

Editorial

Keep it simple? Stupid.

Dirk Fox, Stefan Falk

Am 26.10.2022 zauberte eine überraschende Nachricht große Fragezeichen auf die Stirn ungezählter MINT-Lehrer: Lego [kündigte an](#), seine Robotik-Reihe [Mindstorms](#) zum Ende des Jahres 2022 einzustellen. Zwar bedeutet das keinen kompletten Rückzug von Lego aus der Mikrocontroller-Welt, allerdings zielt das verbleibende [Spike-Education-Konzept](#) auf Fünft- bis Achtklässler – mit einem [Controller](#), der mit sechs Ein-/Ausgängen eher dünnbrüstig daherkommt, trotz integriertem Gyro-Sensor und MicroPython-Betriebssystem.

Ist das weise Voraussicht oder wurde die Entscheidung vom Markt erzwungen? Geht die Zeit der „Mikrocontroller-Boliden“ zu Ende, oder resigniert Lego vor der Komplexität der zu unterstützenden Betriebssysteme und der Konkurrenz von Raspberry Pi und Co. aus der Maker-Szene – oder gar vor Produktionsengpässen und damit verbundenen Preissteigerungen bei den benötigten Elektronik-Komponenten?

Eines ist gewiss: Seit Jahren ist ein Trend zu „Click-and-Run“-Lösungen im Lernbereich zu beobachten. Technische Komplexität wird mehr und mehr unter glatten, hübschen Oberflächen versteckt, um Kindern schnelle Erfolgserlebnisse zu bieten. Da passt ein smarter „Hub“ besser ins Bild als ein komplexer, erklärungsbedürftiger Mikrocontroller wie der EV3. Auch fischertechnik kokettiert mit diesem Trend; der Baukasten „Early Coding“ und dessen Vorläufer „Mini Bots“ bedienen dieselbe Erwartung nach schnellen, nachdenkarmen Lernerfolgen.

Aber sind das die Lernerlebnisse, die unsere Kinder wirklich brauchen?

Das wirkt so, als wolle man das Erlernen von Schreiben und Lesen dadurch vereinfachen, dass man die Zeichensetzung (die ohnehin kaum noch jemand beherrscht), die Grammatik (viel zu komplex) und die Rechtschreibung weglässt – und zum Schluss das Alphabet auf zwei Buchstaben reduziert (z. B. „d“ und „a“). Dann lernt sogar ein zweijähriges Kind lesen...

Der Vergleich ist übertrieben. Aber wenn wir Wissen so simpel vermitteln, dass nichts mehr zu lernen ist, dann wird natürlich auch nichts mehr gelernt. Tatsächlich sind die technischen Zusammenhänge heute komplexer als vor 50 Jahren, als ich zur Schule ging. Sie zu begreifen erfordert ein tieferes und umfassenderes Verständnis als das, was wir damals benötigten. Und wir tun so, als ob das heute alles viel einfacher zu lernen wäre – nur, weil wir bessere Methoden kennen, um Komplexität zu verstecken.

Von Albert Einstein stammt das Bonmot „Man sollte alles so einfach wie möglich machen, aber nicht einfacher.“ Wenn wir das nicht beachten, bekommen wir eines Tages Schulabgänger, die glauben, Energieknappheit ließe sich durch die Installation weiterer Steckdosen lösen.

Bestärken wir fischertechnik also darin, nicht in Legos Fußstapfen zu treten. Eines Tages wird uns das eine Generation von Ingenieuren danken.

Schöne Weihnachten,
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

Keep it simple? Stupid.	2
Ein Stern für Weihnachten	4
Von der Spiralfeder zum Kameradolly.....	7
Zyklisch ungleichförmig übersetzende Getriebe	12
Schlauchpumpe	20
Logikgatter – elektronisch und mechanisch.....	25
Kettenrätsel.....	38
Großprojekt Seilbahn (Teil 6): Reifenförderer.....	41
Optische Datenübertragung mit fischertechnik.....	52
Elektronikmodule (Teil 9): Direkter Vergleich von Multifunktionsmodul und Grundbaustein	66
Silberlinge: Original oder Nachbau (Teil 9).....	74
Einführung in ftScratch (4): Der Morsetelegraf	80
Hard- und Softwareübersicht zur Ansteuerung von fischertechnik.....	89

Termine

Was?	Wann?	Wo?
Bescherung	24.12.2022	überall

Impressum

<http://www.ftpedia.de>

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

Autoren: Florian Bauer, Axel Chobe, Arnoud van Delden,
Stefan Falk, Dirk Fox, Konrad Fox, Hans-Christian Funke,
Fabian Haas, Peter Krijnen, Lutz Mänche, Rüdiger Riedel, Tilo
Rust.

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.

Modell

Ein Stern für Weihnachten

Lutz Mänche

Dieses Jahr habe ich früh genug angefangen, einen langsam drehenden Stern für das Fenster mit LEDs aufzubauen. Es waren mehrere Versuche nötig, angefangen mit dem alten Schleifring, einem Getriebe und dem Gerüst. Deshalb ist das hier die Kurzform, in einer Bilderstrecke ohne viele Erklärungen.

Alles begann mit dem Drehkranz und einem sechspoligen Schleifring von fischerfriendsman [1] (Abb. 1, 2).

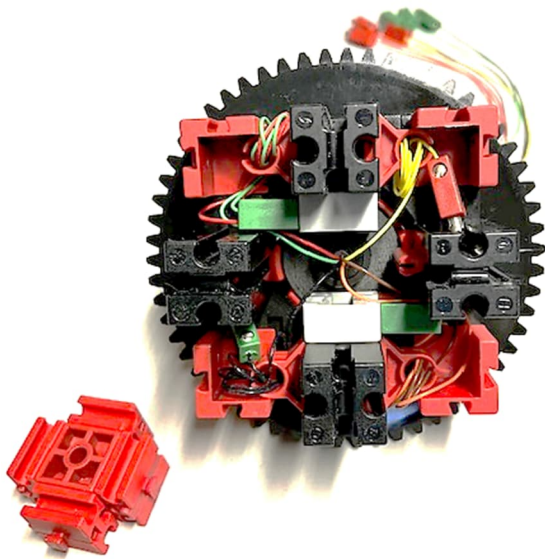


Abb. 1: Der Anfang

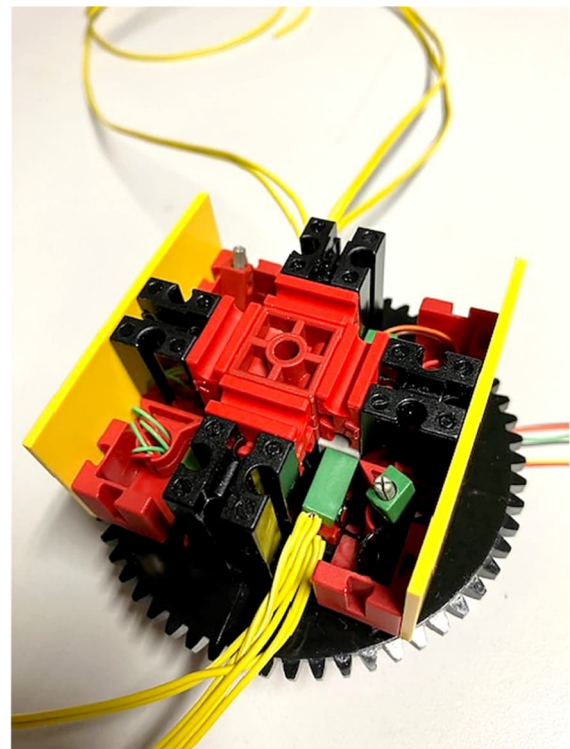


Abb. 2: Der Aufbau für die Achse und die Verkabelung

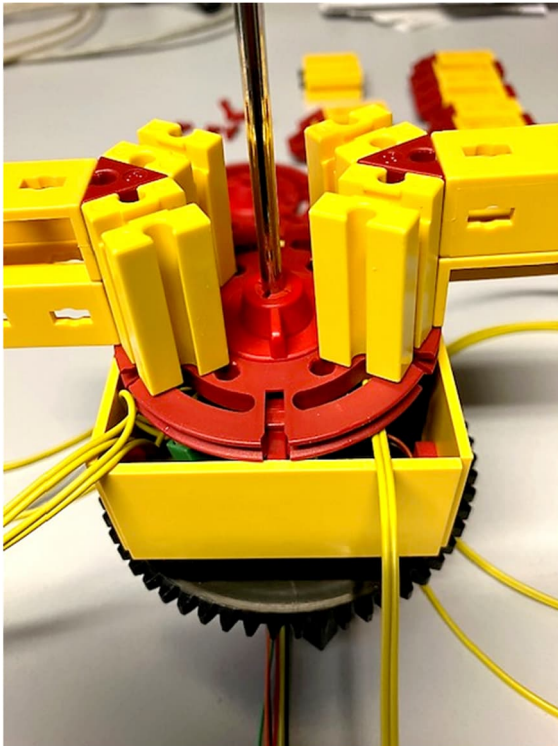


Abb. 3: Die Drehscheibe wird mit zwei Federhaken verbunden. Dann kommt der Aufbau wie in Abb. 4 zu sehen.

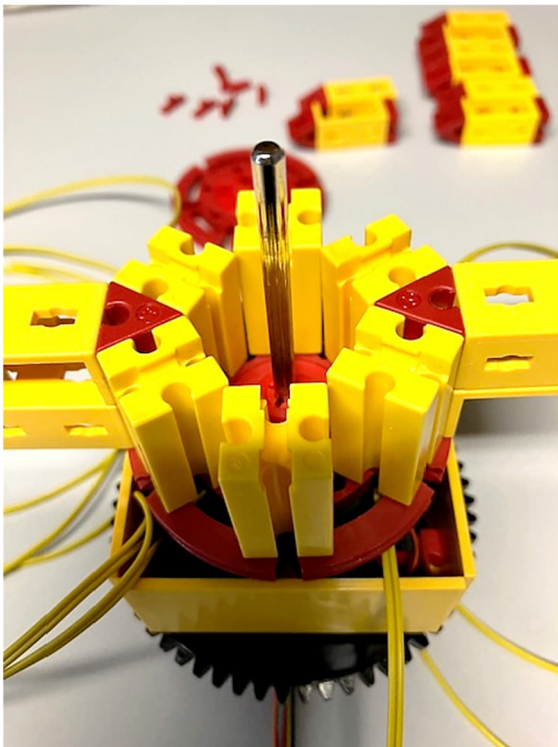


Abb. 4: Der Aufbau auf der Drehscheibe



Abb. 5: Bei den Zacken des Sternes sind die S-Winkelträger 7,5° ([35052](#), in Wirklichkeit $2 \times 7,2^\circ = 14,4^\circ$) verbaut.

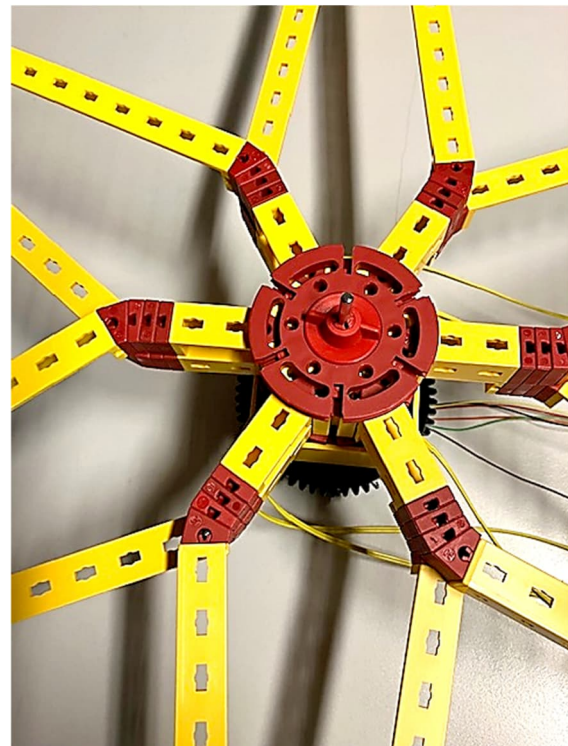


Abb. 6: Die Verbindung mit den sechs Zacken. Die Leitungen für die LEDs wurden durch die Statikteile gezogen.

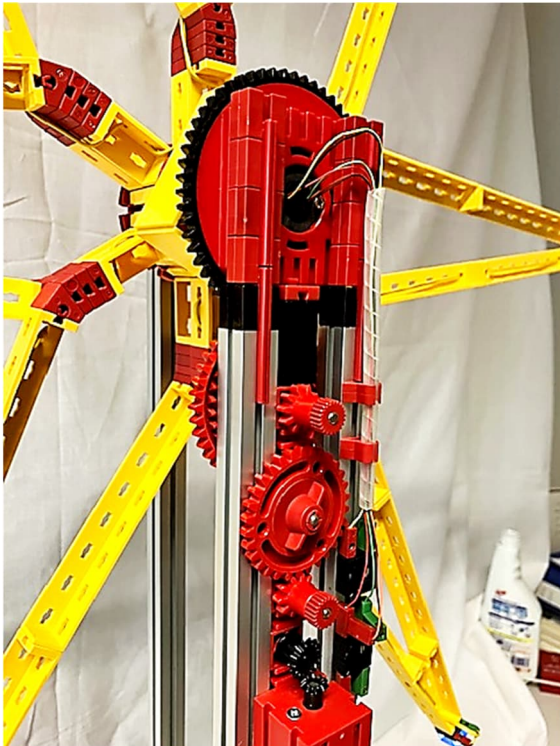


Abb. 7: Das Getriebe ist standardmäßig aufgebaut: klein, groß, klein, groß usw., bis sich der Stern langsam dreht.



Abb. 8: Der Stern von hinten. Das Gerüst ist aus Alu Maker Beams gebaut. Damit die Gardine nicht ins Getriebe kommt, wurde noch ein Schutz aus Statikteilen angebaut.

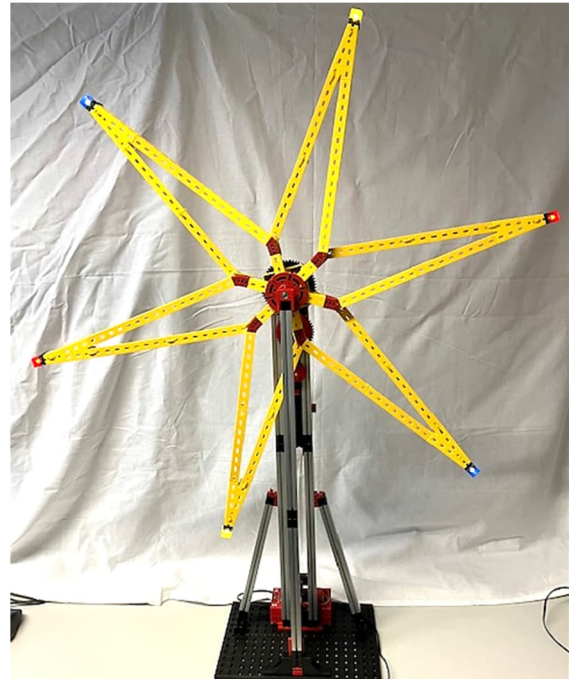


Abb. 9: Der langsame, drehende Stern mit LED-Beleuchtung ist fertig.



Abb. 10: Der Weihnachtsstern im Fenster, von der Straße aus gesehen

Quellen

- [1] Stefan Roth: [fischerfriendsman](#) (ffm).

Modell

Von der Spiralfeder zum Kameradolly

Fabian Haas

Die Folgen eines Besuchs der fischertechnik-Südconvention...

Vorgeschichte

In meiner Vorstellung eines 8x4-Trucks in der ft:pedia [1] blieb eine technische Frage unbeantwortet: Die nach der Federung der Achsen. Diese Frage trieb mich eigentlich schon einige Jahre um; es begann damit, dass ich vor Jahren für Videoaufnahmen eine Kamera auf einen fahrbaren Untersatz, einen sogenannten Dolly¹ oder Kamerawagen, setzen wollte.

Schnell war die Grundplatte 180x90 mit vier Bausteinen 30 zusammengesetzt und an den Metallachsen 120 die Reifen 45 (mit oder ohne Gummiring spielte keine Rolle) montiert, und los ging die Fahrt. Sie war sehr, sehr kurz. Ohne jede Federung in der Konstruktion übertrug sich jede kleine Unebenheit des Laminatbodens direkt auf die Kamera und führte zu deutlichen Stößen und Wackeln in den Videoaufnahmen.

Damit verschwand das Projekt wieder in der Teilekiste.

Nach einigen Jahren in der Kiste kam fischertechnik wieder hervor und es fanden sich Kunststofffedern 26 ([38633](#), blau) in der Teilesammlung. Bei der Entwicklung des 8x4-Trucks gab es einige erfolglose Versuche, diese Federn zu verwenden. Das wesentliche Problem ist, dass die Feder eine zusätzliche Führung braucht, da sie sonst in alle Richtungen ausweichen kann. Vorne-

hinten, links-rechts – sie ist in jede Richtung frei beweglich. Das ist sicher für einige Anwendungen von Vorteil, jedoch nicht für eine Radaufhängung. Eine Einschränkung der Freiheitsgrade ergibt sich durch die Verbindung zweier Federn über eine Achse 120, jedoch bleiben sie weiterhin senkrecht zur Achse sehr frei beweglich.

Lösungsidee

Die Lösung blieb offen, bis ich die fischertechnik-Südconvention im Fördertechnikmuseum Sinsheim im Herbst 2022 besuchte. Dort zeigt mir ein Aussteller sein Fahrzeugmodell, für das er mehrere Radaufhängungen ([31258](#)) benutzt. Diese Teile hatte ich bisher völlig übersehen; außerdem sind sie nicht mehr so ganz einfach verfügbar. Die Radaufhängung erschien mir recht klobig und groß, mit einem Platzbedarf von etwa drei Bausteinen 30, und sie ist nur mit zwei I-Streben 30 komplett, die die Achse führen. Dennoch war diese Radaufhängung der Startpunkt für den Weg, auf dem sich die folgenden Lösungen bis zum Kameradolly ergaben – dank dem Aussteller auf der Südconvention in Sinsheim 2022.

Ausgangspunkt war ein einfacher Nachbau der Radaufhängung. Wie viele Projekte fing es auf einer Grundplatte an. Aber kleiner als

¹ Dolly: *A wheeled apparatus used to transport a movie camera steadily along the ground or along a track.*

vor Jahren; es sollte zunächst die Grundkonstruktion getestet werden (d. h. Grundplatte 120x60, Abb. 1).

Zwischen zwei Bausteinen 30 ließen sich vier I-Streben 30 auf einer Kunststoffachse 40 einbauen, außen mit Riegelscheiben gesichert. In passendem Abstand wurde eine Achse 120 durch die beiden Federböden und die I-Streben 30 geführt (Abb. 1). Das ist praktisch ein direkter Nachbau der Radaufhängung, nur die Führung der Achse durch die I-Streben 30 erfolgt in der Mitte, und nicht auf den Seiten. Der Baustein 5, auf dem die Kunststofffedern 26 stehen, erfüllt keine besondere Funktion; der Baustein 15x30x5 sichert den Abstand zwischen den Bausteinen 5.

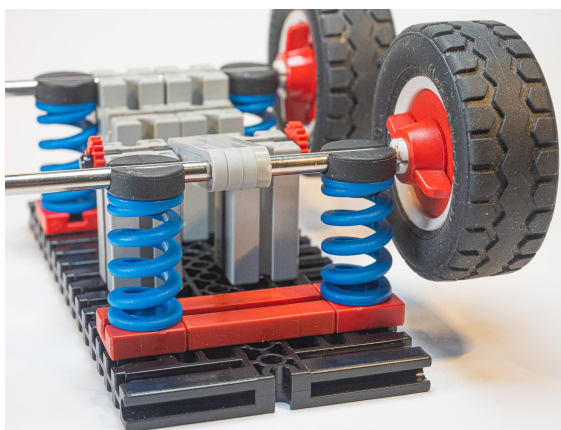


Abb. 1: Nachbau der fischertechnik-Radaufhängung, die sich noch sehr am Original orientiert (siehe auch Abb. 2).

Eine Eigenheit dieser Konstruktion ist, dass sich die Räder auf der Achse 120 bei Belastung auf einer Kreisbahn um die Grundbausteine 30 bewegen. Aufgrund des sehr kurzen Federwegs ist der Betrag gering. Außerdem waren, bedingt durch die Länge der I-Strebe 30, Teile der Radaufhängung außerhalb des Raddurchmessers zu sehen; sie waren also hinter dem Rad sichtbar.

Da ließ sich doch sicher etwas machen (beginnender Optimierungsvirus). Weitere Versuche führten zur Parallelführung zwischen Bausteinen 30 mit Hilfe von Bausteinen 15. Dadurch bewegt sich die

Achse 120 in den Kunststofffedern exakt auf und ab, und der ganze Block verschwindet hinter dem Reifen (Abb. 2).

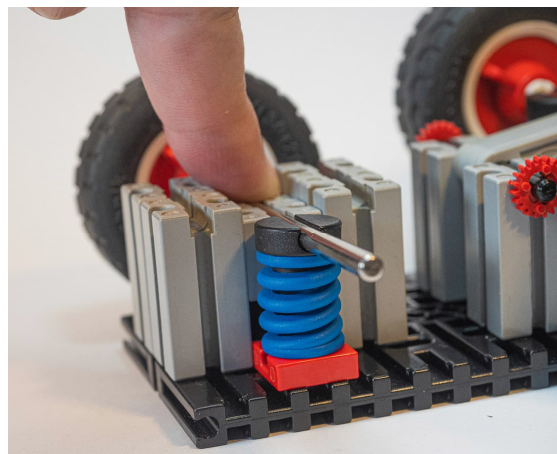


Abb. 2: Eine Geradführung der Achse beim Federn, insgesamt noch sehr massiv

So weit, so gut! Eine Grundplatte ist ideal für die Entwicklung; wie aber sieht das Ganze in einem Fahrzeug mit Rahmen aus?

Als Basis dienen die Aluprofile 210, wie beim 8x4-Truck. Die I-Streben 30 schienen mir zu kurz, zumal es darum ging, den Radius um die Befestigung zu vergrößern. Daher verwendete ich I-Streben 75, die mit Baustein 7,5 über Federnocken am Aluprofil angebracht wurden.

Riegelscheiben und Klemmbuchsen sichern die Position der Achsen 60 und 120.

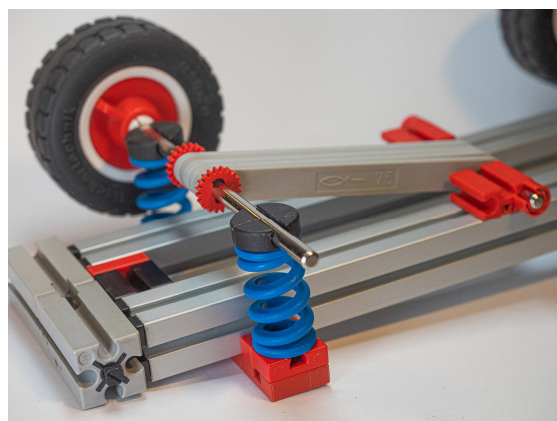


Abb. 3: Umsetzung der Radaufhängung auf ein Fahrzeug in Rahmenbauweise, der Platz wurde für I-Streben 75 genutzt. Möglicherweise lässt sich der Aufbau für eine motorische Kontrolle der Fahrzeughöhe nutzen.

Diese Führung geht aber weit über den Reifen hinaus, was mich ja oben schon störte. Daher ersetzte ich die Führung der Achse durch eine Führung der Feder, indem ich die Kunststofffeder 26 vorne und hinten mit je zwei Bausteinen V15 Eck begleitete.

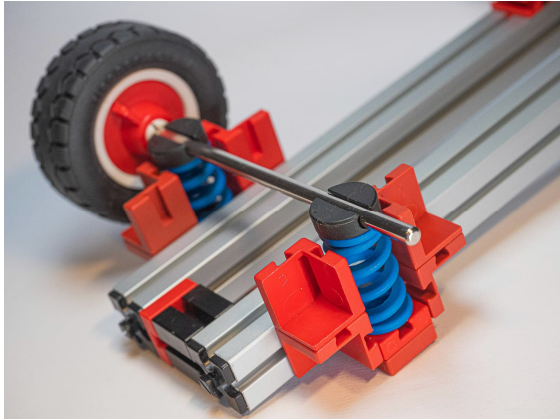


Abb. 4: Anstatt die Achse zu führen, mit den Problemen einer erhöhten Reibung, wurde hier die Führung der Feder realisiert.

Befestigt wird die Kunststofffeder 26 über den Baustein 5 mit einem Baustein 15x30x5, der über eine Federnocke oder ein Verbindungsstück 25 von oben mit dem Aluprofil verbunden wird. Das Ganze führt zu einer sehr geringen Höhe der Befestigung, und die gesamte Führung mit Feder ist schmaler als der verwendete Reifen.²

Die geringe Breite der gesamten Konstruktion – sie „versteckt“ sich ja hinter einem Reifen – ermöglicht die beliebig enge Positionierung von Reifen (solange sie frei drehbar sind) auf dem Aluprofil. Die Abstände ergeben sich einzig durch den Reifendurchmesser, nicht aber durch die Radaufhängung. Daher lassen sich beliebig viele Reifen auf dem Profil positionieren. In meiner derzeitigen maximalen Ausbaustufe sind es vier Achsen, also acht Reifen.

Einen wichtigen Aspekt der Federung habe ich bisher noch nicht diskutiert: die Reifen.

Diese sind integraler Bestandteil jeder Federung, gerade bei Autos, und hier ergab sich zu meinen Anfängen vor einigen Jahren eine entscheidende Änderung. Anstatt der harten Reifen 45 oder Reifen 60 nutze ich jetzt die Gummireifen 50 ([32913](#)) für die Felge 30. Allein dadurch ergibt sich schon ein deutlich weicherer, vibrationsärmerer Lauf. Diese Reifen waren mir damals nicht bekannt, und das ganze zeigt, wie wichtig eine mentale Teileliste ist!



Abb. 5: Gesamtansicht des Dollys

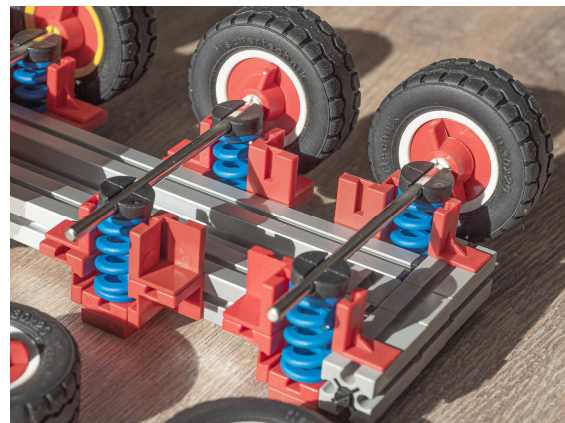


Abb. 6: Verwendete Radaufhängung

Hier also die Gesamtkonstruktion, wie sie derzeit realisiert ist (Obacht, Optimierungsvirus!). Aufgrund des Gewichts der Kamera und der Kamerahalterung entschied ich mich für die vierachsige Konstruktion,

² Die beiden unteren, d. h. der Achsen-nahen Bausteine V15 Eck, lassen sich durch Winkelsteine 10x15x15 ohne Verlust von Führung ersetzen

mit der Führung der Federn durch die Bausteine V15 Eck und Winkelsteine 10x15x15 mit den weichen Gummireifen 50 für die Felge 30.

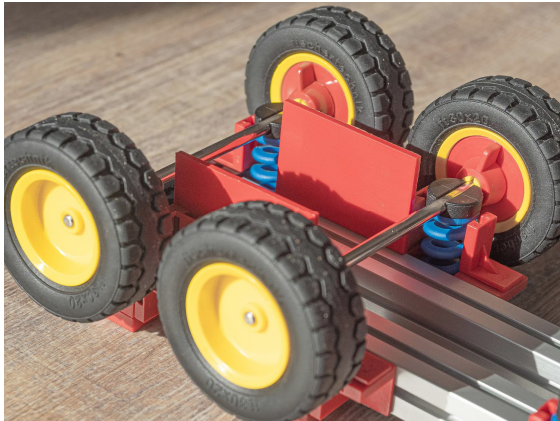


Abb. 7: Die Abstände zwischen den Achsen lassen sich z. B. mit Bauplatten 30x45 sehr gut justieren. Da die Bauplatten nicht weiter stören (z. B. Bodenfreiheit) verblieben sie einfach in den Bausteine V15 Eck.

