wie folgt: Man stelle sich eine Kreislinie (blaue Linie in Abb. 1) um die Erdkugel durch die aktuelle Position P und beide Pole vor. Vom Schnittpunkt S_1 dieser Kreislinie mit dem Äquator ziehe man eine Linie (Lot) zum Mittelpunkt M der Erdkugel, dann von dort eine zweite Linie zum Schnittpunkt S_2 des Äquators mit dem Null-Meridian. Die beiden Linien bilden die Schenkel des Längengrads λ , der einen Wert zwischen -180° und 180° (statt der Vorzeichen auch mit westlicher bzw. östlicher Länge bezeichnet) annimmt. Die Werte -180° , 0° und 180° liegen dabei auf dem Nullmeridian.

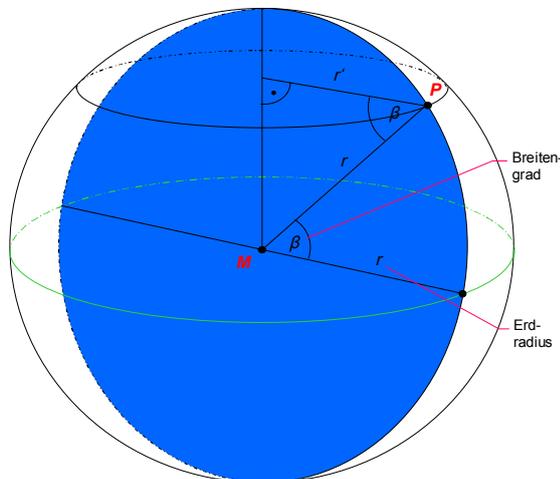


Abb. 2: Breitengrad β

Der Breitengrad β gibt den Winkel an, in dem das Lot der aktuellen Position P die Äquatorebene in M schneidet (Abb. 2). Der Breitengrad nimmt Werte zwischen -90° (Südpol) und 90° (Nordpol) an (statt der Vorzeichen auch mit südlicher bzw. nördlicher Länge bezeichnet); am Äquator ist $\beta = 0^\circ$.

Tatsächlich liegt der Schnittpunkt nicht genau im Mittelpunkt M des Äquatorkreises, da die Erde ein Ellipsoid und keine Kugel ist – dazu später mehr.

Genauigkeit der GPS-Daten

Die Positionsbestimmung ist vor allem auf hoher See von großer Bedeutung. Seit der Entwicklung von genauen Uhren Mitte des 18. Jahrhunderts durch [John Harrison](#)

(1693-1776) reduzierte sich die Ungenauigkeit von Messungen mit Sextanten immerhin auf wenige Seemeilen (1.852 km) [3]. Mit der Satellitennavigation lassen sich – auch bei schlechten Wetterbedingungen – deutlich bessere Ergebnisse erzielen. So begrenzten die nicht-militärischen GPS-Signale die Ungenauigkeit auf etwa die Fläche eines Fußballplatzes. Seit der Abschaltung der künstlichen Verzerrung am 01.05.2000 liegt die mit GPS-Signalen theoretisch erreichbare Genauigkeit unter 10 m.

In der Praxis hängt die mit einem GPS-Empfänger tatsächlich erreichbare Genauigkeit der Positionsbestimmung allerdings von einigen Randbedingungen ab:

- Abschattung (kein direkter Sichtkontakt zu Satelliten, daher Dämpfung der Signale)
- Reflexion (an Gebäuden, vor allem in „Häuserschluchten“)
- Zahl der Satelliten (mindestens vier erforderlich, im Freien meist sieben bis 12 sichtbar)
- Position der Satelliten (Einfallswinkel der Signale)

Außerdem werden die Signale in der Ionosphäre gedämpft und gebeugt; hinzu können Störungen durch die Überlagerung von Fremdsignalen kommen. Daher filtern gängige GPS-Sensoren unplausible Satellitendaten aus und bilden Mittelwerte über mehrere Positionsbestimmungen, um Fehler und Störungen abzuschwächen.

Unter guten Messbedingungen – außerhalb geschlossener Ortschaften und bei wolkenlosem Himmel im Freien, damit die Signale von möglichst vielen, weit auseinander liegenden Satelliten empfangen werden können – erreicht ein GPS-Empfänger etwa die in Tabelle 1 angegebene Genauigkeit.