Eines fehlt aber noch: Die Verbindung mit der „realen Welt“, der Kamera. Glücklicherweise hatte ich schon eine Klemme entwickelt [2], die ich soweit modifizierte, dass sie meine Kamera sicher halten kann.

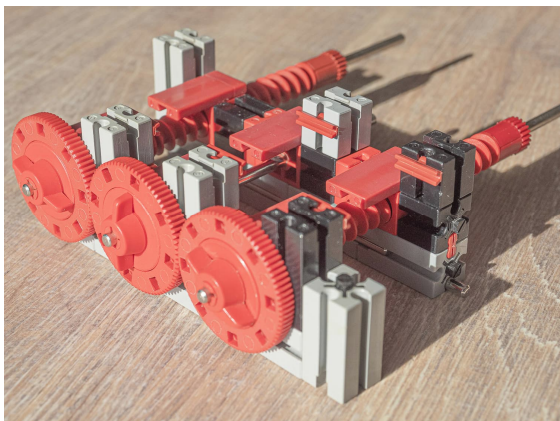


Abb. 8: Mit der modifizierten Klemme lässt sich die Kamera sicher halten und auf dem Dolly befestigen

Die Klemme wird dann mit vier Federn in den Aluprofilen befestigt und so justiert, dass die Kamera horizontal schaut. Durch das Gewicht von Kamera und Objektiv kann sich der Dolly neigen, was

durch die Positionierung auf dem Dolly ausgeglichen werden muss.



Abb. 9: So geht es! Weiche Kamerafahrten über holprigem Laminat sind nun möglich.

Nachtrag

Wer eine Antriebsachse in alle Richtungen federn möchte, findet die Konstruktion in Abb. 10 möglicherweise interessant, in der die Achse ausschließlich von Kunststofffedern 26 gehalten und geführt wird. Das ist vermutlich weniger etwas für ein Fahrzeug, eher für eine Antriebswelle.

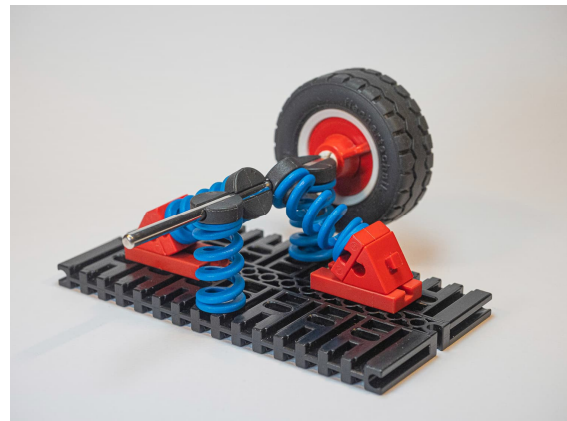


Abb. 10: Führung und Federung einer Achse nur mit Kunststofffedern 26 (Grundplatte 120x60, Baustein 15x30x5, Bausteine 5 und Winkelsteine 60 für den richtigen Abstand)

Abbildungen 11-13 zeigen die derzeitige Ausbaustufe des Dollys. Der Abstand zwischen den Aluprofilen wurde um einen Baustein 30 vergrößert, was die Stabilität gegen seitliches Schwingen deutlich erhöht.

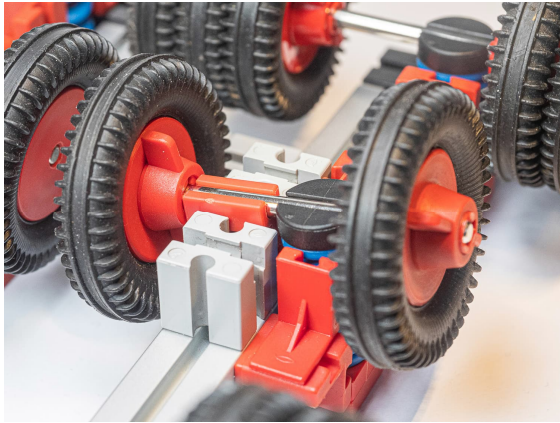


Abb. 11

Aus Materialgründen verwende ich die Reifen 45 (bevorzuge aber die genannten Gummireifen). Als alternative Aufhängung passte ich die Geradföhrung aus Abb. 2 an, indem ich den Baustein 7,5 als Föhrungselement zwischen den Bausteinen 15 einsetze.

Somit ergibt sich eine gefederte Einzelachsaufhängung, was die Dämpfung hoffentlich noch mal verbessert.



Abb. 12

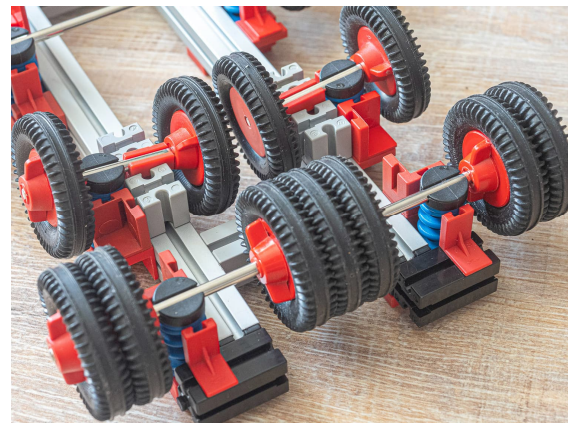


Abb. 13

Referenzen

- [1] Fabian Haas: *Chassis für 8x4-Truck*. [ft:pedia 1/2022](#), S. 21–26.
- [2] Fabian Haas: *Eine Kabelklemme mit beweglichen Klemmbacken*. [ft:pedia 1/2022](#), S. 27–30.

Getriebe

Zyklisch ungleichförmig übersetzende Getriebe

Stefan Falk

Im Jahr 2004 überraschte der deutsche Ingenieur Wilhelm Klopmeier die Welt mit einer neuartigen Mechanik für ein einfach zu konstruierendes Getriebe, das zyklisch ungleichförmig übersetzt. 2011 ehrte er die fischertechnik-Convention mit einem Besuch, auf dem er einige Varianten davon zeigte – er hatte sie nämlich mit fischertechnik erfunden. Dieser Beitrag zeigt ein äußerst einfach zu bauendes Getriebe mit ähnlichen Eigenschaften.

Einführung

Die Arbeiten von Herrn Klopmeier [1, 2] sorgten weithin und natürlich auch bei fischertechnikern für Furore. In den Quellen finden sich Links zu diesen Varianten.

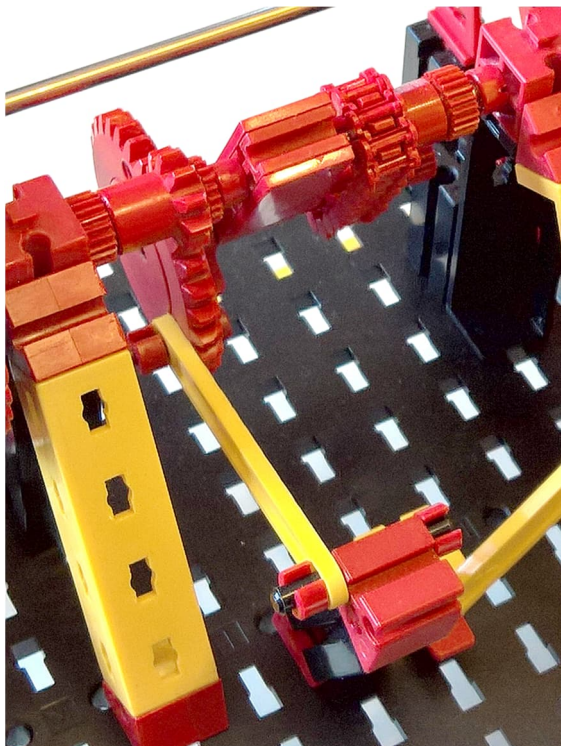


Abb. 1: Ein „Klopmeier-Getriebe“ [9]

In Abb. 1 sieht man eine gelbe Statikstrebe als Hebel, die (unten im Bild in der Achse) schwenkbar, aber ansonsten fix gelagert ist. Sie endet drehbar in einer der Bohrungen eines Zahnrads Z30 gelagert. Das Z30 wird

vom Z10 links im Bild angetrieben und ist in einem schwenkbaren Hebel – einer Reihe von BS 7,5 – gelagert. Die Achse des Z30 endet auf der anderen Seite des Hebels in einem weiteren Z10 (im Bild vom Hebel verdeckt), das über eine Kette mit dem Z10 rechts – dem Abtrieb – verbunden ist. Die Achsen der beiden Z10 treffen sich also fluchtend in der Mitte des BS 7,5-Hebels, sind aber getrennte Achsen.

Was macht nun dieses Getriebe?

Zyklisch ungleichförmige Übersetzung

Im Video [10] sieht man es sehr gut: Diese raffinierte Kinematik führt dazu, dass bei gleichförmiger (d. h. mit konstanter Drehgeschwindigkeit erfolgender) Drehung der Antriebsachse (in Abb. 1 die Achse mit dem linken Z10) die Abtriebsachse in zwei Punkten eine Besonderheit in ihrer Drehung zeigt:

- Die Drehung ist *ungleichförmig*: Die Abtriebsachse dreht sich nicht immer gleich schnell. Je nach der Geometrie des Getriebes kann sie langsamer und schneller drehen, aber auch bis zum Stillstand oder sogar zum Rückwärtsdrehen gebracht werden.
- Die Drehung ist aber trotzdem *zyklisch*: Das heißt, die Übersetzung erfolgt periodisch wiederkehrend. Nach einer

bestimmten Anzahl von Umdrehungen der Antriebsachse ist die Abtriebsachse wieder in der Ausgangsposition, und der Verlauf ihrer Drehgeschwindigkeit wiederholt sich exakt.

Das hat viele Anwendungsmöglichkeiten, wie etwa:

- Ölförderpumpen: Man hat ein Interesse daran, dass die Pumpbewegung (aufwärts) möglichst die ganze Zeit mit der maximal möglichen Förderleistung erfolgt. Zu schnelle Aufwärtsbewegungen könnten nämlich zum Ansaugen von Luft anstatt von Öl führen. Die Abwärtsbewegung hingegen soll möglichst schnell erfolgen, weil sie ja unproduktiv ist. Ein normales Exzentergetriebe leistet das nicht, wohl aber ein Klopmeier-Getriebe.
- Exzenter-Pressen: Dabei möchte man vielleicht, dass die Presse lange oben bleibt, um Zeit zum Werkstückwechsel zu haben, und auch lange unten, damit das Werkstück länger gepresst wird. Die Auf- und Ab-Bewegungen sollen aber wieder, da unproduktiv, schnell erfolgen.

Herr Klopmeier fand noch viele andere Einsatzmöglichkeiten, sogar für Fahrradantriebe.

Eine besonders einfache Bauart

Für einen Workshop – erst im MINT-Ferien-Camp des CyberForum Karlsruhe [12], dann in einem für Lehrkräfte/AG-Leitungen – habe ich eine Bauart entwickelt, die mit fischertechnik fast schon trivial einfach herzustellen ist. Sie kommt mit wenigen Teilen auf kleinem Raum aus, sodass sie auch für Workshops und Schulstunden geeignet ist.

Ein einfaches Zahnradgetriebe mit zwei Zahnrädern kennen wir zur Genüge (Abb. 2): Das Übersetzungsverhältnis wird durch die Zähnezahlen bestimmt, vor allem aber ist es *konstant*. Wenn wir immer gleich

schnell an der Kurbel drehen, so mag sich das Abtriebs-Rad je nach den Zähnezahlen schneller oder langsamer drehen, aber es dreht sich auch *immer gleich schnell*.

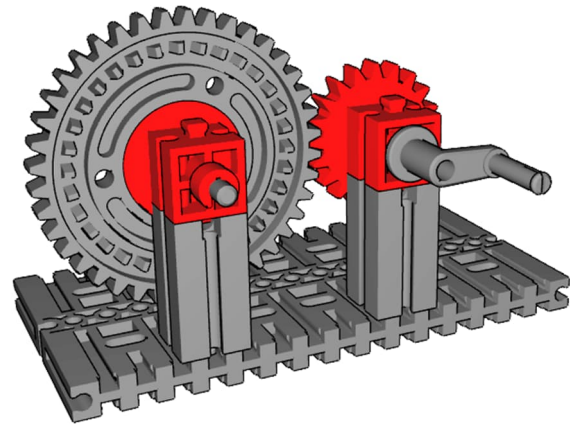


Abb. 2: Einfaches Zahnradgetriebe

Jetzt kommen zwei Tricks:

1. Wir fügen ein drittes Zahnrad zwischen die beiden ein, das aber *nicht* fest gelagert ist. Vielmehr stellen wir – z. B. durch Streben – nur sicher, dass die Abstände zwischen je zwei Zahnrädern gleich bleiben und so alle Zahnräder immer sauber miteinander kämmen.
2. Das erste Zahnrad lassen wir dadurch um die Kurbelachse kreisen, dass wir es selbst wieder auf eine Kurbel setzen, und zwar auf den exzentrischen Teil – also auf den „Griff“ der Kurbel.

Das sieht dann so aus:

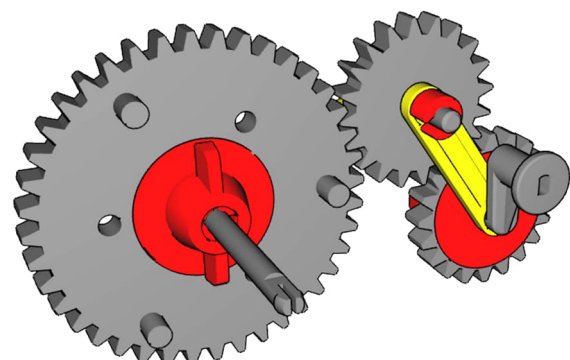


Abb. 3: Die neue Mechanik – man beachte, dass das rechte Z20 auf dem Griff einer Kurbel festgeschraubt ist

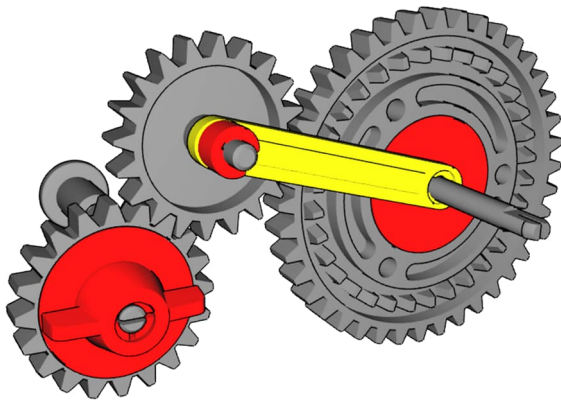


Abb. 4: Die andere Seite des Getriebes

Im fertigen Modell stellt sich das so dar (zu allen hier dargestellten Modellen findet sich die ausführliche Bauanleitung in [13]):

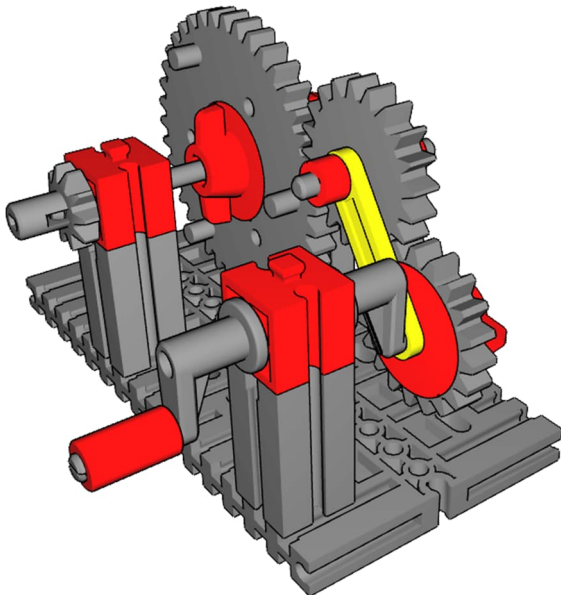


Abb. 5: Das erste Modell

Beim Aufbau muss also auf den Griff der zweiten Kurbel **zuerst die I-Strebe 30 gesteckt** und erst danach das Zahnrad Z20 mit der Flachnabe **auf dem Griff festgeschraubt** werden. Die I-Strebe 30 realisiert dann genau die 30 mm Abstand zwischen den beiden Z20, die zum sauberen Kämmen notwendig sind.

Das Zwischen-Z20 ist in diesem Bauvorschlag ein Freilauf-Kettenzahnrad Z20 m1,5 (schwarz, [31779](#)). Das wird also nicht festgeschraubt, ist aber angenehm schmal, sodass wir auf seiner anderen Seite mit der zweiten Strebe weitermachen können. In

diesem ersten Modell ist das dritte Zahnrad ein Z40, deshalb braucht es für den korrekten Abstand zwischen dem Ketten-Z20 und dem Z40 eine I-Strebe 45.

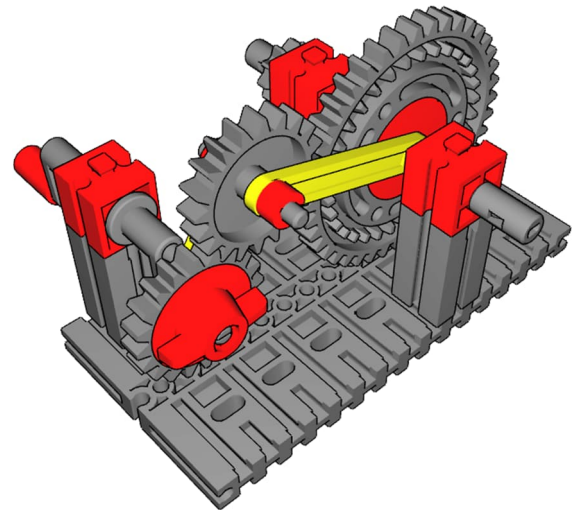


Abb. 6: Die Rückseite des ersten Modells

Wohlgemerkt also: Sowohl das Z20 auf der Kurbel als auch das Z40 am Ende des Getriebes werden jeweils *festgeschraubt*. Nur das Zwischenzahnrad sitzt einfach auf seiner kleinen Achse (eine Achse 30 genügt dafür).

Ausprobieren!

Jetzt können wir an der Antriebskurbel drehen und beobachten, wie sich das Getriebe verhält. Das Zwischenzahnrad wird je nach Kurbelstellung angehoben oder abgesenkt, greift aber immer in beide äußeren Zahnräder ein. Außerdem dreht sich das erste Z20 zusätzlich um seine eigene Drehachse. Es dreht also das Zwischenzahnrad. Das überlagert sich mit der Auf- und Ab-Bewegung des Zwischenzahnrad.

Deshalb dreht sich das Z40 insgesamt mal schneller, mal langsamer. Damit haben wir die *ungleichförmige Übersetzung*. Insgesamt, also über eine komplette Kurbelumdrehung gesehen, ist das Übersetzungsverhältnis immer noch $20:40 = 1:2$, aber das Z40 dreht sich eben nicht immer gleich schnell, wenn die Kurbel mit konstanter Geschwindigkeit gedreht wird.

Nach einer (genau genommen wegen der 1:2-Untersetzung zwei) Umdrehung(en) der Antriebskurbel ist aber der Anfangszustand exakt wieder erreicht. Damit haben wir also ein *zyklisch variabel übersetzendes Getriebe* hergestellt.

Was machen wir nun damit?

Als einige von unzähligen Anwendungsbeispielen mögen drei Modelle dienen, die mit solchen Getrieben hergestellt werden können:

Ein sanftes Schrittschaltwerk

Wenn wir das erste Modell um ein Z30 mit aufgesetzter Drehscheibe ergänzen und auf der Drehscheibe eine Markierung anbringen, erhalten wir ein Getriebe, das mit jeder Umdrehung der Kurbel die Drehscheibe um $1/6$ Drehung bewegt, und zwar sanft beschleunigend und bremsend, mit einer Drehpause im Anschluss (Abb. 7).

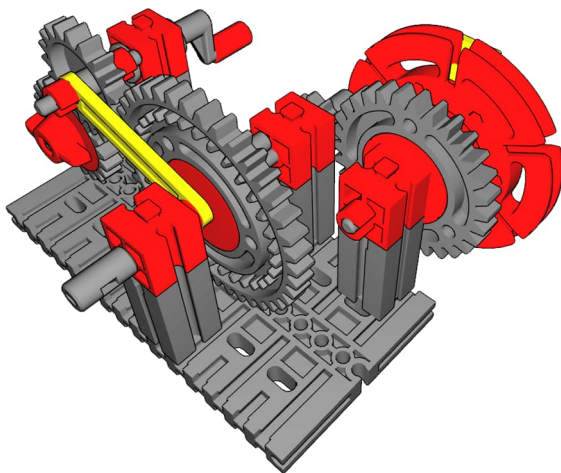


Abb. 7: Ergänzung um ein Z30 plus Drehscheibe

Die nächste Maschine, die – man stelle sich die Drehscheibe liegend angebracht vor – in einem Industriemodell Werkstücke bearbeitet, die auf je einem Segment der Drehscheibe liegen, bewirkt also ganz ohne komplizierte Steuerung mit Elektromechanik, Elektronik oder Computerei eine schrittweise Drehung der Drehscheibe – bei einem ganz simpel stetig laufenden Motor.

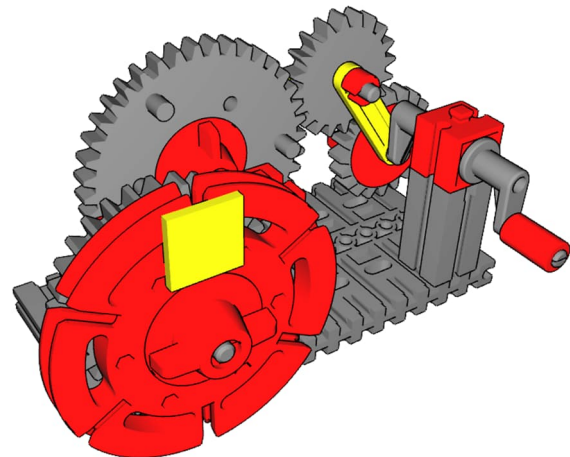


Abb. 8: Markierung am sanften Schrittschaltwerk

Außerdem ist nicht zu vernachlässigen, dass die Beschleunigung und das Bremsen *sanft* geschehen. Es gibt *keinen Ruck* in der Mechanik, es verrutscht also kein Werkstück, das lose auf der Drehscheibe liegt, und auch bei schweren Bauteilen wird die Mechanik niemals ruckartig schwer belastet. Ganz anders, als wenn man einen Motor immer hart ein- und ausschalten würde!

Ein sanfter Linear-Schrittverschluss

So etwas können wir mit einer Zahnstange kombinieren und erhalten einen sanft schrittweise arbeitenden Linearvorschub:

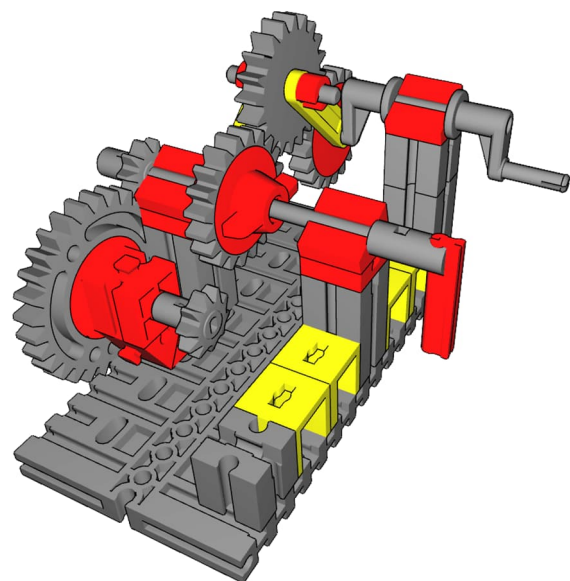


Abb. 9: Linearvorschub, noch ohne eingelegte Zahnstange

In diesem Modell wurde anstatt des Z40 ein drittes Z20 eingesetzt. Die richtigen Abstände zwischen den beiden Z20 auf Flachnaben – so, dass sie sich beim Kurbeln nicht berühren – werden durch Lagerung der Achsen in Winkelsteinen 30° und unterschiedlicher Höhe der Achslager erreicht: Beim Antrieb ein Baustein 30 plus Baustein 15, beim dritten Z20 ein Baustein 30 plus Baustein 5. Die Achse des Z30 sitzt in einem Baustein 15 mit Bohrung, der mittels zweier Winkelsteine 60° auf der Grundplatte befestigt ist.

Nur damit man die ungleichförmige Drehbewegung gut beobachten kann, steckt auf der Achse des dritten Z20 ein Rastadapter und ein Verbinder 30 als „Zeiger“.

In das letzte Z10 lassen wir nun eine Zahnstange eingreifen:

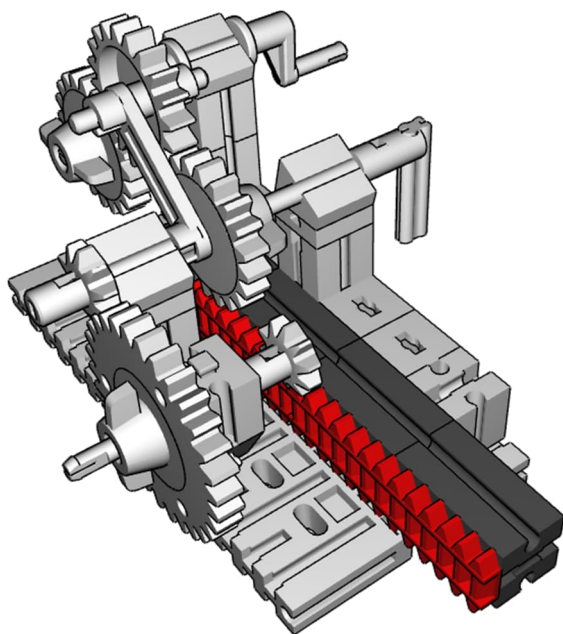


Abb. 10: Die Zahnstange wird sanft schrittweise verschoben

Eine besondere Presse

Im nächsten Modell kommt wieder ein Z40 als drittes Zahnrad zum Einsatz. Es handelt sich um eine klassische Exzenter-Presse (wie sie z. B. zum Tabletten-Pressen eingesetzt wird). Nur hat unser Modell trotz des recht einfachen Aufbaus zwei Besonderheiten zu bieten: Der Presskopf bleibt sowohl oben als auch ganz unten (in den sogenannten *Totpunkten* des Exzenters) sehr lange stehen. Oben, damit möglichst viel Zeit zum Werkstückwechsel zur Verfügung steht, und unten, damit das Werkstück möglichst lange und intensiv gepresst wird. Die unproduktiven Auf- und Ab-Bewegungen hingegen werden mit hoher Geschwindigkeit abgewickelt (Abb. 11-13).

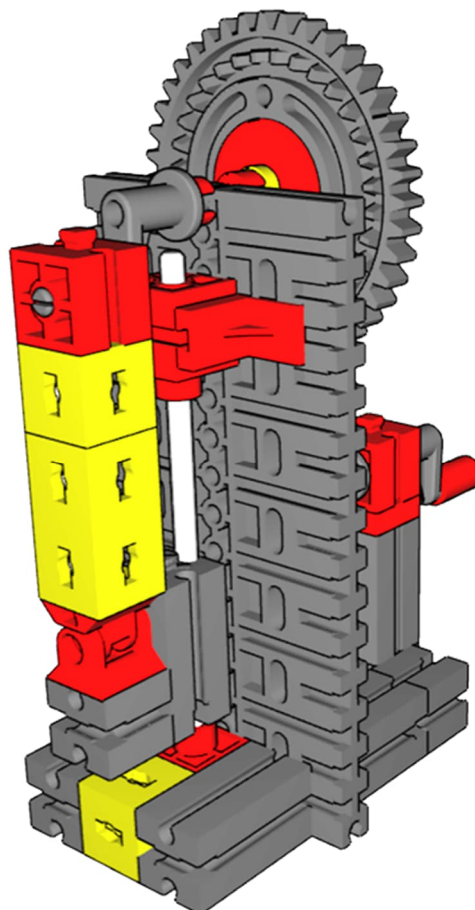


Abb. 11: Die Vorderseite der Presse: Oben der Exzenterantrieb, unten der Presskopf. Das zu pressende Werkstück wird auf den Fuß aufgelegt und von oben gepresst.

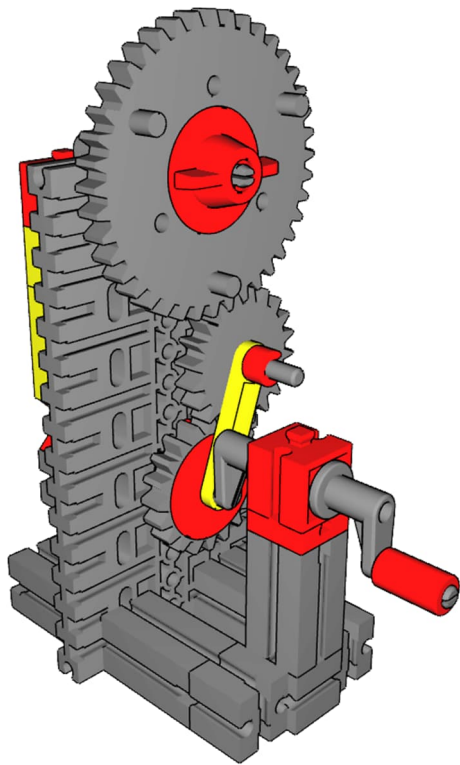


Abb. 12: Die Antriebsseite der Presse. Hier von den Zahnrädern verdeckt und also nicht sichtbar hängt die I-Strebe 45 zwischen dem Zwischen-Z20 und dem Z40.

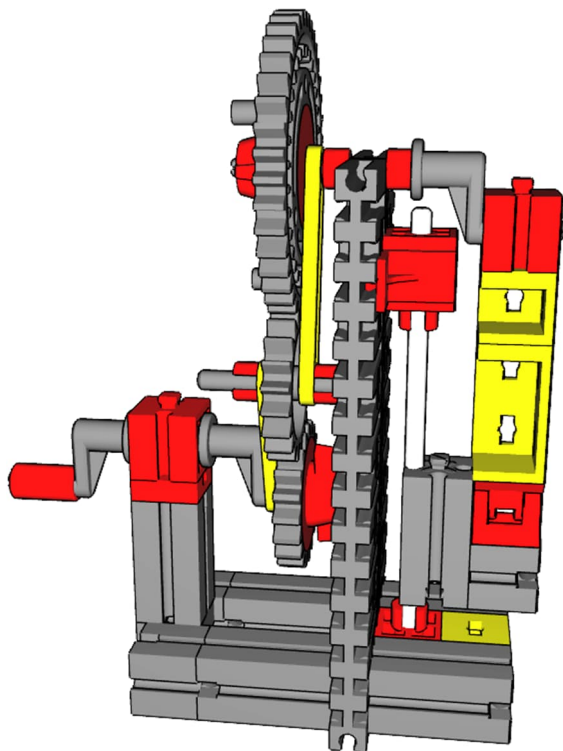


Abb. 13: Seitenansicht der Presse

Geht das auch mit anderen Zahnradkombinationen?

Aber ja! Ich kann nur ermuntern, mit dieser Art Getriebe ausführlich zu spielen und zu experimentieren.

Im 15-mm-Raster (und so mit I-Streben realisierbar) liegen die Abstände von mindestens folgenden fischertechnik-Zahnradkombinationen: Z10/Z30, Z20/Z20, Z20/Z40 und Z30/Z30.

Immerhin im 7,5-mm-Raster liegen die Kombinationen Z10/Z20, Z10/Z40 und Z20/Z30. Dafür muss man anstatt Streben Gestänge aus Grundbausteinen verwenden und mit einem Baustein 7,5 den richtigen Abstand herstellen. Das Modell wird dadurch etwas breiter, aber die Funktion ist dieselbe.

Das Verhältnis zwischen dem ersten und dem dritten Zahnrad legt das generelle Übersetzungsverhältnis fest. Die Zähnezahzahl des Zwischenzahnrad ist dabei – genau wie bei einem „klassischen“ Getriebe – unerheblich. Aber: Die Zähnezahlen der ersten beiden Zahnräder sowie der Abstand zwischen dem ersten und dritten Zahnrad bewirken auch eine Änderung der „Weichheit“ der Bewegung, also der Stärke und dem zeitlichen Verlauf der Beschleunigung.

Insbesondere das erste Zahnrad ist von Bedeutung:

- Unser Z20 hat ungefähr denselben Radius wie die Rastkurbel. Deshalb geht die Abtriebs-Bewegung von Stillstand bis Maximalgeschwindigkeit.
- Schrauben wir ein größeres Zahnrad auf die Kurbel (also etwa ein Z30 oder Z40), so ist die Bewegung zwar immer noch ungleichmäßig schnell, geht aber nie bis zum Stillstand zurück.
- Der verblüffendste Effekt wird sichtbar, wenn wir ein Zahnrad auf die Kurbel setzen, dessen Radius *kleiner* als der der Kurbel ist – etwa ein Z10. Dann gibt es einen Bereich der Kurbelstellung, in der

der Abtrieb sich sogar *rückwärts* bewegt!

- Statt der Rastkurbeln kann man auch solche mit anderen Radien verwenden. Auch ein Z30 z. B. hat ja Bohrungen, in die man eine Achse einigermaßen fest einstecken und darauf wieder ein Zahnrad befestigen kann. Das schreit ebenfalls danach, ausprobiert zu werden.

Ihr merkt schon: Ihr müsst an Eure fischer-technik-Kästen und das alles ausgiebig testen. Bestimmt fallen euch noch viel mehr Anwendungsmöglichkeiten ein.

Kaskaden von solchen Getrieben

Zum Abschluss wollen wir uns noch weiter wagen: Wer genügend Bauteile besitzt, kann mehrere solcher Getriebe hintereinander schalten. Das Modell in Abb. 14 verwendet drei Stufen von Getrieben, die jeweils aus drei Z20 bestehen. Die Verbind-30-Fähnchen dienen wieder dazu, das Verhalten gut sichtbar zu machen.

Damit sich die Z20 nicht ins Gehege kommen, sind die Achsen – in Abb. 14 von rechts nach links gesehen – in folgenden

Bauteilen gelagert, die jeweils auf den Bausteinen 30 sitzen:

1. Winkelstein 30°, nach „außen“ geneigt,
2. Baustein 15 mit Bohrung,
3. Winkelstein 30°, nach „außen“ geneigt und schließlich
4. Winkelstein 60° mit dem Achslager nach „außen“ zeigend.

Wenn wir an der Kurbel drehen, kommt nun etwas ganz Verrücktes heraus:

- Die *Kurbel* dreht sich *gleichmäßig*.
- Die *erste Fahne* (Verbinder 30) vollführt ihre Umdrehung – wie in den vorherigen Modellen – in etwa nur *einer Hälfte* der Kurbelumdrehung; sie bleibt etwa *die Hälfte der Zeit stehen*.
- Die *zweite Fahne* vollführt ihre Umdrehung in etwa nur in *einem Viertel* der Kurbelumdrehung; sie bleibt *3/4 der Zeit stehen*.
- Die *dritte Fahne* vollführt ihre Umdrehung in etwa nur in *einem Achtel* der Kurbelumdrehung; sie bleibt *7/8 der Zeit stehen*.

Damit können wir also noch längere „Standzeiten“ beim Abtrieb erhalten, der

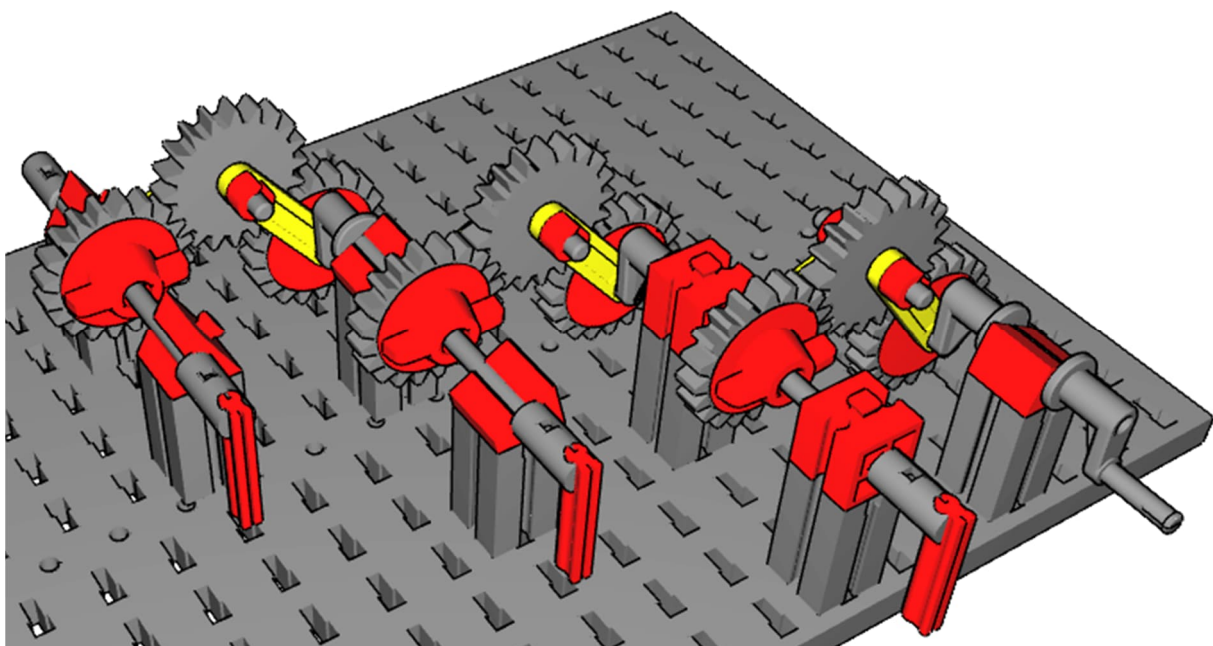


Abb. 14: Dreifach kaskadiertes zyklisch ungleichförmig übersetzendes Getriebe

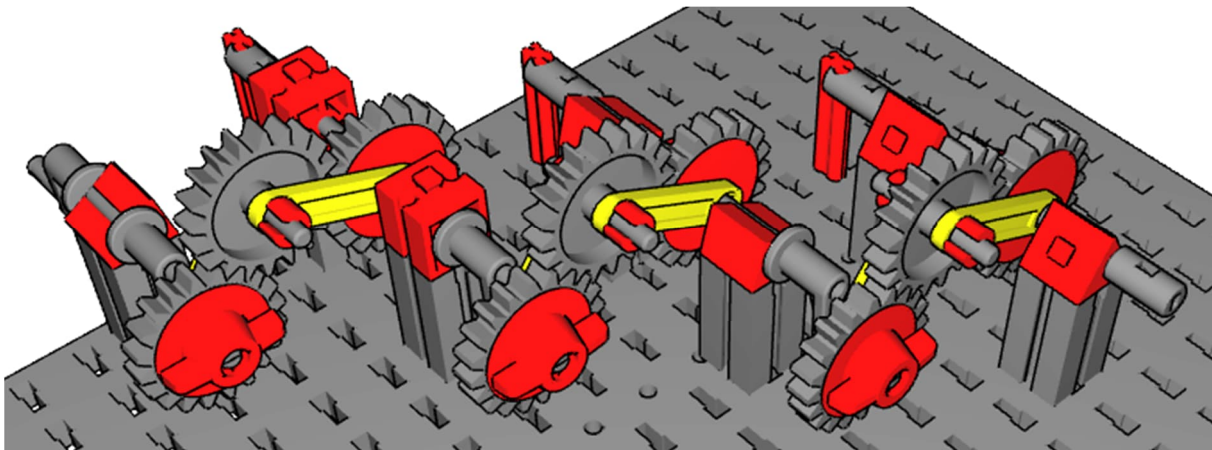


Abb. 15: Die andere Seite des Kaskadengetriebes

sich dafür „kurz vor Torschluss“ sehr schnell bewegt, damit er rechtzeitig zur vollendeten Umdrehung der Antriebskurbel ebenfalls „seine“ Umdrehung vollführt hat. Immer noch haben wir durch lauter Z20 insgesamt ein Übersetzungsverhältnis von 1:1 zwischen der Antriebskurbel und dem letzten Fähnchen – nur eben nicht gleichförmig!

Quellen

- [1] Deutsches Patent- und Markenamt: *Offenlegungsschrift: Getriebe mit zyklisch variablem Übersetzungsverhältnis*. Patentanmeldung von Herrn Klopmeier, auf [Google](#), 2007.
- [2] Klopmeier Ingenieurdienstleistungen: *Zyklisch variables Getriebe zum Antrieb von Tiefenpumpen*. Auf der [ft Community](#).
- [3] Werner Dohmen: *Erfinder baut mit Fischertechnik*. In: *Westdeutsche Zeitung* vom 14.11.2006, auf der [ft Community](#).
- [4] Werner Dohmen: *Erfindung: Willicher baut für Scheichs*. [Westdeutsche Zeitung](#), 2007.
- [5] Dirk Fox: *Wie man mit fischertechnik zum Patent kommt (2)*. [der fischertechnik-blog](#), 2015.
- [6] Wilhelm Klopmeier: *Zyklisch variable Getriebe*. Bilder von der fischertechnik-Convention 2011 in Erbes-Büdesheim auf der [ft Community](#), 2011.
- [7] Peter Damen: *Zyklisch variables Getriebe*. Auf der [ft Community](#), 2007.
- [8] Stefan Falk: *Klopmeier-Getriebe*. Auf der [ft Community](#), 2008.
- [9] Stefan Falk: *Drei zyklisch variable Getriebe nach Klopmeier*. Auf der [ft Community](#), 2017.
- [10] Stefan Falk: *Drei zyklisch variable Getriebe nach Klopmeier*. Auf [YouTube](#), 2017.
- [11] Stefan Falk: *Einfach zu bauende zyklisch variable Getriebe*. Auf [YouTube](#), 2022.
- [12] CyberForum e.V.: *Mint-Workshops*. Auf [mint-feriencamp.de](#), 2022.
- [13] Stefan Falk: *Zyklisch ungleichförmig übersetzende Getriebe*. [Workshop-Präsentation mit Bauanleitung und fischertechnik-Designer-Dateien](#) zu den Modellen dieses Beitrags, 2022.

Hydraulik

Schlauchpumpe

Konrad Fox, Dirk Fox

Wasser spielt im fischertechnik-Universum eine Nebenrolle – vielleicht, weil ein Spielzeughersteller nicht gern für geflutete Kinderzimmer verantwortlich sein möchte. Immerhin: Schlauchpumpen zählen seit 1973 zur Modellsammlung [1]. Dank neuer Bauteile sind heute zuverlässigere und leistungsfähigere Varianten möglich als vor bald 50 Jahren.

Geschichte

Eine der großen technischen Herausforderungen an die Infrastruktur wachsender menschlicher Siedlungen war die Bereitstellung von Trinkwasser – für uns heute eine Selbstverständlichkeit. Staudämme, Kanäle und Aquädukte zählen daher zu den frühen großen Ingenieurleistungen des Menschen [2].

Doch die Kanalisierung des Wasserflusses hilft nicht in jedem Fall: Wo kein Fluss ist, muss Wasser entweder über lange Gefällstrecken transportiert oder auf Grundwasser zurückgegriffen werden. Dafür sind in der Regel Wasserhebetekniken erforderlich.

Kleine Wassermengen wurden mit Eimern aus Brunnen geschöpft, aber für die Versorgung von Siedlungen benötigte man Techniken mit größerem Durchsatz. Für kleinere Hübe wurde die um 250 v. Chr. von *Archimedes von Syrakus* (287-212 v. Chr.) beschriebene und nach ihm benannte Schraube verwendet, von *Marcus Vitruvius Pollio* (kurz ‚Vitruv‘, ca. 80-15 v. Chr.) in seinen 10 Büchern über Architektur auch ‚Wasserschnecke‘ genannt. Sie wurde noch fast 2.000 Jahre später in Fluttermühlen zum Entwässern niederländischer Polder eingesetzt (Abb. 1).

Größere Hubstrecken ließen sich mit Eimerketten bewältigen. *Philon von Byzanz* (3.-2. Jhd. v. Chr.) beschrieb sie bereits um

230 v. Chr. Von den Römern wurden sie in Städten zur Versorgung der öffentlichen Bäder eingesetzt, solange diese noch nicht via Aquädukt an die römische Wasserversorgung angebunden waren. In Pompeii beispielsweise erhielten die Forums- und die Stabianer Thermen so Grundwasser aus 25 m Tiefe [3].

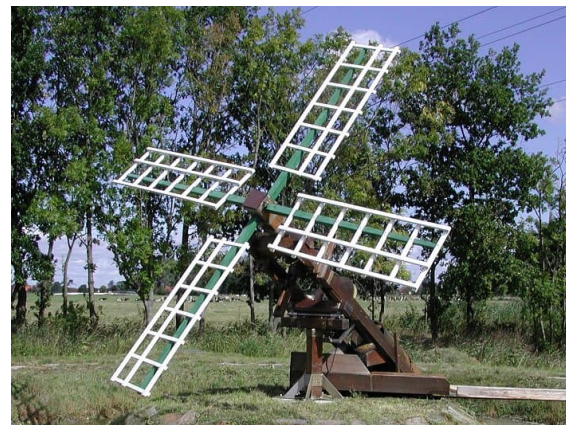


Abb. 1: Fluttermühle bei der Moorseeer Mühle in Nordenham-Moorsee (Foto: Wilfried Wittkowsky, 2003, [CC-BY-SA 3.0](https://creativecommons.org/licenses/by-sa/3.0/))

Eine alternative Technik, um größere Hübe zu bewältigen, wurde ebenfalls von Vitruv beschrieben: Schöpfträder – meist von Flüssen angetriebene große Speichenräder mit bis zu 5 m Durchmesser –, an denen seitlich angebrachte Gefäße das geschöpfte Wasser etwa auf der Höhe von $\frac{3}{4}$ des Raddurchmessers in eine Rinne gossen. Sie waren im Mittelalter vor allem in der arabischen Welt verbreitet und wurden ‚Noria‘ genannt (Abb. 2).



Abb. 2: Norias in Hama, Syrien
(Foto: Herbert Frank, [CC-BY-2.0](#))

Auch Druckkolbenpumpen waren Vitruv und Heron von Alexandria (zw. 300 v. Chr. und 200 n. Chr.) bereits bekannt; sie wurden aus Bronze gefertigt. Scheibenventile verhinderten den Rückfluss.

Schlauchpumpe

Die Schlauch- oder Rollerpumpe wurde hingegen erst 1924 von dem amerikanischen Herzchirurgen *Claude Schaeffer Beck* (1894-1971) für den Einsatz in Herz-Lungen-Maschinen entwickelt. Eine Schlauchpumpe hat die besondere Eigenschaft, dass die Durchflussgeschwindigkeit konstant ist und leicht reguliert werden kann. Sie ist zudem ein geschlossenes System, benötigt keine Ventile und ist wartungsarm.

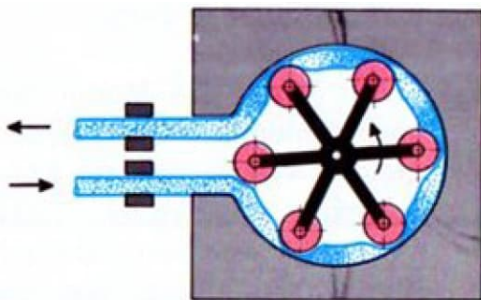


Abb. 3: Funktionsprinzip einer Schlauch- oder Rollerpumpe [3]

Das Funktionsprinzip zeigt Abb. 3: Mehrere Rollen oder Gleitschuhe drücken auf einen kreisförmig angeordneten Schlauch und erzeugen so einen Unterdruck, der die Flüssigkeit ansaugt und „portionsweise“ am anderen Ende wieder abgibt.



Abb. 4: Kommerzielle Schlauchpumpe

In der fischertechnik-Welt tauchte eine Schlauchpumpe erstmals im Jahr 1973 auf – im Clubheft 2/1973 wurde eine solche Konstruktion mit sechs V-Rädern 23x10 ([36581](#)) ausführlich beschrieben (Abb. 5) [1].

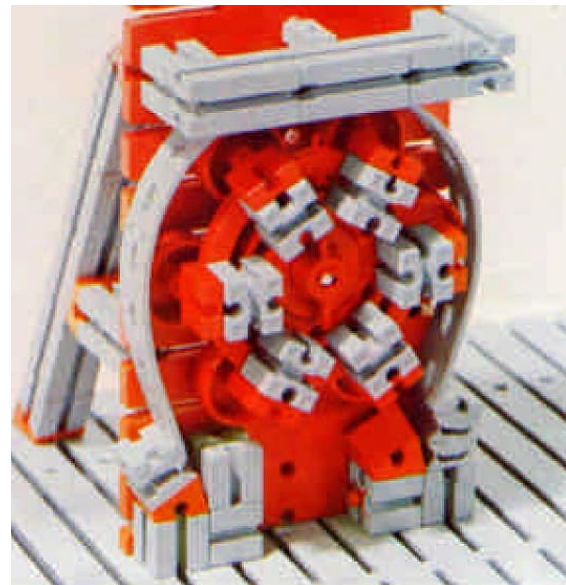


Abb. 5: Schlauchpumpe aus fischertechnik-Clubheft 2/1973 [1].

Wenig später wurde das Modell im letzten Band 4-5 der hobby-Reihe als Beispiel für eine Regelungsaufgabe aufgegriffen (Abb. 6) [4]. Die Konstruktion war – bis auf eine abweichende Befestigung der Schlauchenden – identisch mit der aus dem Clubheft 2/1973 in Abb. 5 [1].



Abb. 6: Schlauchpumpe in Band 4-5 der fischertechnik-hobby-Reihe [4], S. 45

Für seine Tropfen-Fotografie konstruierte Andreas Gail 2016 eine Variante dieser Pumpe, in der er die V-Räder durch Klemmringe Z36 ([31020](#)) ersetzte. Er beschrieb seine Pumpe ausführlich in Ausgabe 1/2016 der ft:pedia [5] sowie in seinem 2018 erschienenen Buch [6]. Dazu veröffentlichte er ein Video der Pumpe im Betrieb [7].

Verbesserungen

Schlauch

Im fischertechnik-Modell aus hobby 4 Band 5 wird ein Silikonschlauch mit 5 mm Durchmesser und einer Wandstärke von 0,5 mm empfohlen [4]. Inzwischen gibt es einen transparenten, weichen Silikonschlauch im fischertechnik-Teileuniversum ([131051](#)), der 2008 mit dem Baukasten „Basic Fire Commander“ ([500877](#)) als Feuerwehrschauch auf den Markt kam. Ein (bis auf die Länge) identischer Schlauch ([156043](#)) findet sich seit 2015 im Baukasten „Profi Dynamic M“; dort dient er als Aufhängung für die Klangrohre ([155957](#), [155958](#), [155959](#)). Dieser Schlauch eignet sich ausgezeichnet für eine fischertechnik-Schlauchpumpe, denn er kann direkt auf die Pneumatik-T-Stücke ([31642](#)) aufgesteckt werden.

Quetschungen

Der Durchsatz der Pumpe hängt ab vom Querschnitt des verwendeten Schlauchs, der Quetsch-Wirkung und der Geschwindigkeit der Rollen. Je leichter sich der Schlauch quetschen lässt, desto kleiner der Widerstand, den die Rollen (und der Motor)

überwinden müssen – damit steigen mit abnehmender Wandstärke des Schlauchs die Umdrehungsgeschwindigkeit der Achse und die Pumpleistung. Auch die Zahl der gleichzeitigen Quetschungen (und damit die Anzahl der Rollen) hat Einfluss auf die Geschwindigkeit: je weniger Quetschungen, desto kleiner ist der zu überwindende Widerstand. Also sollten so wenige Quetschungen wie möglich erfolgen. Damit kein Wasser zurücklaufen kann, benötigt man mindestens zwei gleichzeitige Quetschungen [1] (Abb. 7).

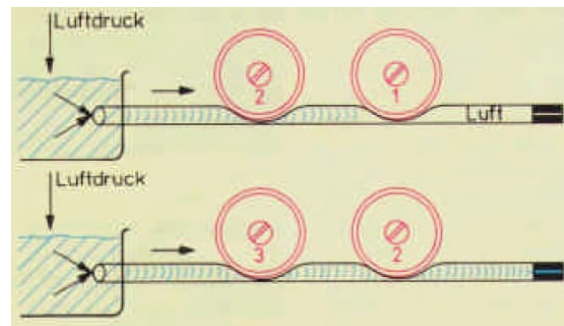


Abb. 7: Darstellung des Funktionsprinzips [1]

Werden die Rollen kreisförmig angeordnet, sind mindestens drei Rollen erforderlich, damit der Schlauch zu jeder Zeit an zwei Stellen gequetscht ist (siehe Abb. 4).

Mit kürzeren Schlauchabschnitten zwischen zwei Quetschungen ist die Pumpleistung gleichmäßiger. Führt man nun den Schlauch – anders als in den oben zitierten Konstruktionen – nicht in einer Schleife um alle, sondern nur über zwei von sechs Rollen (also ein Kreissegment), sind wenige kurze Quetschungen möglich. Damit nähert sich die Konstruktion der Beschreibung des Funktionsprinzips in Abb. 7.

Konstruktiv habe ich das erreicht, indem ich den Flachträger durch eine gebogene Statik-Strebe ersetzt habe (Abb. 8). Den nötigen flexiblen Andruck erzeugen zwei Feder-Gelenksteine 45 ([31308](#)).

Die neuen Statik-Streben haben auf beiden Seiten einen hochstehenden Rand. Dadurch ist die Quetsch-Wirkung zu gering und die Pumpe arbeitet nur sehr eingeschränkt.

Deswegen habe ich eine alte graue 90er Strebe verbaut, die auf der Rückseite keine Vertiefung hat und so für eine optimale Quetsch-Wirkung sorgt.

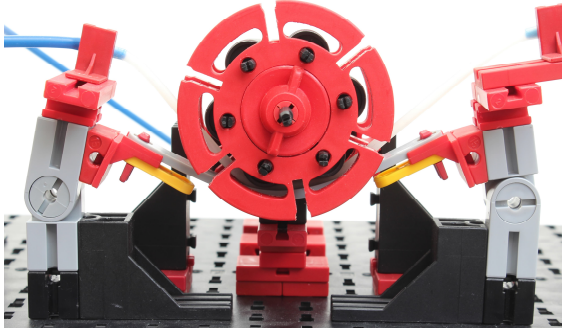


Abb. 8: Zwei Quetschungen, sechs Rollen

Allerdings haben die grauen Streben auch nur auf einer Seite eine Einkerbung für die S-Riegel, daher passt kein S-Riegel 4 ([36323](#)) durch die Strebe und die S-Adapterlasche ([31575](#)). Um dieses Problem zu lösen habe ich eine 15er Strebe ergänzt und alle drei Bauteile mit einem S-Riegel 6 ([36324](#)) fixiert.

Rollen

Je weiter die Rollen von der Antriebsachse entfernt sind, desto größer ist das für die Quetschung des Schlauchs erforderliche Drehmoment. Daher steigt der Pumpendurchsatz, wenn die Rollen näher an der Achse sitzen als in den Modellen in Abb. 5 und 6.

Für die Rollen-Achsen bieten sich die sechs Bohrungen der Drehscheibe 60 ([31019](#)) an (Abb. 8). Allerdings sind die V-Räder 23x10 dafür zu groß; sie sind auch zu breit für die Statik-Strebe. Deshalb habe ich sie durch sechs Paare von V-Rädern 14 ([36573](#)) ersetzt. Zwar sind zwei V-Räder 14 immer noch etwas breiter als eine Statik-Strebe, die Wölbung der Räder verhindert aber ein seitliches „Ausweichen“ des Schlauchs (Abb. 9).

Bei dieser Konstruktion halbieren sich der Abstand der Antriebsachse zum Schlauch,

und damit auch das Drehmoment. Außerdem halbiert sich die Länge der Schlauchabschnitte zwischen zwei Quetschungen.

Dabei ist zu beachten, dass die Drehscheiben 60 mit ausreichend Abstand auf der Achse befestigt werden, damit die V-Räder 14 frei drehen können und so über den Schlauch rollen (nicht gleiten). Das senkt den Verschleiß am Schlauch und den Widerstand.