Anteil	β/λ	Höhe
99 %	+/- 5 m	+/- 8 m
95 %	+/- 4 m	+/- 6 m
50 %	+/- 2 m	+/- 2 m

Tab. 1: Erreichbare GPS-Genauigkeit unter günstigen Empfangsbedingungen [4]

Die Werte decken sich mit meinen Erfahrungen: Auf unserem Balkon (mitten in der Stadt) empfängt mein GFS-Sensor bei freiem Himmel regelmäßig Signale von 8-10 Satelliten. Bei zehn „sichtbaren“ Satelliten liegt der Fehler stabil unter 50 cm.

Abstandsbestimmung

Kennen wir die GPS-Koordinaten zweier Punkte P_1 und P_2 – wie z. B. die aktuelle Position und einen Zielpunkt – können wir daraus den Luftlinien-Abstand bestimmen. Dazu müssen wir zunächst die Differenz der Breiten- und Längengrade der beiden Punkte in Längen umrechnen. Eigentlich ist dafür eine Projektion der gekrümmten Erdoberfläche auf unsere „Kartenebene“ erforderlich – das sparen wir uns jedoch, da wir nur relativ kurze Strecken betrachten, auf denen wir die Erdkrümmung vernachlässigen können. Denn eigentlich verläuft die kürzeste Verbindung zwischen zwei Punkten auf einem Breitenkreis natürlich nicht entlang dieser gekrümmten Linie. Auch mögliche minimale Höhenunterschiede berücksichtigen wir bei der Berechnung nicht.

Und mit noch einer Ungenauigkeit müssen wir leben. Für die Umrechnung der Längen- und Breitengrade in eine Entfernung müssen wir den Erdumfang kennen. Denn der Breitengrad teilt den Erdumfang U in 360° , d. h. der Abstand zweier benachbarter Breitenkreise entspricht einem 360stel des Erdumfangs. Da im Jahr 1793 die Längeneinheit „Meter“ vom französischen Nationalkonvent als der zehnmillionste Teil der Länge des Viertel-Meridians durch Paris definiert wurde, entspräche damit der

Abstand benachbarter Breitenkreise genau $111,111 \text{ km} - 10.000 \text{ km}$ geteilt durch 90° .

Tatsächlich aber ist, wie man inzwischen weiß, der Erdumfang nicht überall gleich, denn die Erde ist keine Kugel, sondern ein (Rotations-) Ellipsoid – daher schwankt der Erdradius nach dem *World Geodetic System* von 1984 (WGS84) zwischen $6.378,137 \text{ km}$ (Äquator) und $6.356,752 \text{ km}$ (an den Polen) [5, 6]. Aus der Länge eines Meridians (von Pol zu Pol) lässt sich ein mittlerer Abstand zweier benachbarter Breitenkreise von $111,133 \text{ km}$ bestimmen.

Der Abstand zweier benachbarter Längengrade – auch *Abweitung* genannt – ist hingegen nicht fest: Er hängt vom jeweiligen Breitengrad ab, da die Breitenkreise sich Richtung Pol verjüngen: Am Äquator (Breitengrad 0) beträgt die Abweitung $111,319 \text{ km}$ (Erdradius von $6.378,1 \text{ km}$). Je größer der Betrag des Breitengrads, desto kleiner die Abweitung – bis sie am (geografischen) Nordpol auf Null sinkt [7].

Wie aber bestimmt man die Abweitung auf einem gegebenen Breitengrad? Da hilft ein wenig elementare Mathematik:

- Der zum Äquator parallele Breitenkreis zu einer Position P am Breitengrad β (Abb. 2) hat den Umfang $U' = 2\pi \cdot r'$. Dabei ist $r' = r \cdot \cos \beta$ mit Erdradius r
- Also gilt: Die Abweitung ω_β zweier benachbarter Längengrade im Breitengrad β berechnet sich wie folgt:

$$\begin{aligned}\omega_\beta &= 2\pi \cdot r' / 360 \\ &= 2\pi \cdot r \cdot \cos \beta / 360 \\ &= U / 360 \cdot \cos \beta\end{aligned}$$

Dabei entspricht U dem Erdumfang am Äquator (Breitenkreisabstand von $111,319 \text{ km}$). Wir vernachlässigen die aus der Erdabplattung resultierende Ellipsoid-Gestalt der Erdkugel (so schwankt der Erdradius in Mitteleuropa nach dem WGS84 zwischen 6.365 und 6.368 km , zzgl. der Meereshöhe des Punkts) [5]; das wirkt sich bei unseren kleinen Distanzen nicht aus.

Damit gilt:

$$\omega_\beta = 111,319 \text{ km} \cdot \cos \beta$$

Damit lässt sich die Abweitung auf dem Eiffelturm (Breitengrad $\beta = 48,858562^\circ$) bestimmen: $\omega_\beta = 73,239 \text{ km}$.