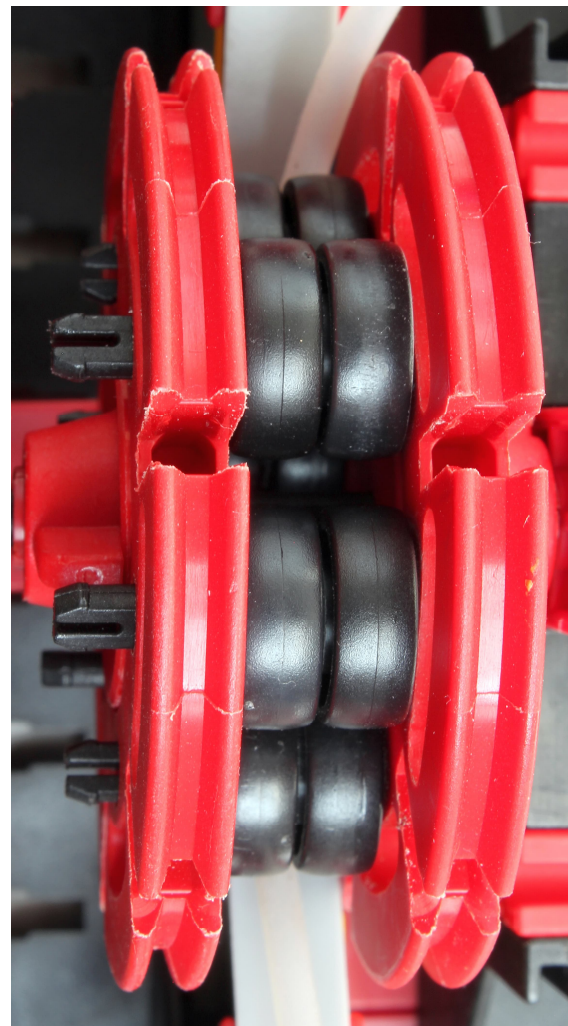


Abb. 9: Detailsansicht der Rollen

Wenn die Pumpe läuft, kann man die Position des Schlauches mit Hilfe der verschiebbaren T-Stücke optimieren.

Antrieb

Zur Optimierung des Pumpendurchsatzes habe ich auch mit Getrieben experimentiert. Meine Tests mit verschiedenen Antrieben

haben ergeben, dass ein Direktantrieb mit einem starken Motor das beste Pumpergebnis erzielt.

Fazit

Bei meinen Messungen erreichte die Pumpe eine Fördergeschwindigkeit von knapp 90 ml/min. Zum Füllen einer Badewanne ist sie also weniger geeignet – dafür benötigt sie über 30 Stunden... Befestigt man jedoch eine Pneumatik-Düse ([31632](#)) am Ende des Schlauchs, sorgt sie für einen ordentlichen Strahl – perfekt für kleine Wasserabenteuer ohne Überdruckpumpen.

Quellen

- [1] fischertechnik: Neues vom fischertechnik-Club. *Clubheft 1973-2*, S. 9-13. Fischer-Werke, 1973.
- [2] Walter Kaiser, Wolfgang König (Hrsg.): *Geschichte des Ingenieurs*. Carl Hanser Verlag, München, 2006.
- [3] Brigitte Cech: *Technik in der Antike*. Theiss Verlag, 2. Auflage 2011.
- [4] fischertechnik: *Niveauregelung bei bewegten Flüssigkeiten*. [hobby 4 Band 5](#), Fischer Werke, 1975, S. 42-45.
- [5] Andreas Gail: *Schlauchquetschpumpe*. [ft:pedia 1/2016](#), S. 40-41.
- [6] Andreas Gail: *fischertechnik-Modelle für Maker*. dpunkt-Verlag, 2018.
- [7] Andreas Gail: [Schlauchquetschpumpe](#) (youtubeVideo).

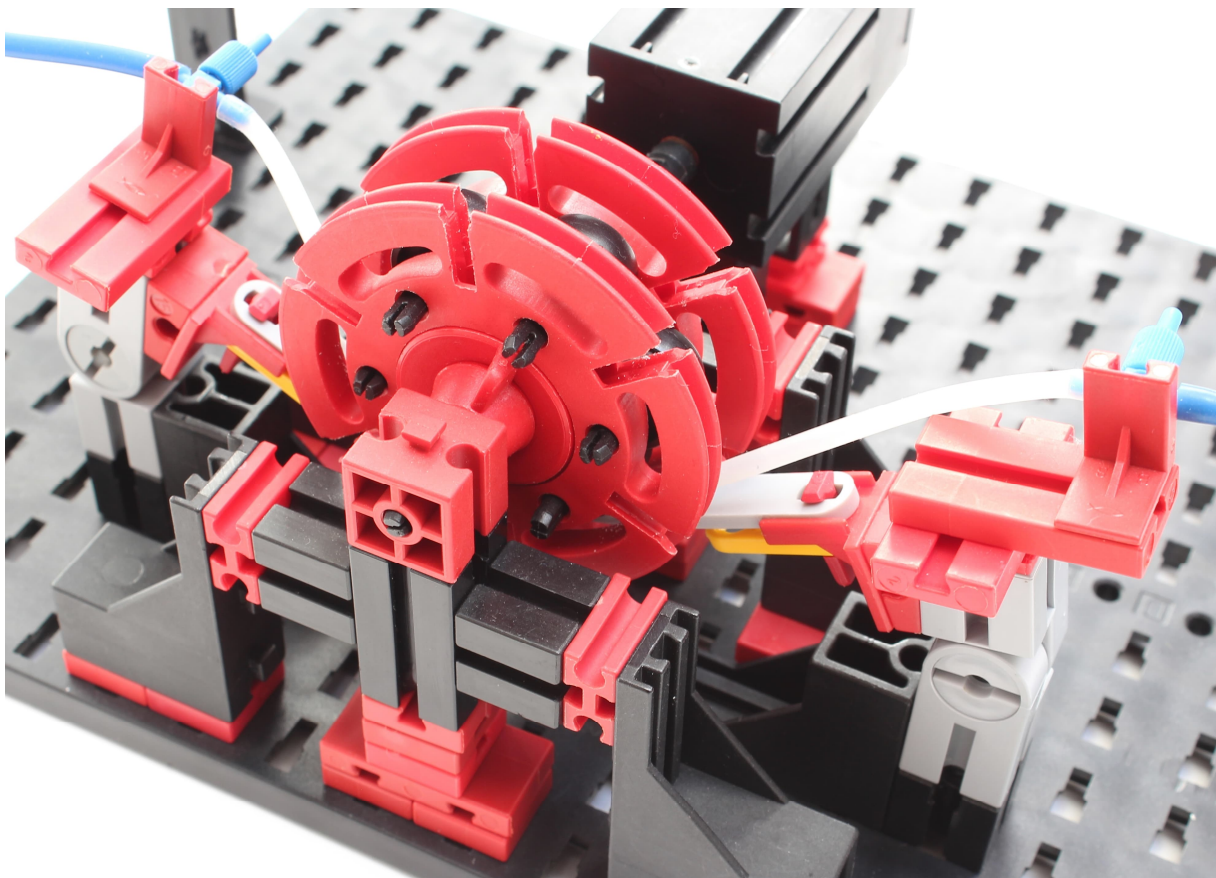


Abb. 10: Gesamtansicht der Schlauchpumpe

Mechanik

Logikgatter – elektronisch und mechanisch

Arnoud van Delden

Die logischen UND- und ODER-Gatter sind wahre „Pioniere“ der digitalen Logik: Schließlich bildet die „Hardware-Programmierung“ mit Kabeln und Verbindungen immer noch die Grundlage aller Steuerungen, die heute zunehmend in Software verwirklicht werden können. Der Beitrag erläutert die logischen Funktionen und elektronischen Grundlagen der beiden fischertechnik-„Silberlinge“ mit logischen Gattern. Ihre Funktionalität ist mathematisch austauschbar, was mit mechanischen Äquivalenten der Gatter demonstriert wird.

Hintergrund

Logikgatter sind die Grundlage der digitalen Elektronik. Sie haben einen oder mehrere Eingänge und einen Ausgang, dessen logischer Signalpegel durch die Funktion des Gatters bestimmt wird. Die logischen Funktionen AND (UND), OR (ODER) und die Negations- bzw. NOT (NICHT) Funktion (das logische Gegenteil des Steuerungssignals) bilden die Grundfunktionen der sogenannten Booleschen Algebra. Jede erdenkliche komplexere Logikfunktion lässt sich aus einer Kombination mehrerer dieser drei Grundfunktionen zunächst einmal mechanisch, dann auch elektromechanisch und schließlich elektronisch (und vielleicht in naher Zukunft noch schneller, nämlich auf optischem Weg) realisieren.

Um diese Logikfunktionen mit Elektronik zu implementieren, wurden zunächst Relais und später Vakuumröhren verwendet. Als die Massenproduktion von Halbleitern möglich wurde, übernahmen Dioden die Funktion der Vakuumröhren und die Logik konnte mit der sogenannten „Diodenlogik“ (DL) elektronisch realisiert werden. In den frühen 1960er Jahren wurde diese Methodik zur „Diode-Transistor-Logik“ (DTL) verfeinert, bei der die Diodeneingänge durch eine Transistorschaltung elektronisch ge-

puffert wurden [1]. Das reduzierte die Störanfälligkeit, verstärkte das Ausgangssignal und machte es einfacher, das oft sehr nützliche invertierte Funktionsergebnis (die Negation oder ‚NOT‘-Funktion) am Ausgang zur Verfügung zu stellen.

Silberlinge mit negativer Logik

Die fischertechnik-Silberlinge wurden mit diskreten elektronischen Bauteilen in Diode-Transistor-Logik aufgebaut. Im Schaltplan fallen sofort die Dioden an den Eingängen auf (Abb. 2).

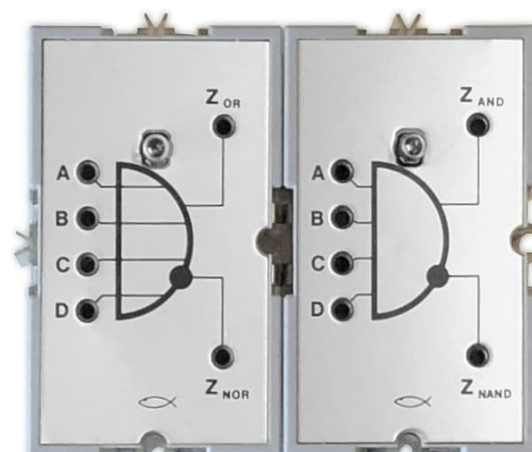


Abb. 1: h4ON- (30818) und h4AN-Silberling (30817)

Einfache Widerstände, Dioden und Transistoren wurden verwendet, um die logische

Ausgangsfunktion elektronisch aufzubauen. Für die Silberlinge wurde die sogenannte ‚negative Logik‘ gewählt, bei der ein logisches ‚1‘- (oder ‚aktives‘) Signal einem Signalpegel von 0 Volt entspricht und die Versorgungsspannung von 10 Volt einem logischen ‚0‘- (bzw. ‚inaktiv‘) Signal. Das Ausgangssignal eines Silberlings ist also ‚0‘, wenn am Ausgang (etwa) die Versorgungsspannung anliegt, und ‚1‘ (bzw. ‚aktiv‘), wenn diese Spannung sich 0 Volt nähert.

Die Wahl zwischen positiver oder negativer Logik ist im Prinzip beliebig [2]. Zwar wird heute hauptsächlich positive Logik verwendet, bei der die Signalpegel genau umgekehrt sind. Beim Anschließen von Sensoren an einen Mikrocontroller wird jedoch manchmal noch negative Logik verwendet, denn dies ermöglicht, nach dem Einschalten der Steuerung einen möglichen Drahtbruch in Sensorleitungen zu erkennen. Da dies aber bei den Silberlingen nicht von Bedeutung ist, konnte ich auf die Frage eines Clubmitglieds, warum sich fischertechnik damals bei den Silberlingen für negative Logik entschieden hat, nicht gleich antworten.

Als ich etwas später darüber nachdachte, vermutete ich zunächst, dass die möglicherweise reduzierte Stömpfindlichkeit ein Grund gewesen sein könnte. Damit sind Signale weniger empfindlich gegenüber Funkenrauschen von beispielsweise einem Motor, der an dieselbe Stromquelle angeschlossen ist.

Aber nachdem ich für diesen Beitrag recherchiert habe, glaube ich, den wahren Grund für die Wahl gefunden zu haben: Bei den Experimenten in den Hobby-Bänden ist die OR-Funktion oft sinnvoller als die AND-Funktion und kann bei Verwendung negativer Logik einfach durch direkte

Verdrahtung der Ausgänge erreicht werden, ohne dass zusätzliche Gatter-Silberlinge gekauft werden müssen. Die Ausgänge der Silberlinge verfügen zu diesem Zweck über einen sogenannten ‚Open Collector‘-Ausgang, der es ermöglicht, Ausgänge miteinander zu verbinden.³

Logikfunktion OR (ODER)

Betrachtet man den Originalschaltplan des h4ON-Elektronik-Bausteins (30818), so sieht man, dass die eigentliche Logikfunktion direkt am Eingang mit nur wenigen Bauteilen als verdrahtete Logikverbindung realisiert wird. Elektronisch ist dieses Prinzip als ‚Dioden-Logik‘ bekannt. So wurde eine sogenannte ‚verdrahtete Logikverbindung‘ geschaffen [4]. Wenn wir die nachträgliche Verstärkung mit Transistor(en) des Signals aus dem Diagramm weglassen, erhalten wir das Prinzipschaltbild in Abb. 2.

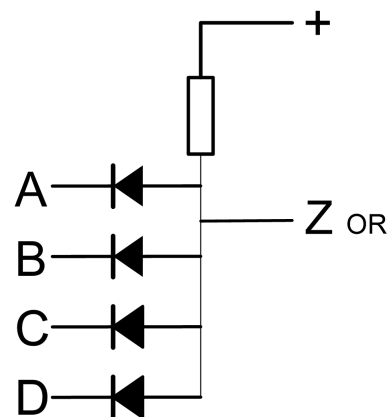


Abb. 2: OR-Diodenlogik

Der originale h4ON-Elektronik-Baustein hat vier Eingänge: A, B, C und D. Diese müssen nicht alle gleichzeitig angeschlossen und verwendet werden. Es ist sogar möglich, das Modul als Puffer oder Inverter (Ausgang Z_{NOR}) mit nur einem angeschlossenen Eingang zu verwenden. Die Anzahl der Eingänge wird im Wesentlichen durch die Anzahl der Dioden bestimmt, die am Knoten zusammenlaufen.

³ Siehe Erklärung ‚Verdrahtetes ODER‘ auf Seite 25 von hobby4, Band 4 [3].

In Hobby4, Band 4, Seite 33 wird erklärt, wie auf Wunsch noch mehr Eingänge nach diesem Prinzip erstellt werden können. Um die logischen Funktionen zu studieren, genügt es in diesem Beitrag, sich auf nur zwei Eingänge A und B zu beschränken [3].

In Formeln der digitalen Elektronik verwendet die OR-Funktion häufig das Symbol +, aber in der Booleschen Algebra ist das Symbol für diese Funktion \vee , auch als „Disjunktion“ bekannt. Dies ist auch das Symbol, das wir in den Hobby4-Bänden finden. Die Funktion der Schaltung kann als Formel geschrieben werden:

$$Z_{OR} = A \vee B$$

Ein Beispiel für die logische Disjunktion oder OR-Funktion im Alltag könnte sein:

A = Ich habe ein Fahrrad

B = Ich habe ein Auto

Z_{OR} = Ich kann zur Arbeit fahren

Die OR-Funktion beschreibt in diesem Fall „Ich kann zur Arbeit fahren, wenn ich ein Auto oder ein Fahrrad habe“. In Tabelle 1 sind alle Möglichkeiten und Ergebnisse in einer sogenannten Wahrheitstabelle zusammengefasst. Das entgegengesetzte oder komplementäre Ausgangssignal von „kann fahren“ könnten wir „nur laufen“ nennen. Die Tabelle mit den Ein- und Ausgabemöglichkeiten ist in Tab. 1 dargestellt.

Fahrrad	Wagen	kann fahren	nur laufen
nein	nein	nein	ja
nein	ja	ja	nein
ja	nein	ja	nein
ja	ja	ja	nein

Tab. 1: Wahrheitstabelle

Wenn wir diese Fallstudie auf die logischen Werte ,1‘ (aktiv oder wahr) und ,0‘ (inaktiv oder falsch) abstrahieren und sofort Spalten für alle komplementären Werte (mit einem Streifen darüber markiert) einschließen, ergibt Tab. 2 die Wahrheitstabelle dieser OR-Funktion.

A	B	\bar{A}	\bar{B}	Z_{OR}	$\overline{Z_{OR}}$
0	0	1	1	0	1
0	1	1	0	1	0
1	0	0	1	1	0
1	1	0	0	1	0

Tab. 2: Wahrheitstabelle der OR-Funktion

Diese nützliche Funktion lässt sich also elektrisch mit nur wenigen Dioden und einem Widerstand realisieren. Mal sehen, was genau elektronisch passiert.

Im Ruhezustand ist der Ausgang Z_{OR} dieser Silberling-Schaltung fast die Versorgungsspannung, also nach dem Prinzip der negativen Logik ein ,0‘ (inaktives) Signal. Das Signallämpchen im Modul ist aus. Wenn einer der Eingänge nach dem Prinzip der negativen Logik eine ,1‘ (aktiv) führt, bedeutet dies, dass dieser Eingang auf den Minuspol geschaltet ist. Dadurch kann durch die betreffende Diode ein Strom fließen, wodurch der Spannungspegel am Ausgang Z_{OR} sinkt. Dies zählt als logisch ,1‘ (aktiv) und das Signallämpchen leuchtet auf. Auf diese Weise lässt sich mit wenigen Dioden und einem sogenannten Pull-up-Widerstand eine logische OR-Funktion realisieren.

Die ODER-Funktion – mechanisch

Lange bevor dies elektronisch möglich wurde, wurden bereits mechanische Konstruktionen hergestellt, um bestimmte logische Funktionen auszuführen. Wie wir den Zusammenhang zwischen den Ein- und Ausgängen technisch realisieren, ist wie bei allen Logikfunktionen prinzipiell unabhängig von der letztendlich realisierten Logikfunktion. Das Innenleben ist trivial, solange alle möglichen Kombinationen der Eingangssignale gemäß der „programmierten“ Logikfunktion ein eindeutiges Ausgangssignal ergeben.

Eine rein mechanische OR-Funktionsverknüpfung lässt sich mit fischertechnik einfach konstruieren. Abb. 3 zeigt ein kompaktes Modell, das auf einer kleinen Bauplatte

aufgebaut ist. Bei einer solchen mechanischen Variante sind natürlich keine elektrischen Spannungspegel für die logischen ‚1‘- und ‚0‘-Werte definiert.

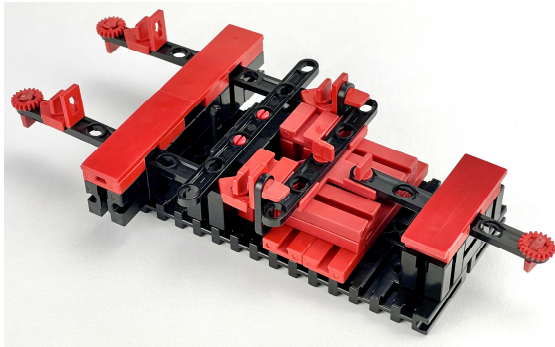


Abb. 3: Mechanisches ODER

Die „Positionen“ der Schieber auf der Eingangsseite (links) bestimmen die Position des Zeigers auf der Ausgangsseite (rechts) des Mechanismus.

Zur Verdeutlichung habe ich das Modell vergrößert und für die verschiebbaren Teile

graue Statik verwendet (Abb. 4). Aus den Schritten in Abb. 4 des größeren Modells ist ersichtlich, dass die Ausgabe (Z_{OR}) für Eingangswerte ‚0‘ oder ‚1‘ der Eingangssignale (A und B) die Wahrheitstabelle 2 der OR-Funktion genau befolgt.

Logikfunktion AND (UND)

Kommen wir nach diesem „mechanischen Ausflug“ zurück zur Elektronik im Silberling. Mit der für das OR-Gatter eingangs beschriebenen ‚Diodenlogik‘ lässt sich die logische Funktion der Schaltung relativ einfach verändern. Abb. 5 zeigt das Prinzipschaltbild.

Im Elektronik-Baustein h4AN ([30817](#)) wurde die Richtung der Dioden und die Funktion des Widerstands am Eingang von ‚pull-up‘ auf ‚pull-down‘ umgestellt. Dadurch wurde aus der oben beschriebenen OR-Funktion eine AND-Funktion, auch „Konjunktion“ genannt.

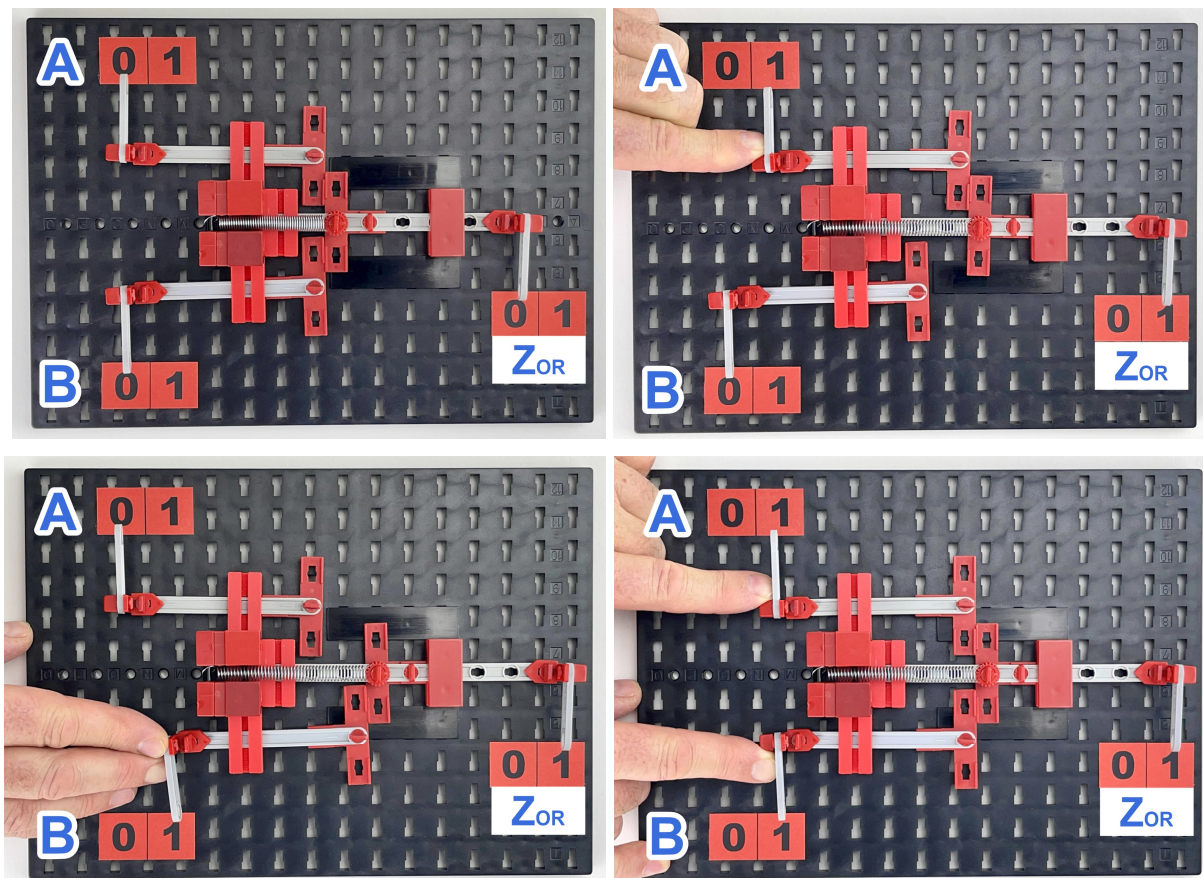


Abb. 4: Mechanisches ODER-Glied in Funktion

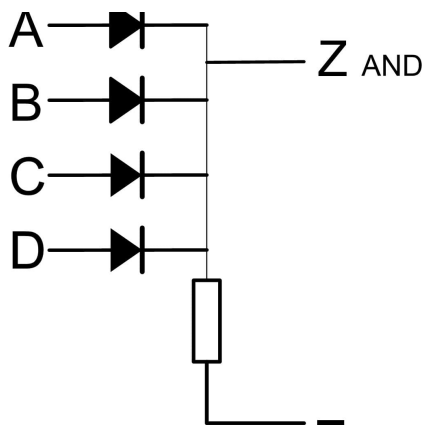


Abb. 5: Prinzipschema ‚wired-AND‘
(Negative Logik)

Formeln für digitale Elektronik verwenden oft einen (dicken) Punkt für die AND-Funktion, aber in der Booleschen Algebra ist \wedge das Symbol für die AND-Funktion. Die Funktion der Schaltung lässt sich damit als Formel schreiben:

$$Z_{AND} = A \wedge B$$

Mit der von Fischertechnik in den Silberlingen verwendeten negativen Logik können wir die AND-Funktion logisch so beschreiben: Der Ausgang darf nur dann ‚1‘ sein (einen Spannungspegel von 0 Volt haben), wenn mindestens einer der Eingänge nicht explizit ‚0‘ (10 Volt) ist. Da der Ausgang intern durch den Pull-Down-Widerstand auf ein niedriges Potential, also logisch ‚1‘ gezogen wird, werden nicht angeschlossene Eingänge bereits als ‚1‘ gesehen: Sie ziehen das Ausgangspotential nicht aktiv auf 10 Volt (die ‚0‘). Das ist schön, denn dann funktioniert die AND-Funktion auch mit ungenutzten ‚offenen‘ Eingängen korrekt (also auch, wenn wir nur zwei oder drei Eingänge brauchen). Es ist daher sogar möglich, nur einen Eingang zu verwenden und das Modul als Inverter am Ausgang Z_{NAND} ($\overline{Z_{AND}}$) einzusetzen, genau so, wie es beim OR-Silberling möglich ist.

Eine mit Diodenlogik erstellte AND-Funktion gibt am Ausgang eine logische ‚1‘ aus, wenn keiner der Eingänge verbunden ist. Ohne angeschlossenen Eingang leuchtet daher bereits das Signallämpchen des h4AN

Elektronik-Bausteins. Dieses Phänomen verschwindet von selbst, sobald mindestens ein Eingang mit dem Ausgang eines anderen Moduls verbunden wird. Wer nur Schalter für den Eingang bzw. die Eingänge verwendet, muss darauf achten, dass mindestens einer dieser Schalter, wenn nicht logisch ‚1‘, tatsächlich auf logisch ‚0‘ (10 Volt) verdrahtet ist.

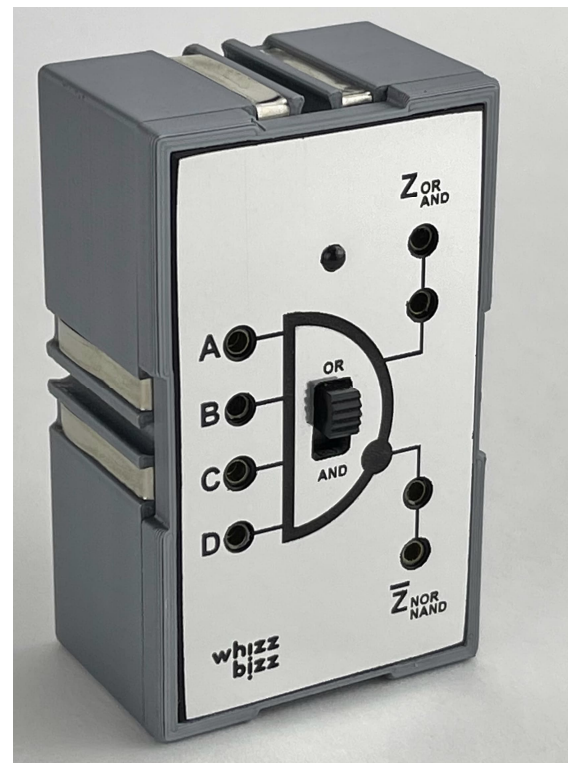


Abb. 6: Umschaltbarer selbstgebauter
OR/AND-Silberling

Wie bereits in der ft:pedia beschrieben [5], ermöglichten diese geringfügigen physikalischen Änderungen, den h4ON- und den h4AN-Elektronik-Baustein auf denselben Leiterplatten aufzubauen. Der Widerstand am Eingang wurde einfach eingelötet, entweder als Pull-up-Widerstand zur Versorgungsspannung oder als Pull-down-Widerstand zum Minuspol bzw. ‚Masse‘. Aufgrund dieser geringfügigen Unterschiede ist eine schaltbare Variante, wie sie in Abb. 6 dargestellt ist, beim Eigenbau dieser Module daher fast schon naheliegend [6].

Ein Beispiel für die AND-Funktion im Alltag könnte sein:

A = Ich habe ein Fahrrad

B = Ich habe ein Auto

Z_{AND} = Ich kann wählen, wie ich fahren möchte

Die AND-Funktion beschreibt in diesem Fall „Ich kann mir aussuchen, was ich fahre, weil ich ein Auto und ein Fahrrad habe“. Es ist möglich, alle Möglichkeiten und Ergebnisse in eine Wahrheitstabelle einzutragen. Das entgegengesetzte oder komplementäre Ausgangssignal von „kann wählen“ könnten wir „keine Wahl“ nennen. Tab. 3 zeigt die Wahrheitstabelle mit den Möglichkeiten für die Ein- und Ausgangssignale.

Fahrrad	Wagen	kann wählen	keine Wahl
nein	nein	nein	ja
nein	ja	nein	ja
ja	nein	nein	ja
ja	ja	ja	nein

Tab. 3: Wahrheitstabelle

Wenn wir die Fälle auf die logischen Werte ‚1‘ (aktiv oder wahr) und ‚0‘ (inaktiv oder falsch) abstrahieren und Spalten für alle Komplementärwerte einschließen, sieht die Wahrheitstabelle der AND-Funktion so aus wie in Tab. 4.

A	B	\bar{A}	\bar{B}	Z_{AND}	$\overline{Z_{AND}}$
0	0	1	1	0	1
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	1	0

Tab. 4: Wahrheitstabelle der AND-Funktion

Die UND-Funktion – mechanisch

Die AND-Funktion ist mechanisch etwas kniffliger als die OR-Funktion. Schließlich muss der Schieber am Ausgang intern die Bewegung eines aktivierten Eingangssignals „aufheben“ und sich erst dann in Bewegung setzen, wenn beide Schieber am Eingang auf ‚1‘ geschoben sind. Dies ist bei einer linearen Bewegung wie hier möglich, indem die Eingangssignale ein schwenkbare Joch bewegen. Eine Folge davon ist, dass die Bewegung des Ausgangssignals

kleiner ist als der Bewegungshub der einzelnen Eingangssignale.

Abb. 7 zeigt einen möglichen funktionalen Aufbau einer mechanischen AND-Funktion. Wer sich eingehender damit beschäftigt, dem fallen sicher noch andere Lösungen ein.

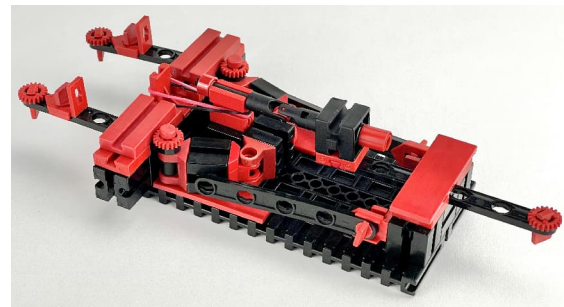


Abb. 7: Mechanisches AND

Zur Verdeutlichung siehe Abb. 8, wo die AND-Funktion auf einer größeren Bauplatte aufgebaut ist. Die Bewegung des Zeigers des Ausgangssignals wird mechanisch mit einem Hebel „verstärkt“.

Elektronisch könnte dies mit der Transistor-Verstärkungsstufe in der „Dioden-Transistor“-Logik der Silberlinge verglichen werden. Eine zweite Transistorstufe in den Silberlingen invertiert das Ausgangssignal (zurück), sodass beide Varianten zur Verfügung stehen. Später in diesem Artikel werden wir die mechanische Variante davon sehen. Schließlich kann ein Hebel auch mechanisch genutzt werden, um eine Bewegung (Signal) umzukehren.

Die De Morganschen Gesetze

Neben den OR- und AND-Funktionen spielt die Negation bzw. NOT-Funktion als logische Basisfunktion eine wichtige Rolle. Es stellt sich heraus, dass durch die Verwendung dieser NOT-Funktion die OR- und AND-Funktionen austauschbar werden. Das praktische Ergebnis dieser logischen Gesetzmäßigkeit wird auch in den fischertechnik-Hobby-Bänden erwähnt [3, 7].

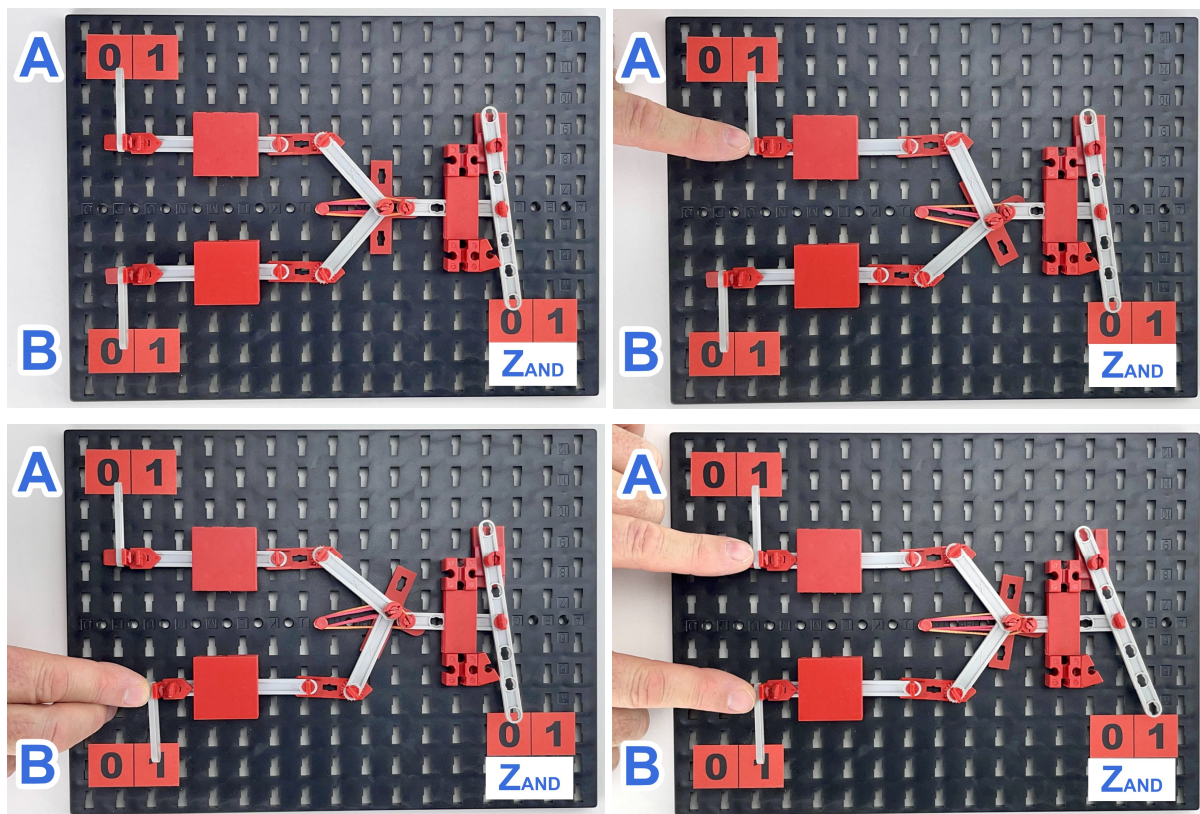


Abb. 8: Mechanisches UND-Glied in Funktion

Derjenige, der diese Gesetze bekannt machte, war der britische Mathematiker *Augustus De Morgan* (1806-1871) [8]. Die nach ihm „De Morgansche Gesetze“ genannten Regeln besagen, dass jede Konjunktion (AND-Funktion) durch eine Disjunktion (OR-Funktion) ausgedrückt werden kann (und umgekehrt). Sie werden häufig in mathematischen Beweisen und Software verwendet. Zusammen mit *George Boole* (1815-1864) gilt De Morgan als Begründer der formalen Logik.

Seine Gesetze lauten:

$$\overline{A \vee B} = \overline{A} \wedge \overline{B}$$

$$\overline{A \wedge B} = \overline{A} \vee \overline{B}$$

Die Formeln beschreiben, dass durch Anwendung der NOT-Funktion auf das einzelne Eingangssignal und das abschließende logisch verknüpfte Ausgangssignal die Basisfunktionen OR (\vee) und AND (\wedge) ineinander übergehen. Mit anderen Worten: Die OR-Funktion zweier zuvor invertierter

Eingangssignale entspricht dem invertierten Ausgangssignal einer AND-Funktion dieser Eingangssignale. Und umgekehrt: Werden die Eingangssignale der AND-Funktion vorher jeweils mit der NICHT-Funktion invertiert, verhält sich das invertierte Ausgangssignal nach außen wie eine OR-Funktion.

Die Gesetze von De Morgan lassen sich auch auf das obige praktische Beispiel mit Fahrrad und Auto anwenden. Mit der Funktion $X = A \vee B$ lässt sich X im Klartext im Prinzip so erklären: „Ich kann zur Arbeit fahren, wenn ich ein Auto ODER ein Fahrrad habe“. Genau dasselbe kann ausgedrückt werden als: „Ich kann nicht zur Arbeit fahren, wenn ich kein Auto UND kein Fahrrad habe“, wobei die resultierende Formel $\overline{X} = \overline{A} \wedge \overline{B}$ das ODER umschreibt als eine UND-Verknüpfung der negativen Bedingungen. Beides bedeutet dasselbe und stellt die gleiche Beziehung zwischen den „Eingangssignalen“ und der Schlussfolge-

rung, dem „Ausgangssignal“ her. Es werden nur zwei verschiedene Grundfunktionen verwendet.

Wenn ihr zurückblättert werdet ihr sehen, dass in Tab. 2 (der OR-Funktion) die Spalte $\overline{Z_{OR}}$ ganz rechts für die Eingangssignale \overline{A} und \overline{B} die UND-Funktion beschreibt. Die Umkehrung ist auch in Tab. 4 (der AND-Funktion) sichtbar. Dort zeigt die ganz rechte Spalte $\overline{Z_{AND}}$ für die Eingangssignale \overline{A} und \overline{B} eine logische OR-Funktion.

In der „Silberling“-Reihe, die fischertechnik in der zweiten Hälfte des letzten Jahrhunderts auf den Markt brachte, gibt es kein Modul, das ausschließlich die Aufgabe hatte, Eingangssignale zu invertieren.

Bei den in den Hobby4-Büchern besprochenen Modellen ist das kein Problem, denn das invertierte Signal jedes Ausgangssignals steht jederzeit für das nächste Logikgatter oder die Steuerung eines Relais zur Verfügung. Der Elektronik-Baustein G (36391) hat ebenfalls einen invertierten Ausgang (A2). Dadurch ist auch bei Sensoren mit „invertiertem“ Signalpegelverhalten (am Eingang E1) kein Inverter notwendig.

NOT-Funktion – mechanisch

Die NICHT-Funktion erweist sich als mechanisch relativ einfach zu realisieren. Es geht lediglich darum, die Bewegung des Ausgangssignals der Bewegung des Eingangssignals entgegengesetzt zu machen. Abb. 9 zeigt einige Beispiele für den Aufbau dieser Funktion auf einer kleinen Bauplatte.

Der mechanische Betrieb ist auf den statischen Fotos möglicherweise schwer zu erkennen. Weitere Details und ein [Video zur Funktionsweise](#) der verschiedenen Gatter findet ihr auf meiner [Projektseite](#), die ich während der Arbeit an den mechanischen Logik-Gattern erstellt habe [9, 10].

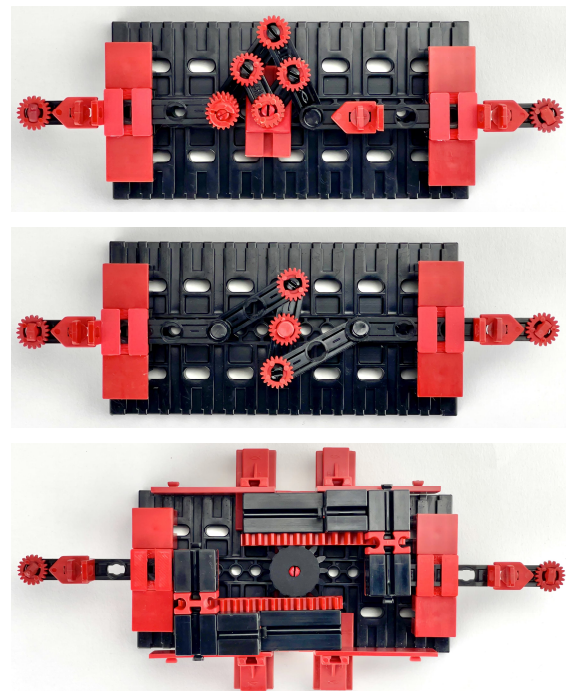


Abb. 9: Verschiedene Versionen einer mechanischen NOT-Funktion

Austauschbarkeit von OR und AND

Die Gesetze von De Morgan können auch mit einem mechanischen Modell visualisiert werden. Die Grundfunktion des Modells von Abb. 10 ist die mechanische OR-Funktion, die wir in Abb. 8 aufgebaut gesehen haben. Mit einfachen Hebeln an den Ein- und Ausgängen werden jedoch sowohl die Eingangssignale als auch der Ausgang invertiert. Und obwohl in der Mitte noch die OR-Funktion erkennbar ist, verhält sich dieses mechanische Gatter nach außen wie eine AND-Funktion.

Der Aufbau von Abb. 10 entspricht dem Diagramm 35.1 auf Seite 35 von hobby 4, Band 4 [3] mit dem Unterschied, dass wir jetzt einen Schritt weiter gehen und nicht Z_{OR} , sondern $\overline{Z_{OR}}$ (auch Z_{NOR} genannt) als Ausgang verwenden. Dadurch ändert sich die Logikausgangsfunktion von OR zu AND.

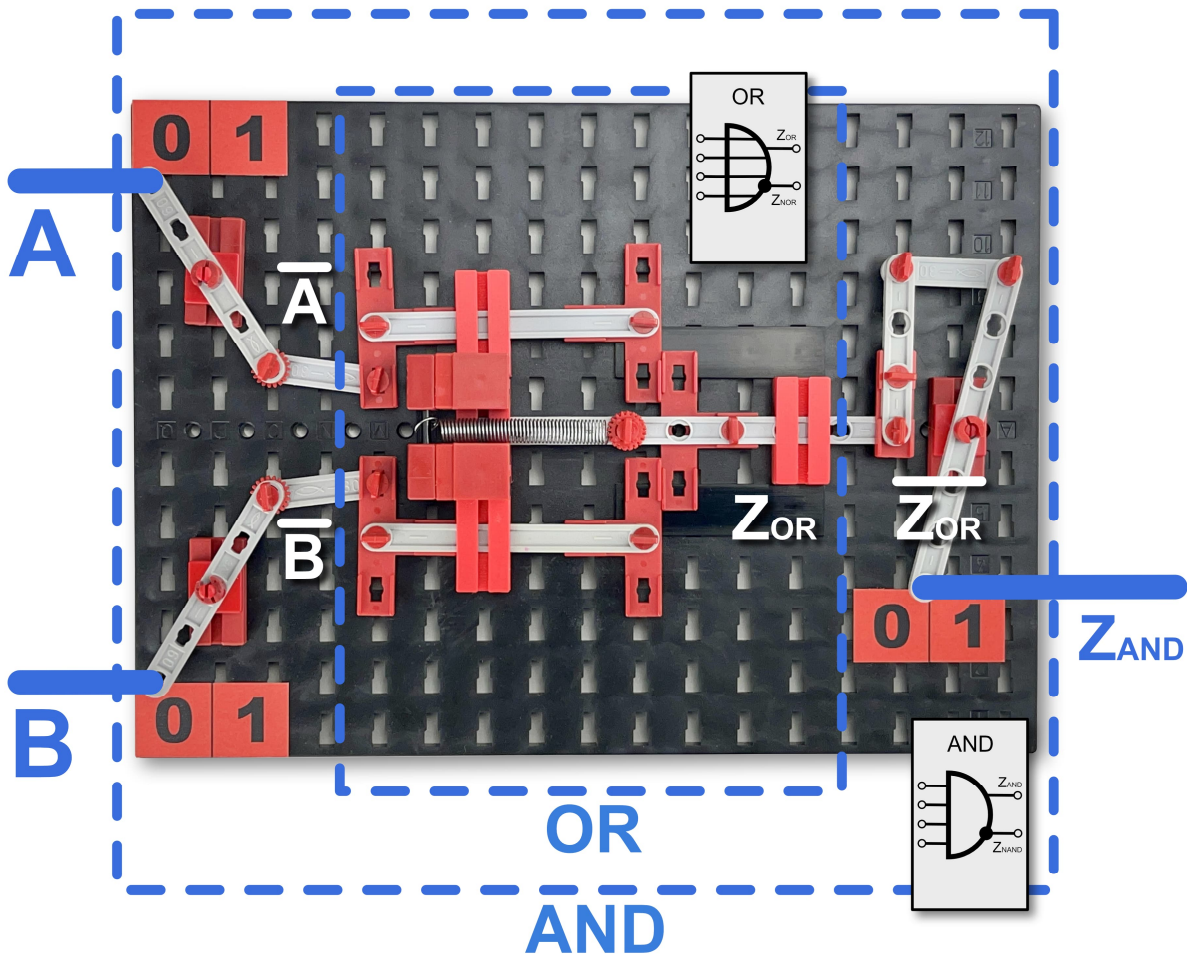


Abb. 10: Inverter an Ein- und Ausgängen machen aus einem OR-Gatter ein AND-Gatter

Wenn man die Ausgabefunktion von Abb. 11 in Wahrheitstabellen darstellt, erkennt man, dass die Ausgabe Z_{AND} der Spalte $\overline{Z_{OR}}$ aus Tab. 2 entspricht, unter Verwendung der invertierten Eingangssignale \overline{A} und \overline{B} , und damit auch vollständig dem Verhalten des AND-Gatters, dargestellt in Tab. 4.

Das logische OR kann sich also wie ein logisches AND verhalten und umgekehrt. Sobald wir sowohl die einzelnen Eingangssignale als auch das resultierende Ausgangssignal invertieren, gelten die De Morganschen Gesetze, auch dann, wenn wir statt negativer Logik (wie in den Silberlingen) positive Logik oder den invertierten Ausgang der bisherigen Silberlinge als Eingangssignal verwenden und den invertierten Ausgang als Ausgangssignal.

Faktisch ist das nichts anderes als das Vertauschen der vereinbarten Signalpegel ‚0‘ und ‚1‘.

Dies wird in Abb. 13 verdeutlicht. Wenn wir, wie in der obersten Zeile angedeutet, die Eingangssignale und das Ausgangssignal logisch invertieren, ändert sich die logische Funktion. Dies ist funktional genau dasselbe wie die Verwendung der invertierten Logik für die Eingangssignale und die Verwendung des komplementären Ausgangs Z_{NAND} ($=\overline{Z_{AND}}$) oder Z_{NOR} ($=\overline{Z_{OR}}$) des jeweiligen Logikbausteins als Ausgangssignal. Wer die klassischen Silberlinge mit Signalen in positiver Logik (wie den Ausgängen der TX- und TXT-Controller oder modernen DIY-Varianten der Silberlinge in CMOS-Technik) verwenden möchte, muss berücksichtigen, dass alle Signalpegel bereits invertiert sind.

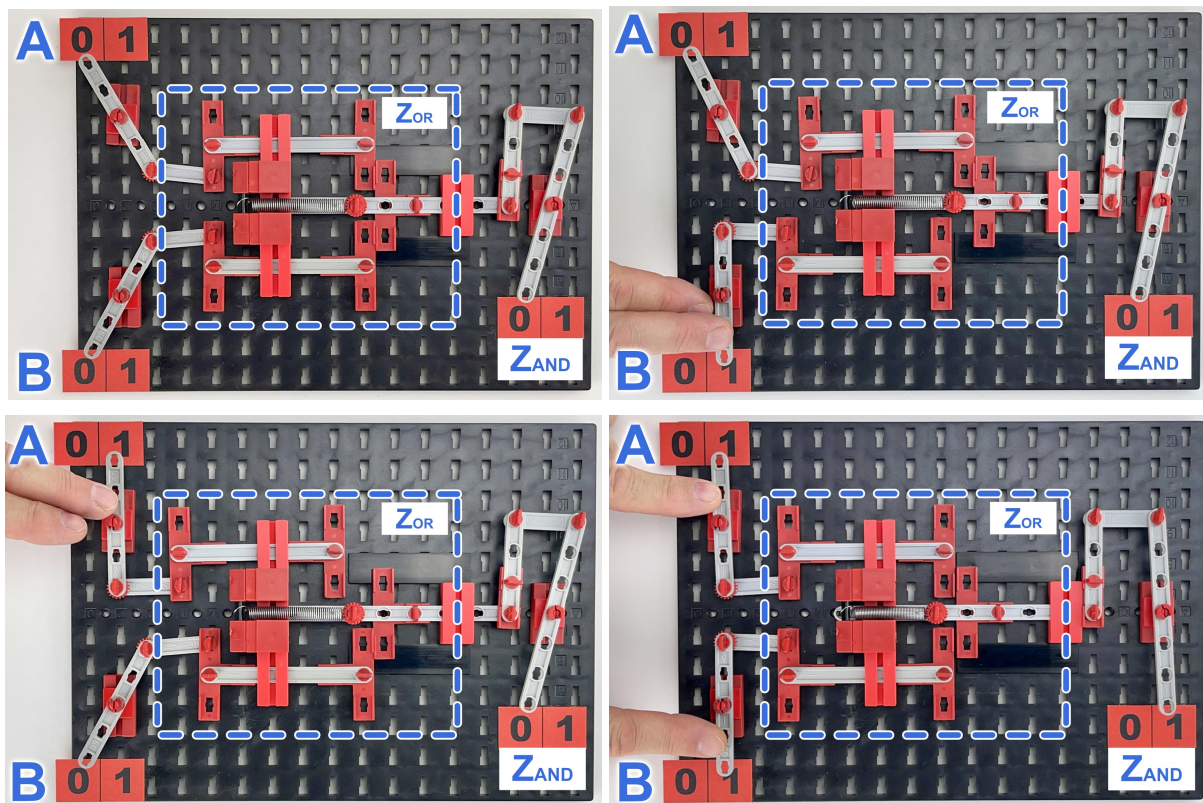


Abb. 11: De Morgansche Gesetze in Aktion: Das innere OR-Gatter verhält sich wie ein AND-Gatter

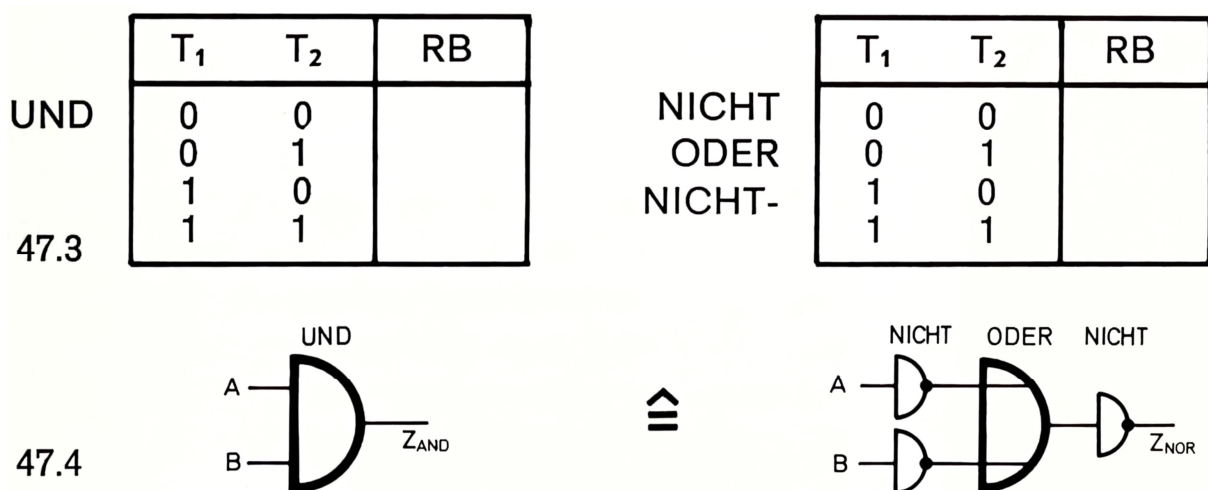


Abb. 12: In Hobby4 Band 4, Seite 47, wurde dieses Thema ebenfalls berührt [1]

Negative 'Silberlingen' Logik
'0'=10 volt - '1'=0 volt

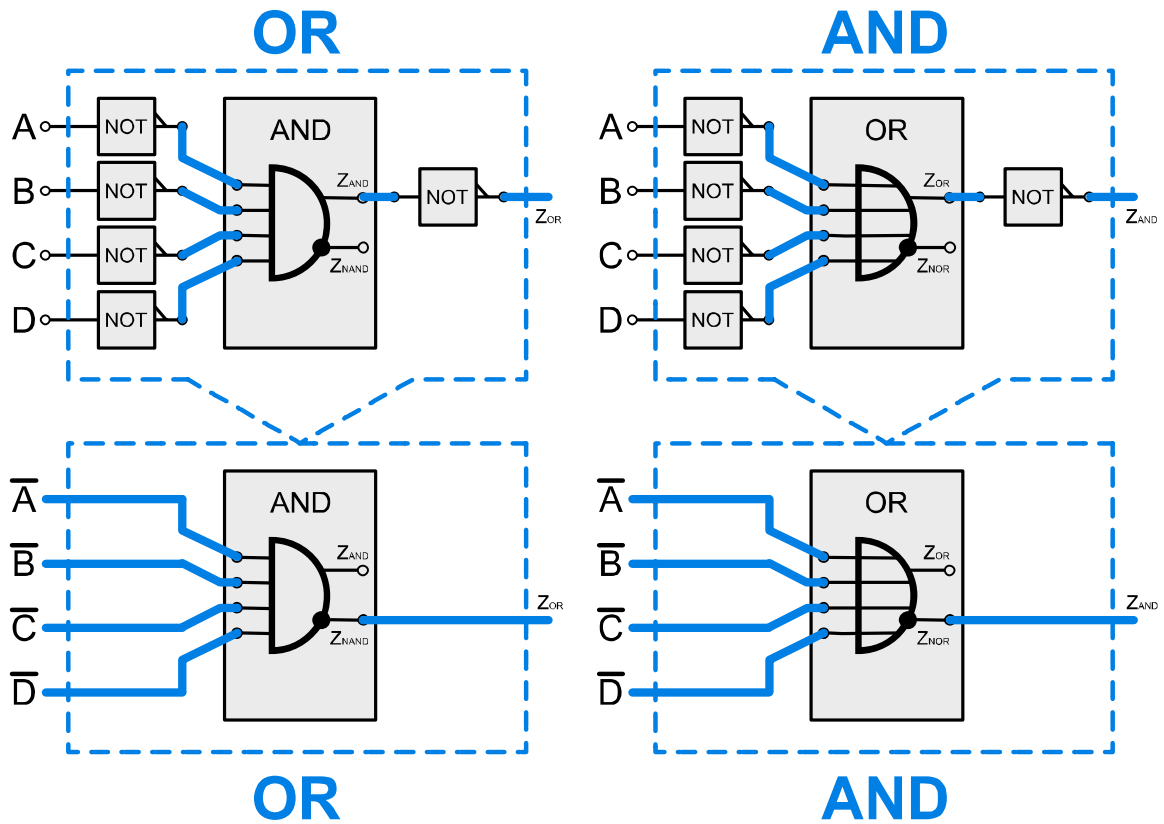


Abb. 13: Silberlinge als „logische Chamäleons“

Positive (CMOS) Logik
'0'=0 volt - '1'=10 volt

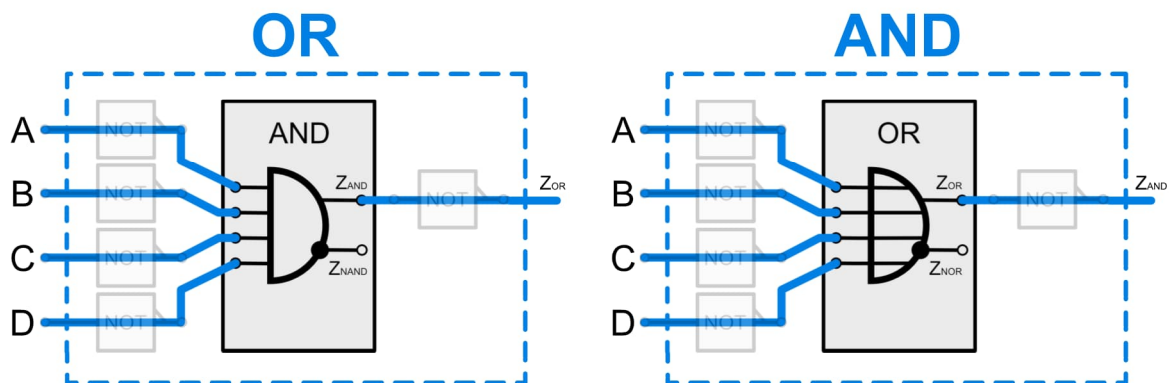


Abb. 14: Silberlinge mit positiver Logik verwendet

Damit kann ein Elektronik-Baustein h4AN direkt als OR- und ein Elektronik-Baustein h4ON als AND-Gatter verwendet werden (Abb. 14).