Aus den GPS-Koordinaten unserer Punkte $P_1 = (\beta_1, \lambda_1)$ und $P_2 = (\beta_2, \lambda_2)$ können wir mit der Abweitung ω_{β_1} nun der (Luftlinien-) Abstand d unter Vernachlässigung der Erdkrümmung und etwaiger Höhenunterschiede nach dem *Satz des Pythagoras* berechnen:

$$\Delta_\lambda^2 + \Delta_\beta^2 = d^2$$

Die Abstände auf dem Längen- und Breitengrad erhalten wir dabei wie folgt:

$$\Delta_\beta = (\beta_2 - \beta_1) \cdot 111,319 \text{ km}$$

$$\Delta_\lambda = (\lambda_2 - \lambda_1) \cdot \cos \beta_1 \cdot 111,319 \text{ km}$$

Damit lässt sich der Abstand d (in km) aus den Abständen im Breiten- (Δ_β) und Längengrad (Δ_λ) wie folgt bestimmen (Abb. 3):

$$d = 111,319 \cdot \sqrt{(\beta_2 - \beta_1)^2 + (\lambda_2 - \lambda_1)^2 \cdot \cos^2 \beta_1}$$

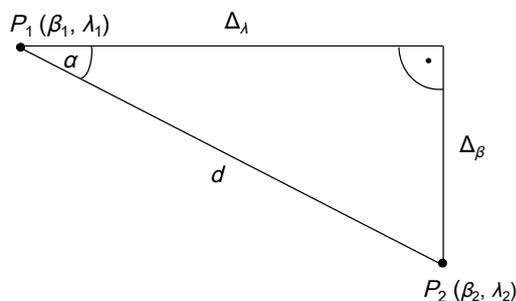


Abb. 3: Bestimmung der Distanz d nach dem Satz des Pythagoras

Richtungsbestimmung

Wenn wir den Abstand zwischen unserer Position P_1 und dem Zielpunkt P_2 berechnet haben, können wir daraus mit ein wenig Trigonometrie die Richtung bestimmen, in die wir fahren müssen, um den Zielpunkt zu erreichen. Denn es gilt:

$$\sin(\alpha) = \frac{\Delta_\beta}{d}, \text{ also:}$$

$$\alpha = \arcsin\left(\frac{\Delta_\beta}{d}\right)$$

Der Arkussinus ist im Intervall $[-1, 1]$ definiert. Da $|\Delta_\beta| \leq d$ und $d > 0$, sofern P_1 und P_2 nicht identisch sind, nimmt Δ_β/d nur Werte aus dem Definitionsbereich an. Der resultierende Winkel α liegt im Intervall $[-90^\circ, 90^\circ]$.

Ist (wie im Beispiel in Abb. 3) $\Delta_\beta \geq 0$, dann ist auch $\alpha \geq 0$, denn es gilt:

$$\arcsin(-x) = -\arcsin(x)$$

Um die Abweichung einer Ausrichtung auf den Zielpunkt P_2 von einer Nord-Ausrichtung zu erhalten, müssen wir 90° zum resultierenden Winkel α addieren.

Von der Lage der Punkte P_2 und P_1 hängt ab, in welche Richtung wir uns von Norden aus drehen müssen, um auf direktem Weg zu unserem Zielpunkt zu gelangen: Ist $\Delta_\lambda < 0$ (liegt also P_2 östlich von P_1), dann müssen wir uns nach rechts (im Uhrzeigersinn) um $\delta = 90^\circ + \alpha$ drehen, sonst nach links um $\delta = -(90^\circ + \alpha)$.

Anstatt uns aber zuerst nach Norden auszurichten, können wir mit Hilfe unseres Kompass-Sensors [2] die Drehrichtung direkt bezüglich unserer aktuellen Ausrichtung bestimmen: Ist der Winkel zur Zielausrichtung δ kleiner als unsere aktuelle Abweichung von Norden, müssen wir nach links, sonst nach rechts drehen, bis wir den Winkel erreicht haben.

Robo Pro-Programm

Jetzt ist es nicht mehr weit bis zur kompletten GPS-Steuerung des Roboters in Robo Pro. Abb. 4 zeigt das Unterprogramm, das die Berechnung des Abstands zum Zielpunkt und der Kompass-Richtung vornimmt. Der Rest des Programms ist euch als Fingerübung überlassen – mit den Treibern aus [1] und [2] ist das keine allzu große Herausforderung.