Zur besseren Nachvollziehbarkeit der aufgebauten Logikschaltung erscheint es ratsam, die verwendeten Silberlinge mit einem kleinen Hinweis zu versehen; das wird euch sicherlich helfen, mögliche Schaltprobleme zu lösen.

Fazit

Die mechanischen Modelle der Logikgatter geben Einblick in die Gesetze von De Morgan. Wenn ihr im Internet sucht, werdet ihr schnell auf Modelle von Lego und Knex stoßen. Seltsamerweise war Fischertechnik in diesem Bereich noch nicht oder kaum vertreten. Und das, obwohl sich unser beliebtestes Konstruktionsmaterial perfekt zum Experimentieren eignet, weil für jede logische Funktion viele verschiedene mechanische Lösungen möglich sind.

Beispielsweise können die Funktionen mit Zahnrädern, mit Kugeln oder auch mit Seilen und Gewichten konstruiert werden. Thomas Püttmann hat sich in der Vergangenheit bereits mit mechanischen Gattern beschäftigt, die [mit radialer Bewegung und Getrieben](#) funktionieren. In meinen mechanischen Experimenten habe ich hauptsächlich lineare Bewegungen untersucht, vielleicht wegen der visuellen Analogie zur elektronischen Funktionsweise der „Transistor-Transistor-Logik“, die ich von den späteren ICs kannte. Je mehr man über die Materie nachdenkt, desto mehr Parallelen entdeckt man zwischen der Elektronik und den mechanischen Modellen.

Die lineare, gleitende oder schwenkende Bewegung lässt sich mit Fischertechnik

natürlich vielfältig realisieren. Als Drehpunkt kann nicht nur eine Achse durch ein Loch (z. B. in einem Baustein) dienen. Drehpunkte können auch mit einem Kardan oder den verschiedenen Scharniersteinen realisiert werden.

Eine Öffnung, durch die eine Statik-Strebe gleiten kann, ist leicht herzustellen. Es war aber etwas schwierig, eine Lösung zu finden, die im 15x15 mm-Raster bleibt und bei der die Strebe sauber, ohne zu viel vertikales Spiel durch die Öffnung gleitet. Zu diesem Zweck habe ich zwei Steine entworfen (Abb. 15, 16): Der kleine entspricht der Größe eines Baustein 5 ([37237](#)), der breitere der einer Bauplatte 5x15x30 ([35049](#)).

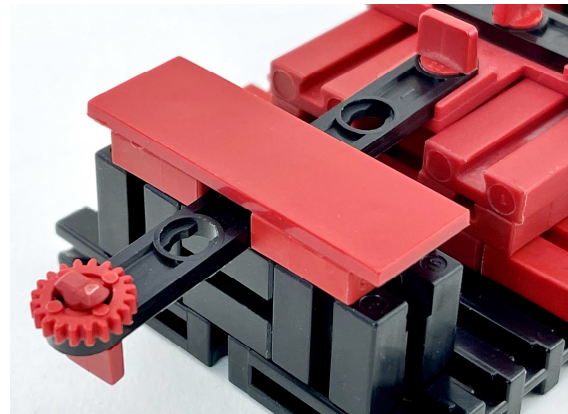


Abb. 15: Mögliche Führung der Statik-Strebe

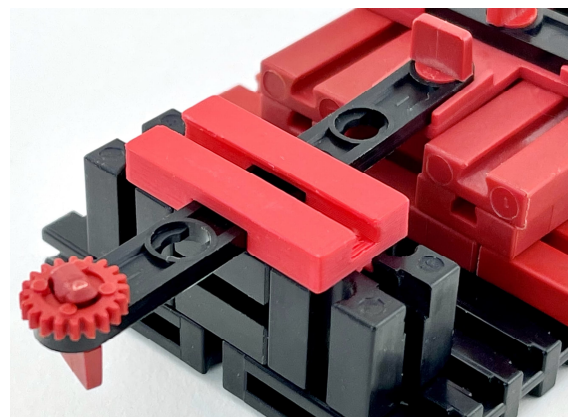


Abb. 16: Führung mit weniger Spiel dank 3D-gedrucktem Baustein⁴

⁴ 3D-Daten des Bausteins zum Download unter: <https://www.printables.com/model/317251-slider-gatebridge-stone-for-fischertechnik>

Wenn ihr denkt, dass sie nützlich sind, könnt ihr sie selbst in 3D drucken (gerne bin ich dabei behilflich).⁵

Die Bewegung des Ausgangs in diesen mechanischen Modellen wird vollständig durch die Bewegungen der Eingangssignale gebildet. Um jedoch komplexere Funktionen wie ein XOR-Gatter oder ein Flip-Flop aufzubauen, müssen mehrere grundlegende mechanische Funktionen miteinander verkettet werden. Aufgrund von Spiel und Reibung in den mechanischen Zapfen lassen sich die hier gezeigten Gatter jedoch nicht so einfach hintereinander schalten; die Kraft an den Ausgängen nimmt schnell ab.

Analog zu den hierfür verwendeten Lösungen in den elektronischen Varianten dieser Logikgatter könnte in einem nächsten Schritt nach Möglichkeiten gesucht werden, die Kräfte am Ausgang zu „verstärken“ oder mechanisch zu „entkoppeln“. Für's Erste hoffe ich, dass die besprochenen Grundfunktionen euer Interesse geweckt haben. Experimentiert damit, und wer weiß, vielleicht werden wir bald mehr mechanische Gatter und komplexere mechanische Logikfunktionen in der ft:pedia finden.

Quellen

- [1] Wikipedia: [Diode-Transistor-Logik](#).
- [2] Wikipedia: [Positive und negative Logik](#).
- [3] fischertechnik: *NICHT ODER oder ODER NICHT?* [hobby 4 Band 4](#), Fischer Werke, 1975, S. 34-35.
- [4] Wikipedia: [Wired Logic Connection](#).
- [5] Peter Krijnen: *Silberlinge: Original oder Nachbau (Teil 7)*. [ft:pedia 2/2022](#), S. 87–100.
- [6] Donaupeter: [AND-NAND / OR-NOR Baustein](#).
- [7] fischertechnik: *IC-Digital-Praktikum* (6396304), fischer-werke, 1977 (Seite 57).
- [8] Wikipedia: [Gesetze von De Morgan](#).
- [9] Arnoud van Delden: [Mechanical Logic Gates](#). YouTube, 2022.
- [10] Arnoud van Delden: [Mechanical Logic](#). Whizzbizz.com, 2022.

⁵ Die fischertechnik-bezogenen 3D-Objekte, die ich drucken kann, findet ihr [auf meiner Webseite](#).

Physik

Kettenrätsel

Rüdiger Riedel

Kettenfontänen und Kettenfünfecke – manchmal versucht man das Eine und findet das Andere.

Kettenfünfecke

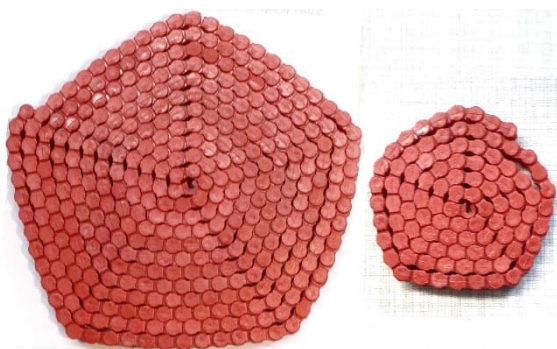


Abb. 1: Zwei Kettenwickel

Wickeln wir eine alte fischertechnik-Kette ([36263](#)) spiralförmig auf, ergibt sich immer (?) ein einigermaßen regelmäßiges Fünfeck. Aber warum? Liegt es an der fischertechnik-Kette? Erwartet habe ich Sechsecke; beim kleinen Wickel ist in der innersten Schicht eine Andeutung davon zu sehen.



Abb. 2: Linksgewickelt

Der Wickelsinn scheint unerheblich zu sein, vergleiche Abb. 1 rechts und Abb. 2.



Abb. 3: Neue ft-Kette

Die neue fischertechnik-Kette ([36248](#)) verhält sich ähnlich.

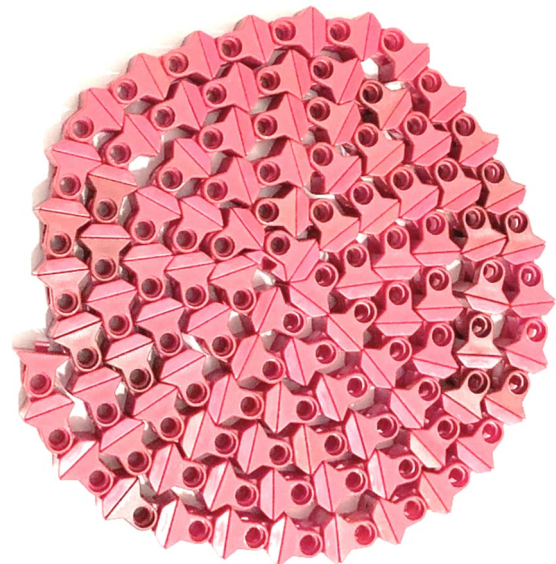


Abb. 4: Gelenkkette (Beginn: Nut in der Mitte)

Die Gestaltbildung in Abb. 4 ist noch nicht abgeschlossen; die Zahl der Glieder ([31426](#) und [31436](#)) ist zu gering. Könnte sich hier auch eine fünfstrahlige Symmetrie herausbilden?

Kettenfontäne

Im Internet bin ich auf das Phänomen der aufsteigenden Kette beim Rutschen aus einem Glas gestoßen [1, 2].

Ich startete den Versuch mit einer 9 m langen Kugeldette, wie sie den Stöpsel im Waschbecken hält. Sie wird in ein kleines Glas gefüllt und am Anfang mit einem Baustein beschwert, damit sie schneller auf Geschwindigkeit kommt. Das Ganze wird an den Rand des Tisches gestellt (Abb. 5).



Abb. 5: Kettenfontäne 1

Während die Kette herunter fällt, zieht sie sich selbst aus dem Glas. Aber statt am Glasrand herunterzurutschen, steigt sie ein Stück auf zu einem in der Luft schwebenden Bogen (Abb. 6).

So etwas hätte ich gerne mit fischertechnik-Bauteilen realisiert. Aber das ist eine Herausforderung. Eine Schwierigkeit besteht im flachen Aufbau der fischertechnik-Ketten. Der Ablauf muss also so geführt werden, dass keine Drehung erfolgt.



Abb. 6: Kettenfontäne 2

Der Behälter von Abb. 7 ist so gestaltet, dass die Kette mäanderförmig darin zu liegen kommt. Dann wird das Kettenende herausgenommen und fallengelassen [3].

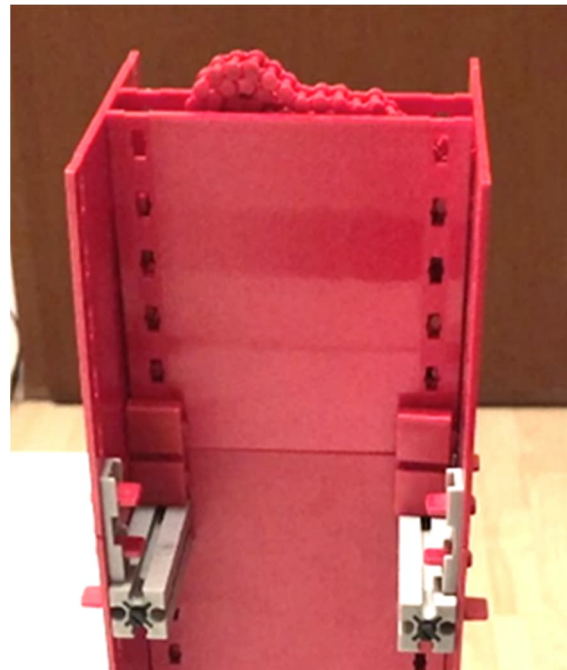


Abb. 7: Versuchsaufbau

Der Bogen, den die Kettenfontäne in Abb. 8 beschreibt, ist nicht sehr hoch, weil die Kettenglieder sehr leicht sind. Bei der schwereren Kette in Abb. 5, 6 bewirkt das Beschleunigen der Glieder nach oben, dass durch deren Trägheit die Fontäne deutlich sichtbar wird – die Glieder wehren sich sozusagen dagegen, einfach flach aus dem

Glas gezogen zu werden, weil sie ja Impuls in Richtung nach oben besitzen.

Dieser Effekt ist bei der fischertechnik-Kette erkennbar, aber eben deutlich weniger ausgeprägt.

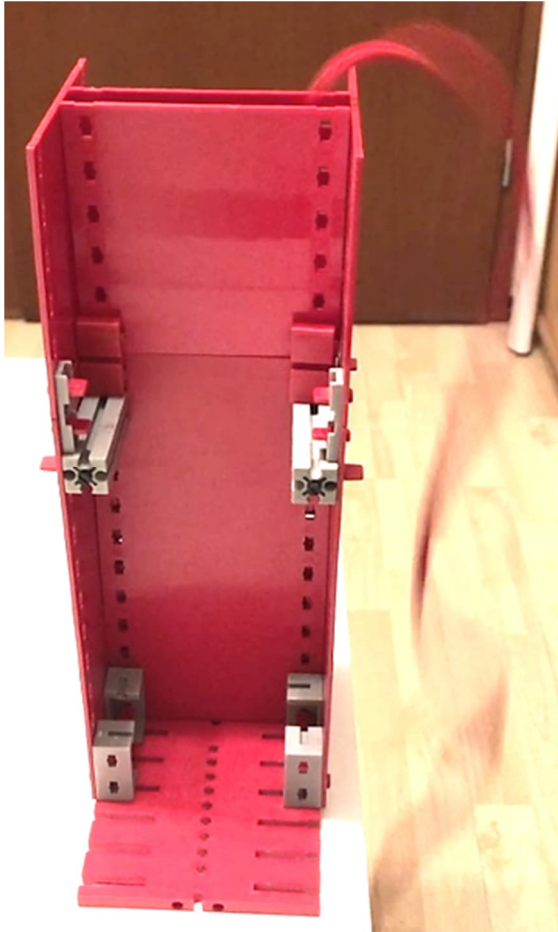


Abb. 8: Kettenfontäne

Referenzen

- [1] A. Ruh: [Kettenfontäne. Der Mould-Effekt](#). 2015.
- [2] J. S. Biggins, M. Warner: [Understanding the Chain Fountain](#). Submitted to Royal Society, 2013.
- [3] Rüdiger Riedel und Stefan Falk: [Kettenfontäne mit fischertechnik](#). Auf [YouTube](#), 2022.

Modell

Großprojekt Seilbahn (Teil 6): Reifenförderer

Tilo Rust

Diese Serie begleitet das Großprojekt „Kuppelbare Einseilumlaufbahn (10-MGD)“ im Förder-technik-Museum Sinsheim von Anfang bis zur Fertigstellung und Ausstellung auf der BUGA 2023 [1] in Mannheim. In diesem Teil dreht sich alles um das Detail der Reifenförderer, jenem zentralen Mechanismus, der die Gondeln innerhalb der Station auf den Schienen fortbewegt. Und es dreht sich um ein wunderbares Lehrstück zum Thema Scheitern und Neubauen.

Umlauf

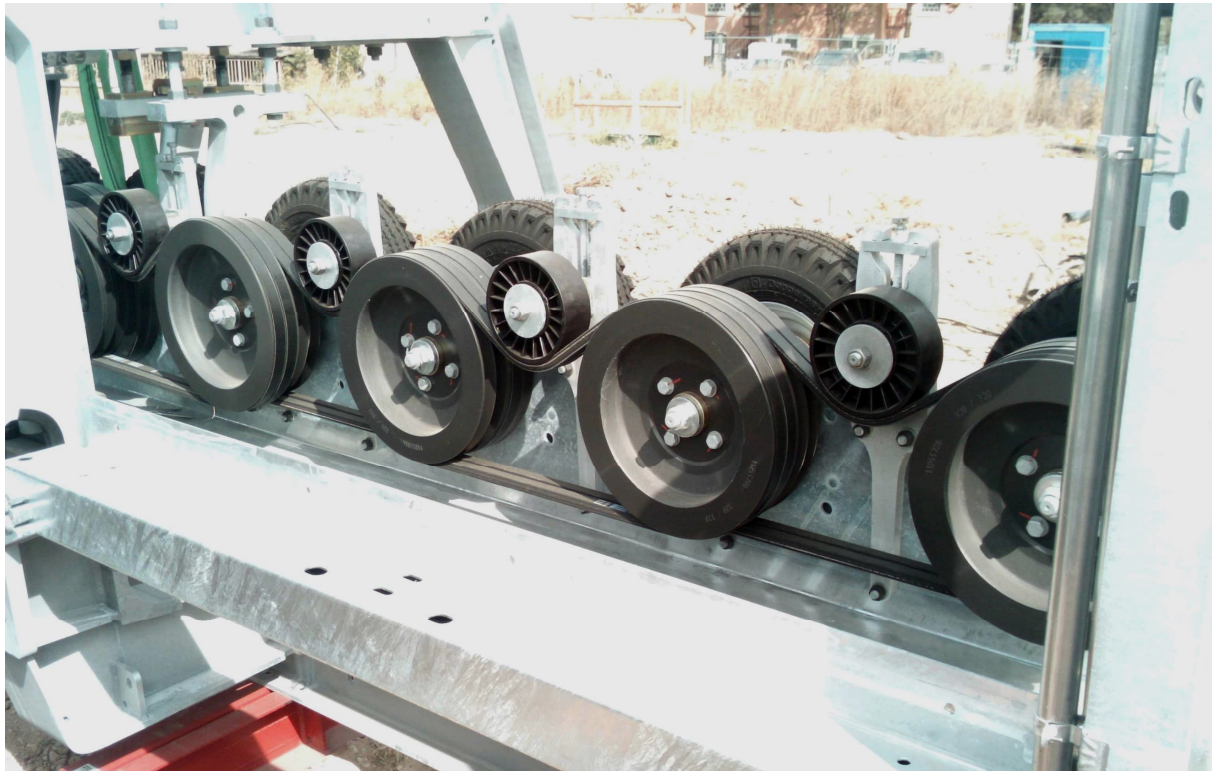
Wenn bei einer kuppelbaren Einseilumlaufbahn die Gondeln oder Sessel (fachlich korrekt als „Fahrzeuge“ bezeichnet) am Seil in die Station einfahren, dann haben sie eine Geschwindigkeit von etwa 6,5 m/s (bei unserem Modell maßstabsgetreu also ca. 65 cm/s). Das ist natürlich zu schnell, um einzusteigen. Aus diesem Grunde werden die Fahrzeuge mit Hilfe einer Klemme –

ähnlich einer Wäscheklammer – vom Seil getrennt. In der Station fahren sie dann auf einer Schiene, werden abgebremst, die Tür bzw. der Bügel öffnet sich, man kann aus- und einsteigen. Dann, auf der anderen Stationsseite, schließen sich diese Sicherheitsvorrichtungen wieder, das Fahrzeug wird beschleunigt – exakt bis auf die Geschwindigkeit des Seils – und schließlich in nur 0,6 s eingehängt und festgeklemmt.



Abb. 1: Die Reifenförderer der originalen BUGA23-Seilbahn im Spinelli-Gelände Mannheim vor dem Einbau. Die Stahlkonstruktion, welche diese wie ein Bogen überspannt, nennen wir „C-Stücke“; sie wurden mehrfach in unseren Beiträgen erwähnt.

(Foto: TR, mit Dank und Genehmigung von Doppelmayr/Garaventa)



*Abb. 2: Die Rückseite (vom Stationsinneren gesehen) zeigt die Verbindung aller Reifen mit Hilfe von Keilriemen, welche jeweils mit einer kleinen Rolle gespannt werden. Man beachte: Die Lagerung ist im Original „nur“ mittig – für unser „Problemchen“ im Modell von Relevanz.
(Foto: TR, mit Dank und Genehmigung von Doppelmayr/Garaventa)*

Diesen Weg nennt man *Stationsumlauf*. Über die Klemme und die Gondel sowie die Türöffnung werden wir noch detailliert berichten.

Während das Fahrzeug aber auf besagter Schiene fährt, muss es auch angetrieben werden. Dies geschieht mit einer Reihe von Gummireifen, die sich permanent drehen und auf die Oberseite eines „Schuhs“ drücken, einer Lauffläche an der Klemme, und diese so vorantreiben.

Die ganze Konstruktion befindet sich im Dachbereich der Seilbahn über den Gondeln.

Funktion

Bei den Reifenförderern laufen die ersten Reifen exakt in der Geschwindigkeit des Seils, sodass die Gondel stabilisiert werden kann und quasi doppelt geführt wird. Der

anschließende Auskuppelvorgang geschieht in 0,6 bis 0,9 s. Ab diesem Moment treibt das Seil die Gondel nicht mehr an. Die aufeinander folgenden Reifen werden nun jeweils etwas langsamer, etwa 10 % pro Reifen, bis die Gondeln auf ca. 0,3 m/s abgebremst sind. In unserem Modell erreichen wir das durch Paarung eines Ritzels 30 und Ritzels 22 als 3D-Druck (dank an Michael Sengstschmid), im Original geschieht dies mit Keilriemen auf unterschiedlich großen Riemenscheiben.

Für einen Stationsumlauf sind in unserem Modell pro Station 120 Reifen nötig. Das zeigt, wie wichtig deren Ausfallsicherheit und modularer Aufbau sind. (Gebaut aus 120 Achsen, 240 Ritzeln mit Nabe, 240 BS 15 mit Loch als Lager – pro Station; das nennt man Massenproduktion.)

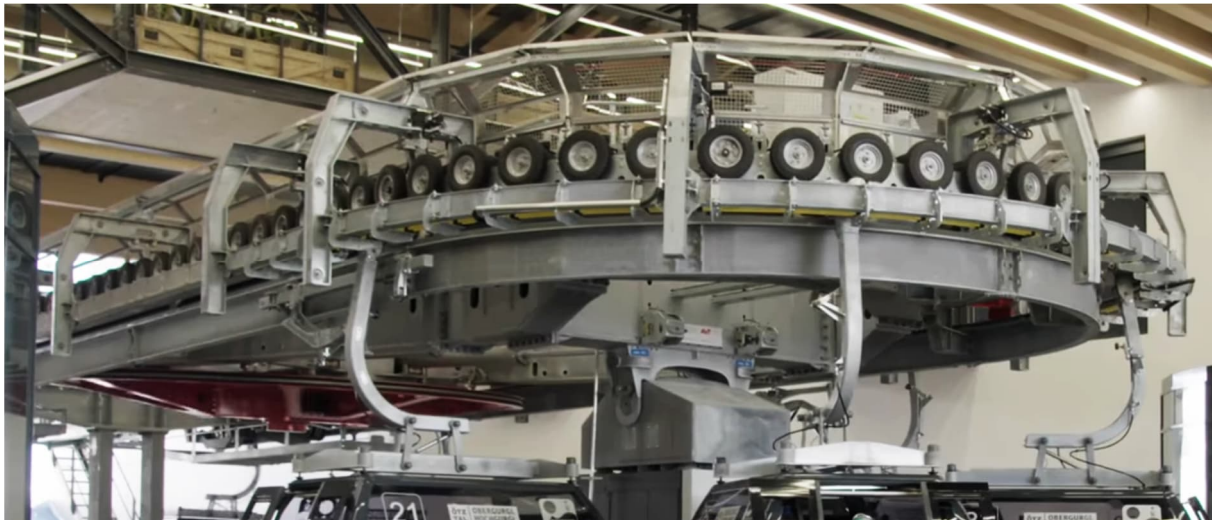


Abb. 3: Der Stationsumlauf als Ganzes gesehen (hier bei einem Skilift) zeigt, wie die vielen Reifen sich auch um die Kurve anordnen, wie die Gondeln auf der Schiene hängen und wie die C-Stücke die Niederhalteschiene tragen. Das wollen wir in unserem Modell natürlich so gut wie möglich nachbauen. (Foto mit Dank und Genehmigung von Doppelmayr/Garaventa)

Umsetzung

Wir haben uns das Original auf der Baustelle der BUGA23 genau anschauen können und die Details nachgebaut, soweit das machbar war. Mit folgenden wichtigen Änderungen: Bei allen Tests haben wir feststellen müssen, dass eine einseitige Lagerung der Achsen in der Mitte zu ungenau war und zu viel Reibung verursachte. Da unsere Anlage aber mehrere Stunden pro Tag ohne Ausfall laufen sollte, erschien es uns sicherer, ein zweites Lager auf der Innenseite anzubringen. Außerdem verwenden wir Ketten statt Keilriemen.

Ein weiteres Problem: Die Toleranzen im Original sind, gemessen an jenen, die wir zustande bringen, minimal. Diese werden von den luftgefüllten Reifen ausgeglichen. Um ebenfalls flexibel zu sein, entwickelten wir einen Reifenförderer, der sich selbst wie eine Schlange nach oben und unten bewegen kann und damit an die Klemmen anpassen sollte. Der Andruck sollte über Eigengewicht und zusätzliche Federbeine erfolgen. (Ist der Andruck zu klein, rutschen die Reifen. Ist er zu groß, sind Reibung und Kräfte zu stark, die Motoren müssen zu viel leisten, Ausfälle werden häufiger.)

Die Aufstellung der Prototypen zeigt unsere Versuche. Zuerst entwickelten wir ein Modell, das recht steif war, aber aus sehr vielen Teilen bestand. Hier hatten wir Bedenken, dass wir die Toleranzen nicht ausgleichen können (wie weitreichend dieser Irrtum war, dazu gleich mehr). Zudem war der Bau aus sehr vielen Bauteilen zeitaufwändig und wuchtig, so dass wir sogar an der Klemme Änderungen vornehmen mussten. Die gelben Streben waren nötig, um alle Teile zu verriegeln.

Die vereinfachte und schmalere Version basierte immer noch auf den alten Reifen 45 und einer Lagerung auf beiden Seiten. Bemerkenswert sind die Gelenkbausteine 45 nach jeweils vier Reifen. In der ersten Version hatten wir jede Achse einzeln in den Bohrungen dieser Gelenksteine gelagert („Schlange“). In der eingebauten Version wurden diese Module mit Ketten abgehängt und zusätzlich mit Flexschiene angedrückt bzw. in Längsrichtung stabilisiert. Die Überlegung war, recht steife Reifen zu benutzen, diese aber flexibel zu lagern, um so die Höhentoleranzen abzufedern. Daher wählten wir die alten harten Reifen 45 ([31018](#)), da wir befürchteten, dass die

Mäntel der neuen Reifen auf den Felgen rutschen könnten. Dies erwies sich später als großer Nachteil und Irrtum.