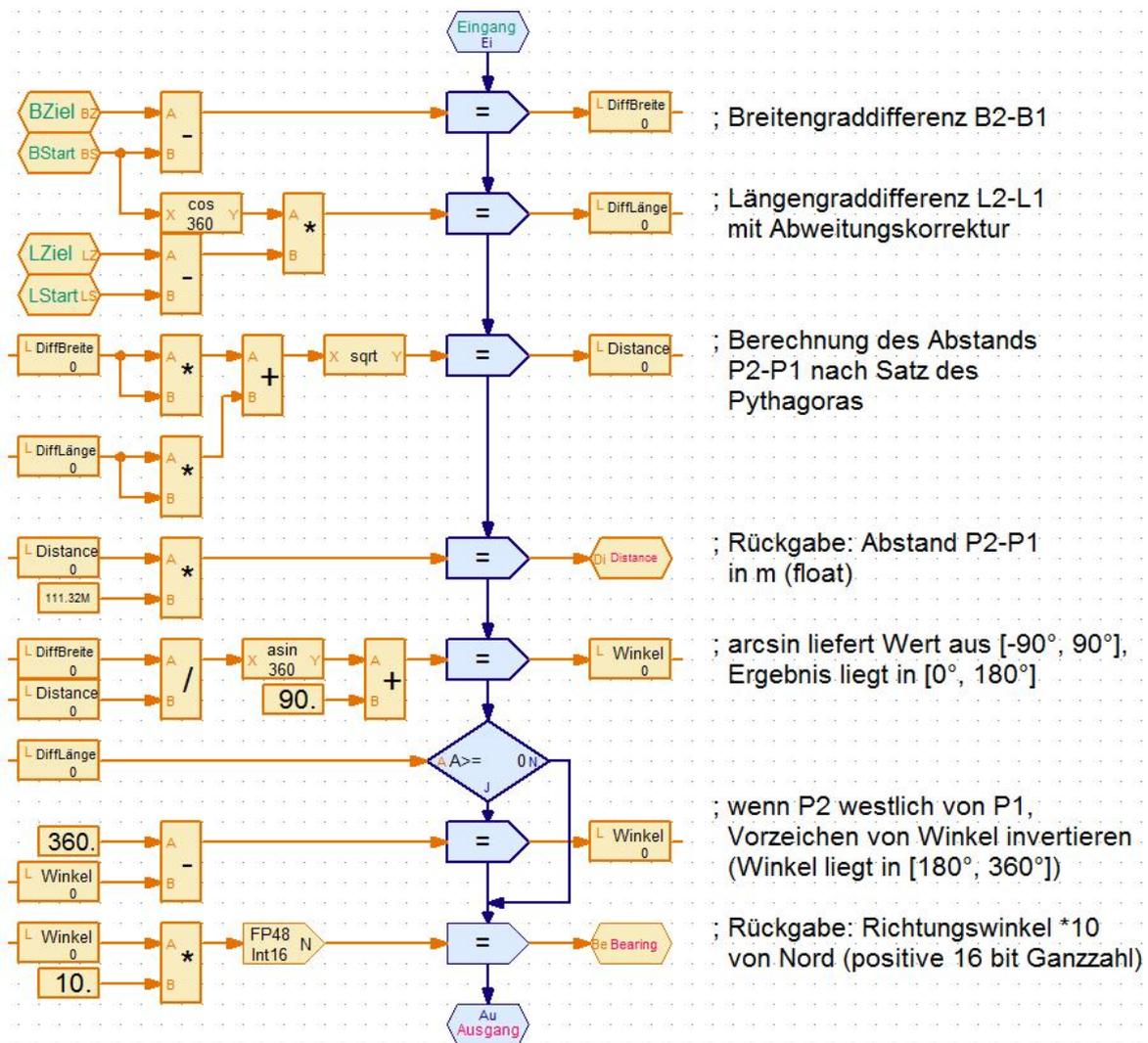


Abb. 4: Robo Pro-Implementierung der Abstands- und Richtungsbestimmung

Quellen

[1] Dirk Fox: *I²C mit dem TX (Teil 6): GPS-Sensor*. [ft:pedia 3/2013](#), S. 54-62.

[2] Dirk Fox: *I²C mit dem TX (Teil 10): Kompass-Sensoren*. [ft:pedia 2/2014](#), S. 57-64.

[3] Wikipedia: [Astronomische Navigation](#).

[4] OpenStreetMap: [Genauigkeit von GPS-Daten](#).

[5] Wikipedia: [World Geodetic System 1984 \(WGS84\)](#).

[6] Wikipedia: [Erdradius](#).

[7] Wikipedia: [Abweitung](#).



Roboter mit neuem TXT-Controller und Kamera