Wir bauten ca. drei Meter dieser Version in die Antriebsstation ein, motorisierten diese und versuchten, die Gondeln damit zu

befördern. Das gelang zwar, aber sobald eine zweite Gondel in der Strecke oder die Klemme nicht genau gefertigt war, bewegte sich ein Teil der „Schlange“, welches eine Übertragung der Vertikalbewegung in die nächsten Module zur Folge hatte – die erste

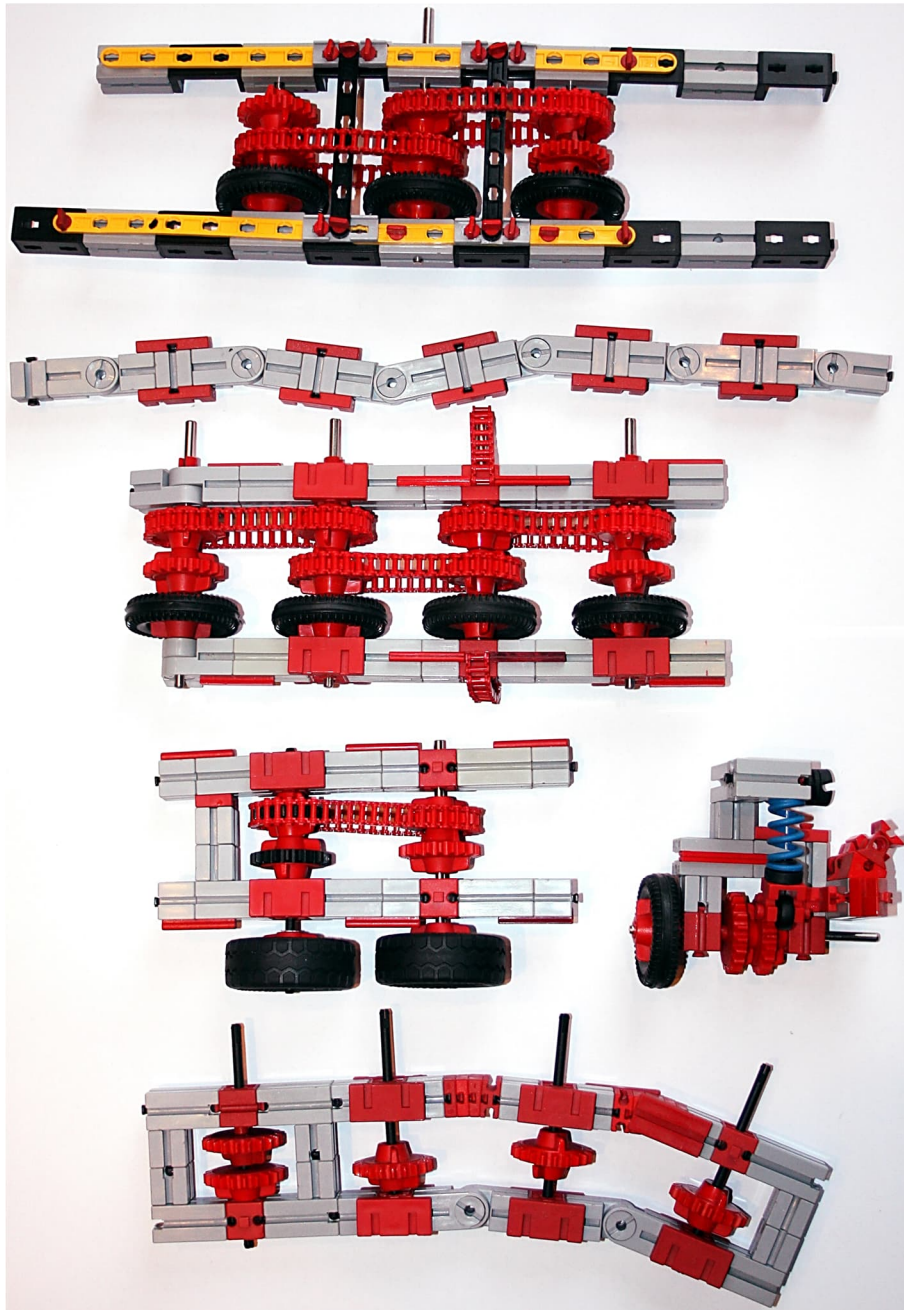


Abb. 4: Der erste Versuch (erste Baugruppe, mit den gelben Streben) erschien uns zu steif, Einzelaufhängungen zu aufwändig. Die „Schlange“ (zweite Baugruppe) und deren verbaute Version (dritte Baugruppe) schienen perfekt. Das Endprodukt (vierte Baugruppe) sieht aber vollkommen anders aus. Rechts daneben: ein Entwurf einer Einzelaufhängung. Unten: die erste Konstruktion einer Kurve – leider mit unpassenden Radien.

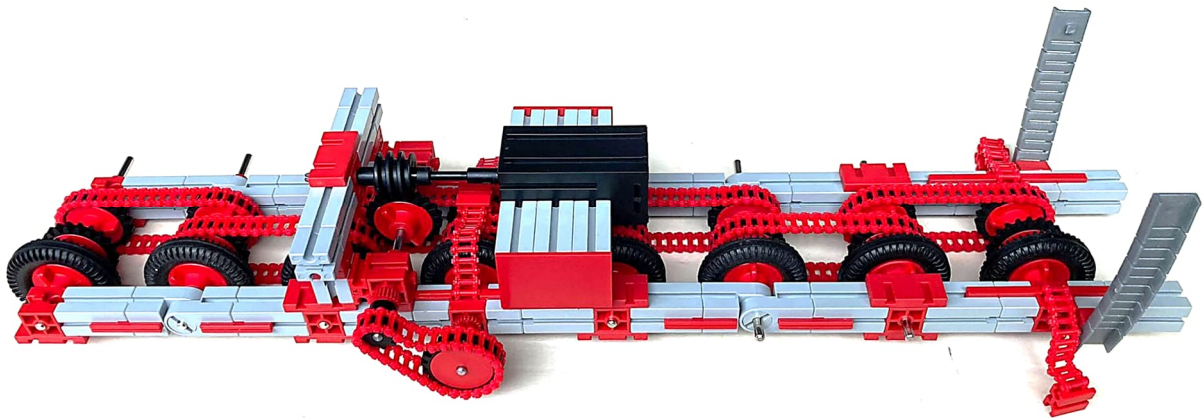


Abb. 5: Zwei komplette Module des alten Reifenförderers. Er ist mit den alten Gelenksteinen 45 (31008) flexibel gebaut, um Toleranzen auszugleichen – unser Fehler. Aufgehängt ist er an den kurzen Ketten seitlich (rechts, direkt vor den Flexschienen und in der Mitte hinter dem Ritzel 20), gefedert und geführt über die Flexschienen. Das Problem: Jede Gondel hebt den Reifenförderer an, bewegt damit die komplette zusammenhängende Anlage und bringt den Motor aus dem Gleichgewicht. Ein dynamischer Prozess, der nicht zu beherrschen ist.

Gondel ließ sich nicht mehr bewegen. Komplex war außerdem, dass die Motoren exakt in der Mitte gelagert werden mussten, da ihr Gewicht den Reifenförderer nach einer Seite kippen ließ.

Es war aussichtslos, wir verzweifelten an den Justage-Versuchen und scheiterten an zu vielen Parametern. Nach mehreren Wochen des Testens und Optimierens waren mehr und mehr Bauteile angebracht, nichts war mehr so wie geplant und nichts funktionierte.

Als ich (Tilo) nach einem sehr frustrierenden Bautag nach Hause fuhr und mein Leid, unsere Verzweiflung und die Angst des Scheiterns einem unserer Projektmitglieder schilderte, setzte dieser [SS] sich mitten in der Nacht hin und hatte den Eureka-Moment! Zwei Fotos später bastelte auch ich noch schnell auf dem Weg ins Bett. Und keine 24 Stunden später stand in einer Entwickler-Videokonferenz fest: Wir reißen alles ab! Die ganzen Reifenförderer zerlegen und alles von vorne. Scheitern tut weh.

Das Problem: Technisch gesehen verhielt sich unsere Entwicklung wie zwei aufeinander stehende Kugeln, die in alle Richtungen

beweglich waren. Sobald man Druck von oben aufbringt, weichen diese seitlich aus. Dieses labile Konstrukt kann nicht funktionieren.

Neustart

Der Reifenförderer musste komplett stabilisiert und seine Position genau definiert werden. Die nötige Toleranz und der Anpressdruck mussten aus dem Reifen kommen.

So schwer es mir als „Klassik-fischertechniker“ fiel, aber die alten, harten Reifen mussten den neuen (Felge 30 und Reifen 50) weichen. Stahlachsen wurden durch Kunststoffachsen ersetzt und die Lagerung so gestaltet, dass sich nicht viel bewegt, während die Module einfach zu bauen sind und sich fest in die C-Stücke einfügen.

All das geschah innerhalb von 24 Stunden: Eine komplette Neukonstruktion, erste Prototypen, fischertechnik-Designer-Zeichnungen und Funktionstests – Videokonferenz, Diskussion bis zum Entschluss und Bestellung der fehlenden Teile.

The Show must go on

Das war etwa eine Woche vor der ftc-Süd-Convention 2022, auf der unser Modell ausgestellt werden und laufen sollte, da uns Sponsoren besuchen wollten.

Deswegen fuhren wir im wahrsten Sinne des Wortes zweigleisig. Auf dem einen Gleis förderte der alte Reifenförderer immer schön eine Gondel von vorne nach hinten. Nicht zu häufig, um nicht auszufallen. Auf dem anderen (hinteren, versteckten) Gleis hatten wir den neuen Prototypen bereits verbaut und getestet – erfolgreich! Aber eben nur so klein, dass wir ihn nicht zeigen wollten.

Drei Wochen später: Die neuen Teile sind angekommen, die Halbzeuge gefertigt, unzählige Hände helfen beim Umbau. Nur Stunden später fuhren die ersten Gondeln ohne Probleme und ohne zu großen Justage-

Aufwand über mehrere Meter entlang der Schiene – und zwar mehrere Gondeln gleichzeitig.

Anschließend ging es an den Antrieb, der nun platziert werden konnte, wo wir wollten. Danach kam die Kurve an die Reihe. An dieser hätten wir uns mit dem alten Reifenförderer sicher die Zähne ausgebissen. Jetzt aber waren es nur wenige Bautage, bis der Stationsumlauf vollkommen funktionierte.

Zeitgleich wurde eine Blocksteuerung mit Hilfe von TXTs entwickelt, die für gleichbleibenden Abstand der Gondeln und Kollisionsvermeidung sorgt, ebenso für die Synchronisation von Seilgeschwindigkeit und Reifenförderer. Über beide Themen werden wir noch ausführlich berichten.

Die Neuentwicklung weicht in einigen Punkten deutlich von der alten Version ab –



Abb. 6: Der Antrieb der neuen Version ist als eigenständiges Modul ausgeführt und kann leicht gewartet werden. Je nach Bedarf kann er vorne oder hinten angebracht werden und stabilisiert dabei gleichzeitig den Reifenförderer.



Abb. 7: Der Stationseinlauf

was letztlich auch der Grund für den Erfolg ist:

- Die weichen Reifen geben nach und gleichen einige Ungenauigkeiten aus. Zudem ist deren Auflagefläche flach, im Gegensatz zu den alten Reifen. Damit stabilisieren sie die ebenfalls flachen Laufflächen der Gondeln.
- Die Achsen sind aus Kunststoff (nicht mehr aus Metall) und geben daher etwas nach. Das gibt uns mehrere Millimeter Toleranz.
- Hierzu mussten wir die Reifen einseitig lagern, damit die Förderschuhe die Reifen leicht anheben, die Kunststoffachsen biegen und damit den Anpressdruck aufbringen können.
- Das bedeutet, dass die vorderen Lager weiter nach hinten kommen, was die Stabilität und Montage vereinfacht. Außerdem schaffen wir so wieder Platz für die (gut gestalteten) Klemmen und vermeiden eine weitere Neukonstruktion.
- Alle Gelenke (alte Gelenksteine 45) wurden entfernt. Damit ist der Reifenförderer in alle Richtungen statisch

definiert. Gut abgespannt an den C-Stücken und zusätzlichem Fachwerkbau bewegt er sich nun nur noch minimal – genau so viel wie nötig, um die Toleranzen auszugleichen, aber genug Definition zu bieten.

- Da viele Bauteile (Flexschiene, Winkel, Streben, Gelenke, Ketten) entfallen, ist der neue Reifenförderer schneller und einfacher zu bauen und zu warten.
- Da der Reifenförderer nun nicht mehr beweglich aufgehängt ist, können die Antriebe beliebig angebaut werden.

Abb. 7 zeigt den Einlauf in die Station: Von links kommen die Gondeln und werden mit den „Trichtern“ auf die Gleise geführt, wenn die Fahrzeuge etwas schwingen. Direkt nach dem ersten Reifen des Förderers beginnt die Kuppelstrecke. Hier wird in 0,6 – 0,9 s die Klemme geöffnet. Dabei drückt die rote Kulissee die Klemme über den Totpunkt nach unten und gibt damit das Seil frei. So lange bewegt sich die Gondel noch mit Seilgeschwindigkeit. Danach (markiert mit dem gelben Statikbaustein 15, rechts oben) beginnt die Bremsstrecke. Auf der anderen Seite der

Station befindet sich das Gleiche noch einmal, die Gondel wird dort eingekuppelt und verlässt die Station nach links aus dem Bild. (Die gelbe Gondel im Hintergrund hängt nicht auf der Schiene, sondern hängt dort, wo sie ist, zu hoch.)

Krumme Dinger

Gerade zu bauen ist ja kein Problem – eine Kurve mit vorgegebenem Radius dagegen schon. Vor allem, wenn um diese Kurve in genau definierten Abständen Laufschiene, Niederhalteschiene, zwei Lagerschienen des Reifenförderers, zwei Schienen (aus Statik) zur Stabilisierung und eine Schiene zum Tragen der Last eingebaut werden müssen – und zwar so, dass sich die Reifen an genau definierten Stellen befinden, alle Lager genau fluchten und alles hängend an einer gemeinsamen Senkrechten befestigt ist.

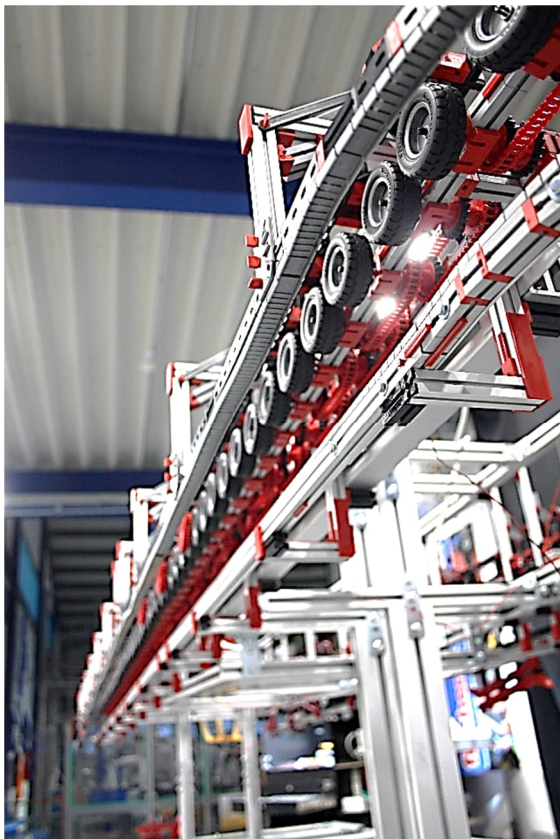


Abb. 8: Die imposante Ansicht der linken Laufbahn mit Reifenförderern. Eine Serienproduktion, die funktioniert.

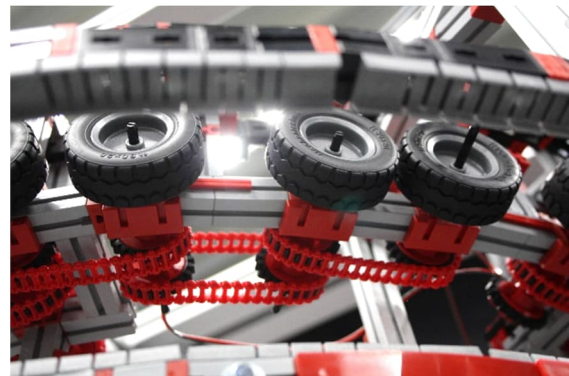


Abb. 9: Blick von unten auf die Kurve

Es bedurfte mehrerer Versuche und unterschiedlicher Ansätze. Doch jetzt sind wir damit fertig und schon die erste Gondel lief wie geschmiert.

Interessant hierbei: Die Ketten lassen sich, wie bei einer Kettenschaltung am Fahrrad, auch in der Kurve um einen gewissen Winkel biegen. Diese Idee stammt aus [3].

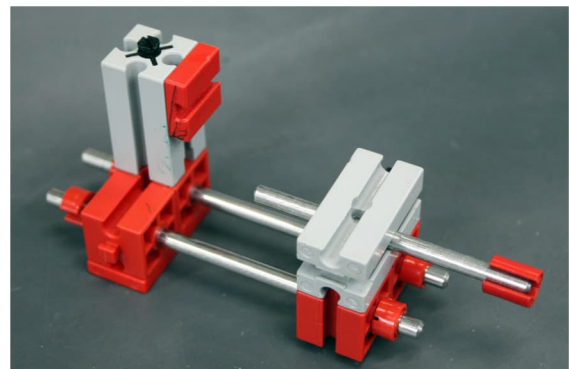


Abb. 10: Lochlehre

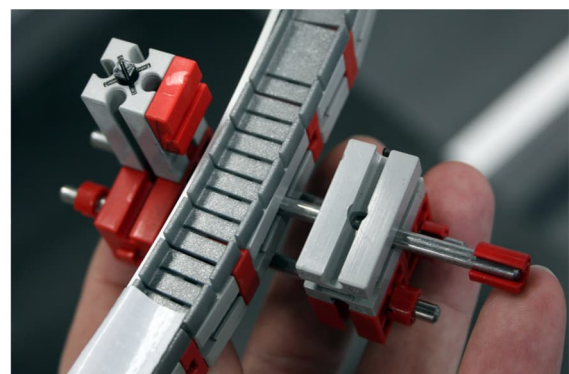


Abb. 11: Lochlehre beim Anbau

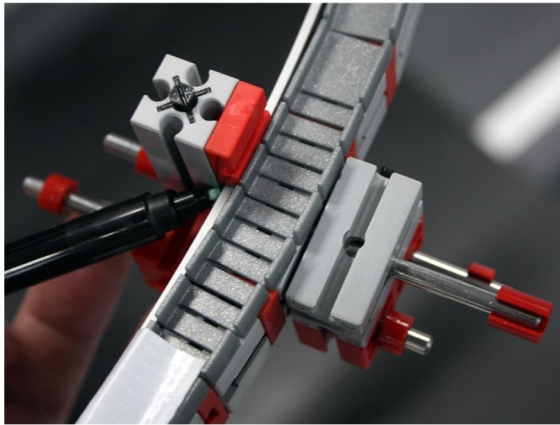


Abb. 12: Lochlehre im Einsatz

Abb. 10-12 zeigen die Loch-Lehren, kleine Werkzeuge aus fischertechnik: Mit dieser Lehre können wir die Löcher in den BS 30 mit Loch genau fluchtend auf die Rückseite der Alu-Schiene übertragen, die für die Verstärkung der Kurve eingebaut werden soll. Da die Gondeln etwa 1,5 kg wiegen

werden, können wir das Gewicht nicht mit der Konstruktion aus BS 30, BS 15 und Winkel 7,5° tragen, ohne dass diese sich über die zulässige Toleranz nach unten durchbiegt und damit die Reibung am Reifenförderer zu gering wird. Daher haben wir (so unauffällig wie möglich) eine Alu-Flachschiene innen in der Kurve angebracht. Diese hält mit Schrauben die BS 30 Loch auf Position.

Die Schwierigkeit ist dabei, die Löcher zu markieren, während die Bögen spannungslos aneinander liegen. Die gezeigte Lehre wird von innen auf die Alu-Schiene gehängt, gleitet an ihr entlang und wird mit der Achse und dem Klemmring 10 im Loch eines BS 30 ausgerichtet. Die seitliche Kante des inneren BS 30 (ohne Loch) fluchtet jetzt genau mit dem Loch.

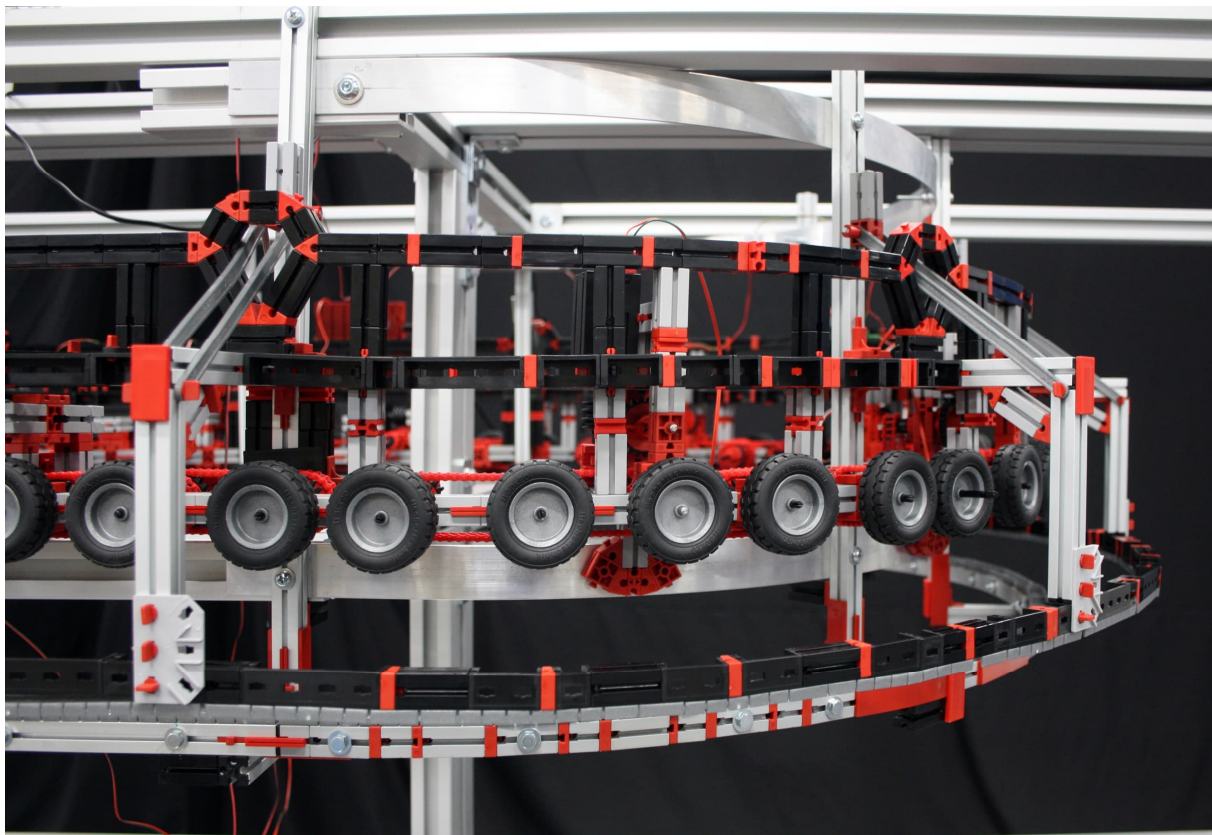


Abb. 13: Die Kurve stellt eine besondere Herausforderung dar. Mit viel Tüftelarbeit und vielen Winkelbausteinen bewältigten wir aber auch diese. Schön zu sehen ist im Detail von oben, wo die Antriebe sitzen und im Detail von unten, dass wir teilweise die Achsen einseitig lagern, wie im Original.



Abb. 14: Die Kurve von oben gesehen

Kuppelstrecke

Die nächsten Herausforderungen waren die Kuppelstrecke und die Türöffnung (nicht abgebildet). Hier muss der Vortrieb der Gondeln so stark sein, dass die Kräfte aufgebracht werden, um die Klemme in 0,6 s zu öffnen bzw. zu schließen oder die Türen zu betätigen. Mit der alten Version wäre das technisch nicht machbar gewesen. Mit der neuen Version gelang das schon beim ersten Durchlauf und ist bisher hundertfach getestet worden (Danke an Maximilian Schuch und Christian Bergschneider).

Diese Fehlentwicklung hat uns zwar fast sechs Wochen gekostet, aber sehr viel gelehrt – und das sei die Botschaft an alle Tüftler der Nation: Wenn du merkst, dass etwas nicht geht, entwickle weiter und betrachte den Aufwand, den es benötigt, um

das Ding zum Laufen zu bringen. Und wenn du merkst, dass der Aufwand zu groß wird, dann entscheide dich, mit aller Konsequenz noch einmal von vorne zu beginnen, mache reinen Tisch – aber mache es richtig!

Zahl des Tages: 18.000

So viele Kettenglieder haben fleißige Hände ineinander geklickt, um die Ketten für die Reifenförderer zu bauen. Das entspricht ca. 88 Meter Gesamtlänge und benötigte etwa 40 Stunden Arbeit.



Abb. 15: Die abgebildeten Ketten sind für die zweite Station „Luisenpark“ gedacht. Jene für „Spinelli“ sind schon verbaut.



Abb. 16: Danke an die lachende Christina Petry und andere fleißige Hände

Außerdem sind bisher ca. 18.000 Bausteine (ohne Kettenglieder) für unser Modell geliefert worden; davon sind etwa die Hälfte in der Talstation „Spinelli“ verbaut. Nicht eingerechnet: Die je 1.000 Bauteile für die Seilscheiben, vier Rollenbatterien und weitere ca. 300 pro Gondel!

Erhöhter Zeitdruck

Jetzt brauchen wir in den nächsten Wochen alle Helferinnen und Helfer vor Ort⁶, um die zweite Station (Luisenpark) zu bauen, die erste (Spinelli) zu verkabeln und zur Funktion zu bringen. Das ist ein Aufruf an alle Leserinnen und Leser – meldet euch bei uns, wir brauchen euch, der Einstieg ist jederzeit möglich!

Außerdem haben wir eine „Stellenausschreibung“ (Ehrenamt): Anlagenbediener (m/w/d) auf der BUGA23 zwischen April und Oktober, stunden- bzw. schichtweise. Das Seilbahnmodell der Öffentlichkeit zeigen? Bitte bei uns melden⁷!

Quellen

- [1] BUGA23 – [Bundesgartenschau 2023 in Mannheim](#)
- [2] [YouTube-Kanal](#) des Projekts
- [3] Florian Bauer: *Kettenschaltung mit fischertechnik*. Auf der [ft Community](#), 2022.
- [4] Tilo Rust: *Großprojekt Seilbahn (Teil 1): Von der Idee zum Kick-Off*. [ft:pedia 1/2021](#), S. 56–65.
- [5] Tilo Rust: *Großprojekt Seilbahn (Teil 2): Erste Elemente*. [ft:pedia 2/2021](#), S. 45–53.
- [6] Tilo Rust: *Großprojekt Seilbahn (Teil 3): Die Stationen*. [ft:pedia 3/2021](#), S. 28–37.
- [7] Tilo Rust: *Großprojekt Seilbahn (Teil 4): Fundamentale Arbeiten*. [ft:pedia 1/2022](#), S. 37–43.
- [8] Tilo Rust: *Großprojekt Seilbahn (Teil 5): Bewegung*. [ft:pedia 3/2022](#), S. 10–18.

⁶ [Fördertechnik-Museum Sinsheim](#), Untere Au 4, 74889 Sinsheim (direkt neben dem Technik-Museum)

⁷ Kontakt zur Projektgruppe „Seilbahn“ über den Projektleiter Tilo Rust, Schifferstadt, ft.seilbahn@gmail.com

Optik

Optische Datenübertragung mit fischertechnik

Florian Bauer

Am Anfang stand die Idee, einen rotierenden Entfernungsmesser optisch auszulesen, um ohne Schleifringe auszukommen. Ein erster Schritt dafür wird in diesem Beitrag vorgestellt: Eine bidirektionale optische Datenübertragung für I²C-Signale mit fischertechnik. Wie sich zeigt, ist fischertechnik dazu geeignet, prototypische optische Aufbauten zu realisieren. Durch die Wahl von mehrfarbigen LEDs wird auch visuell etwas geboten.

Ende des 19. Jahrhunderts erfanden *Alexander Graham Bell* (1847-1922) und *Charles Sumner Tainter* (1854-1940) ein drahtloses Lichttelefon zur Fernübertragung von Schall: das Photophon [6], bei dem Sonnenlicht über einen durch Schallschwingungen zur Vibration angeregten Spiegel zu einem entfernten Empfänger übertragen wird, wo das optisch modulierte Signal von einem großen Parabolspiegel konzentriert und mit Hilfe einer Selenzelle und einem Lautsprecher wieder in akustische Schwingungen umgewandelt wird. Bell, der als Erfinder des Telefons gilt, war davon überzeugt, dass das Photophon seine bedeutendste Erfindung gewesen war. Bemerkenswert ist, dass Bell als Lichtquelle die Sonne verwendet hat. Zumindest der Sender war damals schon „klimafreundlich“.

Von dem Prinzip, Signale und Daten mit Licht zu übertragen, geht eine gewisse Faszination aus. Man denke an den berühmten „Lichtspruch“ aus der Science-Fiction-Serie „Raumpatrouille“ („Orion“). Auch beim Militär wurden bisweilen gerichtete Lichtstrahlen zur abhörsicheren Kommunikation oder auch zur Unterwasser-Kommunikation mit U-Booten [8] verwendet. Ein weiteres Beispiel ist das Lichtsprechgerät 80 der Deutschen Wehrmacht [9].

Aber erfreulicherweise gibt es auch Einsätze im zivilen Bereich: Neben der inzwischen zum Standard gewordenen Datenübertragung über Glasfaser z. B. das Open-Source-Projekt Ronja [7], bei dem Enthusiasten eine Möglichkeit für optische Netzwerkverbindungen geschaffen haben.

Heute, viele Jahre nach Bells Erfindung, ist die Signalübertragung mit Licht wieder aktuell. So wird derzeit (endlich) bundesweit das Glasfasernetz ausgebaut, mit dem Ziel, größere Übertragungsbandbreiten als über Kupfer zu realisieren.

Auch fischertechnik hat Experimente zur Signalübertragungen mit Licht vorgesehen. Die dafür notwendigen Teile sind Linsen, Lichtleitstäbe und Fotowiderstände.

Neben dem Interesse an dieser Technik an sich gibt es auch praktische Gründe, die für einen Einsatz optischer Datenverbindungen sprechen – zum Beispiel, wenn man eine galvanische Trennung der Kommunikationspartner benötigt, starke elektromagnetische Störungen vorhanden sind oder man Daten kontaktfrei zu einem rotierenden System übertragen möchte.

In diesem Beitrag soll gezeigt werden, wie man mit fischertechnik eine digitale bidirektionale Datenübertragung bis 100 kHz über einen einzigen Lichtwellenleiter (LWL) realisieren kann. Der Aufbau erlaubt

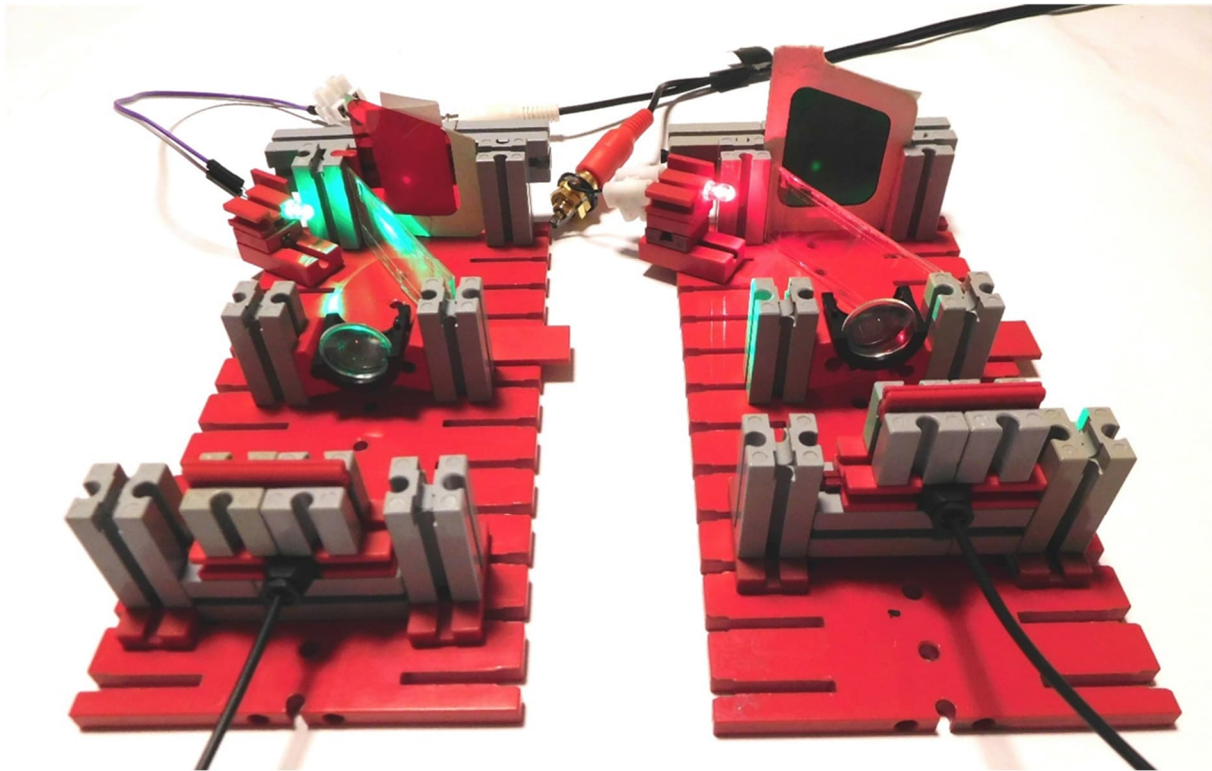


Abb. 1: „Langsame“ bidirektionale optische Lichtleiter-Übertragungsstrecke mit LEDs als Transmitter und Fototransistoren als Empfänger. Als Strahlteiler wurden Objektträger aus der Mikroskopie eingesetzt, als Linsen Kunststofflinsen mit Brennweite $f = 25 \text{ mm}$. Zur Sichtbarmachung der Empfänger-Reflexe stecken zwischen den Fototransistoren und den Farbfilterfolien Papierstreifen.

eine optische Isolation einer I²C-Verbindung und wird im weiteren Verlauf beschrieben.

Der Lichtleiter wird dabei in beide Richtungen verwendet. Wesentlich einfacher – und in der Praxis weiter verbreitet – ist es, getrennte Lichtleiter für Hin- und Rückkanal zu verwenden. Es gibt aber Anwendungsfälle, wo man mit nur einem Lichtwellenleiter auskommen muss, wie die erwähnte Übertragung in ein rotierendes System.

Ausgangspunkt für diesen Beitrag war ein schnell zusammengesteckter fischertechnik-Aufbau für eine Übertragung in zwei Richtungen (Abb. 1). Für Datenübertragungen mit Frequenzen im Audibereich wäre dieser Aufbau mit Fototransistoren ausreichend; für Datenraten bis 100 kHz werden schnellere optische Empfänger benötigt, von denen hier berichtet wird.

Mit den im weiteren Verlauf vorgestellten Elementen ist sowohl eine asynchrone Übertragung von RS232 oder eine synchrone Übertragung von I²C-Signalen möglich. Natürlich sind auch andere Protokolle denkbar.

Das fischertechnik-Baukastensystem hat sich für das Prototyping des optischen Aufbaus als sehr nützlich erwiesen. Die im Beitrag vorgestellte Elektronik wurde auf Proto-Boards zusammengesteckt.

Einfacher als die Übertragung mittels Lichtleiter ist die Übertragung im freien Raum (mehrere Meter), da hier leichter größere Signalpegel erreicht werden. Ich empfehle sogar, beim Nachbau erst einmal die LEDs direkt auf die Fotodioden scheinen zu lassen, um die Elektronik unabhängig von der durchaus anspruchsvollen optischen Justage zum Laufen zu bringen.

Transmitter und Sende-Lichtquellen

Als Lichtquellen kommen lichtstarke farbige LEDs (mit Lichtstärken > 4000 mcd) zum Einsatz, deren Licht auf ein Ende des Lichtleiters fokussiert und am Ausgang wieder kollimiert [1] wird (Tab. 1).

Um das Übersprechen von Sende- und Empfangskanal zu verhindern, werden die verschiedenen Kanäle mit LEDs unterschiedlicher Emissions-Wellenlängen (Farben) realisiert. Wichtig für die Auswahl der LEDs ist neben der Lichtstärke ein möglichst ähnlicher Emissions-Winkel.

Wenn die LEDs über längere Leitungen angesteuert werden, sollten diese abgeschirmt werden. Hi-Fi-Kabel mit Cinch-Anschlüssen bieten eine brauchbare Lösung. Lange, nicht abgeschirmte Leitungen können zu elektrischen Störsignalen führen, die die Detektion erheblich beeinträchtigen.

Strahl-Kombination verschiedener Lichtquellen (Kanäle)

Falls es mehr als einen Sende-Kanal gibt (wie für synchrone Datenübertragung notwendig), müssen die Strahlen der Sende-LEDs zusammengeführt werden, sodass sie von den Fokussierungslinsen gleich gut auf den Lichtwellenleiter konzentriert werden. Der zunächst naheliegende Einsatz von Mehrfarb-LEDs ist optisch ungünstig, da die Einzel-LEDs auf dem LED-Chip auf

leicht unterschiedlichen Positionen sitzen, so dass immer nur eine LED optimal auf den Eingang des LWL abgebildet werden kann.

Für die Strahlkombination werden Strahlteiler benötigt. Als billige Variante kann man 1 mm starke Glasplatten (Objektträger für die Mikroskopie) wie im Modell aus Abb. 1 einsetzen. Das funktioniert zwar, jedoch müssen wegen der geringen Reflexion (ca. 10% – 20%) sehr starke LEDs eingesetzt werden.

Kollimation und Fokussierung des Sende-Signals

Da das Licht der Sende-LEDs divergent ist, muss es zunächst mit einer Sammellinse kollimiert werden. Das parallelisierte Lichtbündel kann dann mit einer weiteren Linse kleiner Brennweite zum Einkoppeln auf das Ende des Lichtwellenleiters fokussiert werden. Es ist auch möglich, das Licht der LEDs durch eine einzige Linse zu fokussieren wie im Aufbau in Abb. 1. Wegen des Platzes, den die Strahlteiler erfordern, muss man die LEDs weiter weg von der Linse platzieren.

Es gilt zu beachten, dass durch seine endliche Größe jedes optische Element im Strahlengang das Lichtbündel beschneiden kann und den einzukoppelnden Lichtstrom zusätzlich durch Reflexionsverluste reduziert. Daher sollte man in diesem Fall auf LEDs mit kleinem Emissionswinkel

LED	Typ	Hersteller	Lichtstärke	Emissionswinkel	Peak-Wellenlänge
Rot	L-7113SEC-J3	Knightbright	11,0 cd	20°	640 nm
Blau	LL-504BC2E-B4-2GC	LuckyLight	8,5 cd	20°	462 nm
Grün	SLOAN L5-G81N-GU	Sloan	43,3 cd	15°	525 nm

Tab. 1: Verwendete LEDs

zurückgreifen. In meinem Aufbau verwende ich klare 5-mm-LEDs mit einem Emissionswinkel von ca. 20°. Eventuell sind 3-mm-LEDs besser, da sie wegen ihrer kleineren Emitterfläche stärker fokussiert werden können. Weil die kleineren LEDs im Allgemeinen eine geringere Leistung haben, müsste man ausprobieren, ob ihr Einsatz eine Verbesserung bringt.

Lichtwellenleiter

Als Lichtwellenleiter können die fischertechnik-Lichtleitstäbe verwendet werden, sofern sie blasenfrei sind, oder die im Hi-Fi-Bereich üblichen TOS-Link-Kabel, welche einen Kerndurchmesser des Lichtleiters von ca. 1 mm haben. Der Lichtwellenleiter arbeitet nach dem Prinzip der Total-Reflexion: Wenn Licht aus einem optisch dichteren Medium (Glas oder Kunststoff) in ein optisch dünneres (z. B. Luft) tritt, wird es ab einem bestimmten Grenzwinkel vollkommen (total) reflektiert. In einem Lichtwellenleiter macht man sich diesen Effekt zu Nutze. Das Licht wird beinahe verlustfrei durch den transparenten Kern übertragen. Daher hat jeder Lichtwellenleiter einen Akzeptanz-Kegel für das einfallende Licht. Nur Lichtstrahlen, deren Einfallswinkel innerhalb des Akzeptanz-Kegels liegt und für die Totalreflexion auftritt, werden übertragen (Details dazu in [10]). Der Rest geht verloren.

Kollimation und Fokussierung des Empfangs-Signals

Beim Austritt des Lichts aus dem Wellenleiter ist das Licht stark divergent. Aus diesem Grund sollte eine Linse mit kleiner Brennweite und großem Durchmesser nahe am Ausgang platziert werden. Es ist zu beachten, dass diese Linse auch zur Fokussierung bei der bidirektionalen Übertragung verwendet wird.

Trennung und Filterung des optischen Signals

Wenn Licht mehrerer Farben aus dem Lichtleiter kommt, muss es getrennt und auf die jeweiligen Detektoren geleitet werden. Glasplatten als Strahlteiler und nachgeschaltete Filter eignen sich hierfür nur bedingt, da das aus dem LWL tretende schwache Licht noch weiter abgeschwächt wird. Besser ist ein dichroitischer [2, 3] Strahlteiler aus einem Beamer. Solche keineswegs mehr exotischen Teile haben Wellenlängen-selektive Beschichtungen und trennen das Licht in drei Farben auf (siehe Abb. 2). Da es sich inzwischen um Standard-Teile handelt, kann man sie für wenige Euro über den Online-Versandhandel (oft aus China) beziehen.



Abb. 2: Recycelter dichroitischer Strahlteilerwürfel aus einem verschrotteten Beamer, der von einer weißen LED-Taschenlampe (unten) durchstrahlt wird

Eine weitere Filterung des optischen Signals kann durch Farbfolien geschehen, die aus transparenten, gefärbten Konsumprodukten (z. B. transparenten Duschgel-Behältern) oder Farbtransparent-Einbandfolien aus Kopiershops gewonnen werden können. Auch die Folien einer billigen Rot-Grün-Brille sind eine gute Quelle für Farbfiler.

Der Transmitter

Die LEDs können über eine zweistufige Transistor-Schaltung oder einen Open-Collector-Treiberbaustein wie dem TTL-IC 7407 angesteuert werden. Abb. 3 zeigt eine Schaltung. Da manche I²C-Sender den TTL-Eingang nicht direkt treiben können, ist ein CMOS-Buffer davor geschaltet. Für jeden Sendekanal wird eine solche Schaltung benötigt.

Der Empfänger

Als Detektoren kommen schnelle Fotodioden zum Einsatz, da handelsübliche Fototransistoren für die angestrebte Übertragungsrate (100 kHz für I²C) zu langsam sind.

Wenn die Fotodiode weiter weg von der Schaltung platziert wird, muss sie mit einer abgeschirmten Leitung angeschlossen werden, da die Leitungen gerne elektromagnetische Störsignale einfangen. Auch die Kapselung der Fotodioden in abgeschirmten Gehäusen hat sich als sehr vorteilhaft herausgestellt, da ansonsten wegen der eingesetzten schnellen Operationsverstärker auch schon mal ein unerwünschter „Radioempfänger“ entsteht.

Einfache Gehäuse für die Fotodioden können aus Kupfer-Lötfitting und Messingscheiben mit 15 mm Durchmesser aus dem Baumarkt zusammengelötet werden. Ins innere Loch der Messingscheibe wurde eine Schraubenfeder als Kabelknickschutz für das abgeschirmte Kabel gedreht und angelötet. Die Fotodiode ist auf eine zurechtgefeilte Platine gelötet.

Die schwachen Signale der Fotodioden müssen verstärkt werden. Dies geschieht mit einem zweistufigen Verstärker aus schnellen Operationsverstärkern und einem nachfolgenden Komparator, der das Signal auf Digitalpegel bringt. Die Schaltung wird im Folgenden detailliert beschrieben.

Empfangsschaltung

Die Kathode einer schnellen Fotodiode D1 (z. B. BPW 34 oder SFH 203) ist am invertierenden Eingang des ersten Operationsverstärkers angeschlossen. Ihre Anode liegt auf negativem Bezugspotential, in diesem Fall GND. Um Ladung der Diode schneller abzuführen und die Schwingungsneigung der ersten Stufe zu verringern, wird die Kathode über den Widerstand R1 zusätzlich auf Masse gezogen.

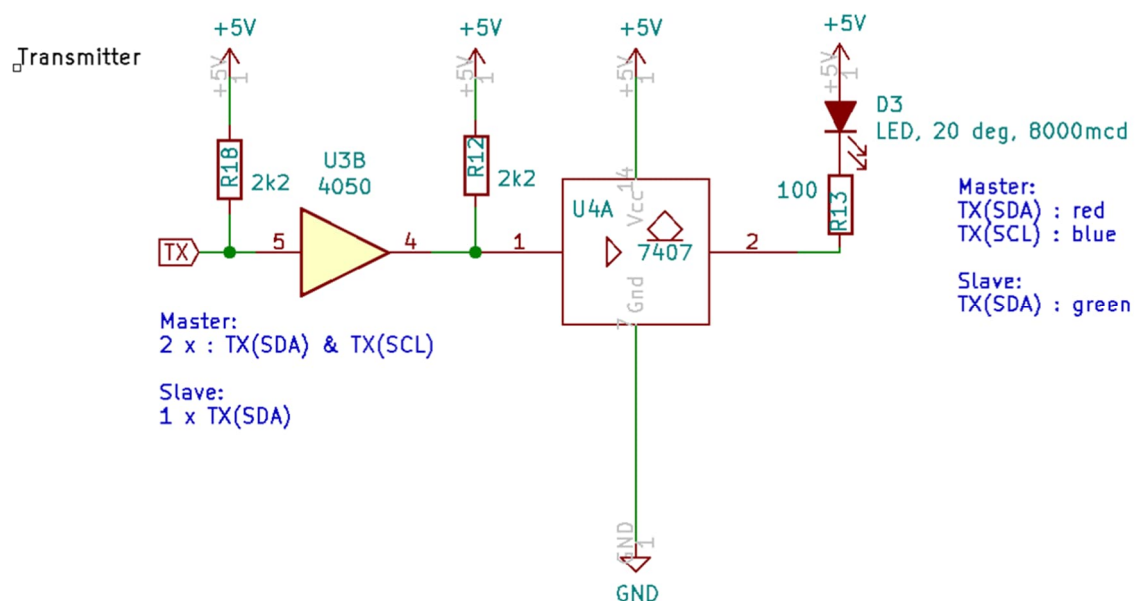


Abb. 3: LED-Transmitter mit TTL-IC 7404

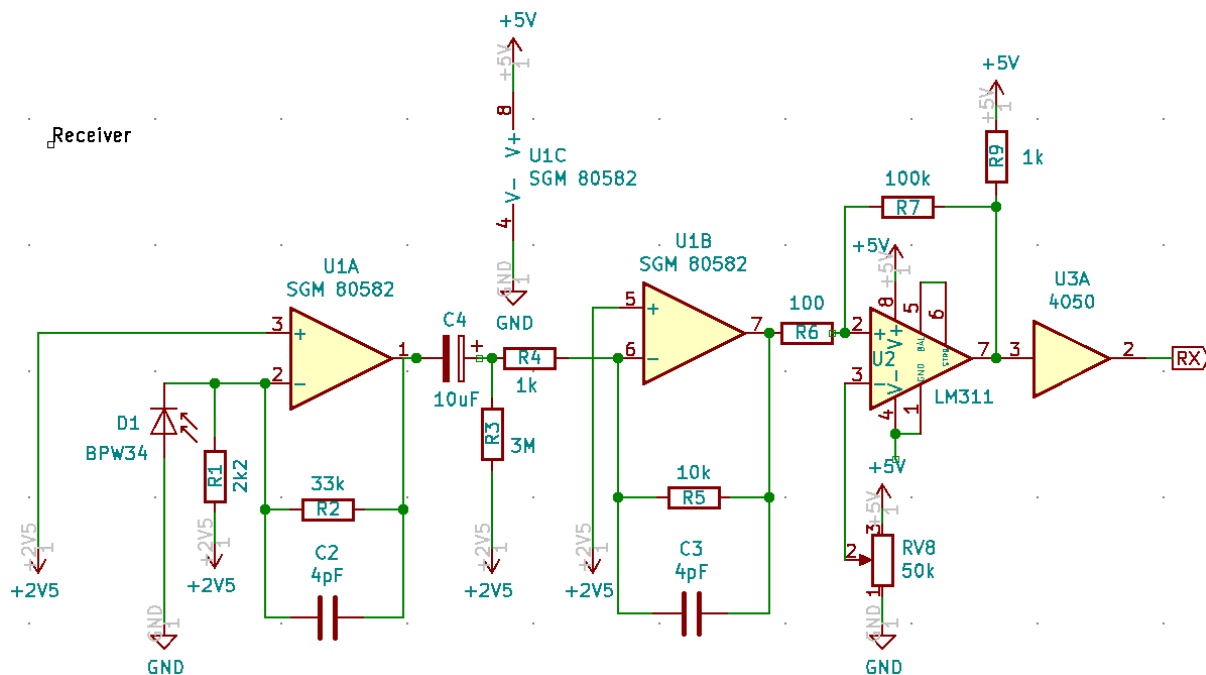


Abb. 4: Empfängerschaltung für optische Datenübertragung

Mit dem Feedback-Widerstand R2 entsteht ein invertierender Verstärker. Er ist über den Kondensator C4 an eine zweite invertierende Verstärkerstufe angeschlossen. Die kapazitive Kopplung verbessert die Unempfindlichkeit gegenüber Fremdlicht erheblich. Wegen der kapazitiven Einkopplung und des großen Eingangswiderstandes des Operationsverstärkers muss das Ruhepotential des Eingangs der zweiten Stufe mit Widerstand R3 nach Masse festgelegt werden.

Als Operationsverstärker ist ein Modell mit einem großen Verstärkungs-Bandbreiten-Produkt (oder auch *Gain Bandwidth Product*, GBP) zu verwenden, um scharfe Signale zu erhalten. Gute Ergebnisse habe ich mit LF357 erzielt, der ein GBP von 20 MHz hat. Da dieser aber erst mit symmetrischer Spannungsversorgung mit $\pm 5\text{ V}$ richtig läuft, bin ich auf SGM 80582 mit GBP von 220 MHz umgestiegen, der mit einer Gesamtspannung von 5 V funktioniert, was die Stromversorgung über einen Arduino ermöglicht. Darüber hinaus vereint der IC zwei Operationsverstärker in einem Gehäuse.

Allerdings steigt mit dem hohen Verstärkungs-Bandbreiten-Produkt auch die Schwingungs-Neigung der invertierenden Verstärker. Um sie zu verringern werden kleine Kondensatoren C2 und C3 (4 pF) parallel zu den Rückkopplungswiderständen platziert.

Virtual_Ground

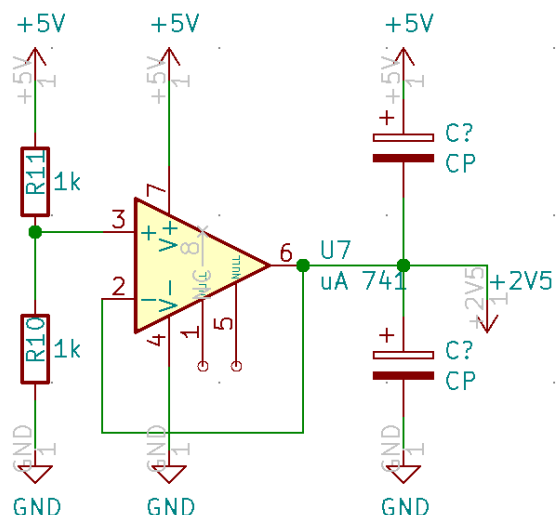


Abb. 5: Übliche Schaltung zur Erzeugung einer virtuellen Masse für den Receiver. Hier reicht ein einfacher Standard-Operationsverstärker (uA 741) aus.

Ab 200 mV Signal-Pegel nach der zweiten Stufe ist man „im Geschäft“ mit der Datenübertragung. Wenn das Signal frei von Störungen ist, kann auch schon ein geringerer Spannungspegel von ca. 50 mV ausreichen. Allerdings ist die Schaltung störempfindlicher und die Einstellung erheblich schwieriger. Insbesondere können Spikes dazu führen, dass sich die I²C-Übertragung „aufhängt“ und dann ein Reset des Mikrocontrollers notwendig ist.

Der nachfolgende Komparator, ein LM311, wandelt die analogen Spannungs-Impulse der zweiten Verstärkerstufe in saubere, kräftige Digital-Signale um. Der Schwellenpegel wird mit dem Potentiometer RV8 eingestellt. Da die Komparator-Schaltung bei verrauschten Signalen zu Spiking [4] und Ringing [5] neigt, führt ein großer Feedback-Widerstand R7 einen Teil des Signals auf den positiven Eingang des Komparators zurück. Es entsteht eine kleine Hysterese und der Komparator arbeitet

dann als Schmitt-Trigger. Dadurch werden die Signale sauberer und Fehlsignale verringert. Dieses aufbereitete Signal kann dann im Digitalteil der Schaltung weiter verarbeitet werden.

Aufbau für I²C-Übertragung

In Abb. 6 ist der optische Aufbau für eine bidirektionale optische Übertragung für I²C-Signale schematisch skizziert.

Für die I²C-Übertragung werden drei Kanäle benötigt:

- Ein Kanal für das Clock-Signal (SCK), das vom Master, einem Microcontroller (Arduino), an den Slave geschickt wird.
- Ein Kanal für das Daten-Signal (SDA) vom Master zum Slave.
- Ein Rück-Kanal für das Daten-Signal (SDA) vom Slave zum Master.

Der Master-Transceiver hat zwei Sender und einen Empfänger. Er sendet das Clock-Signal mit der blauen LED und das SDA-

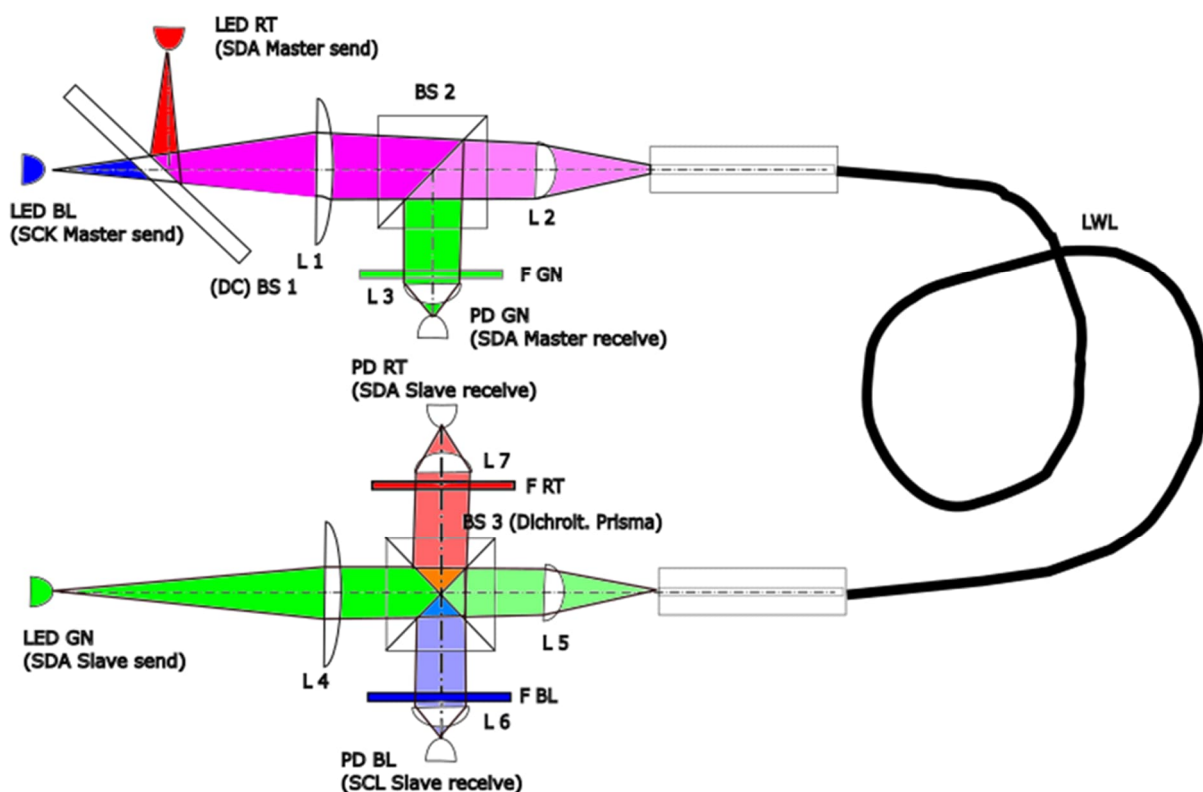
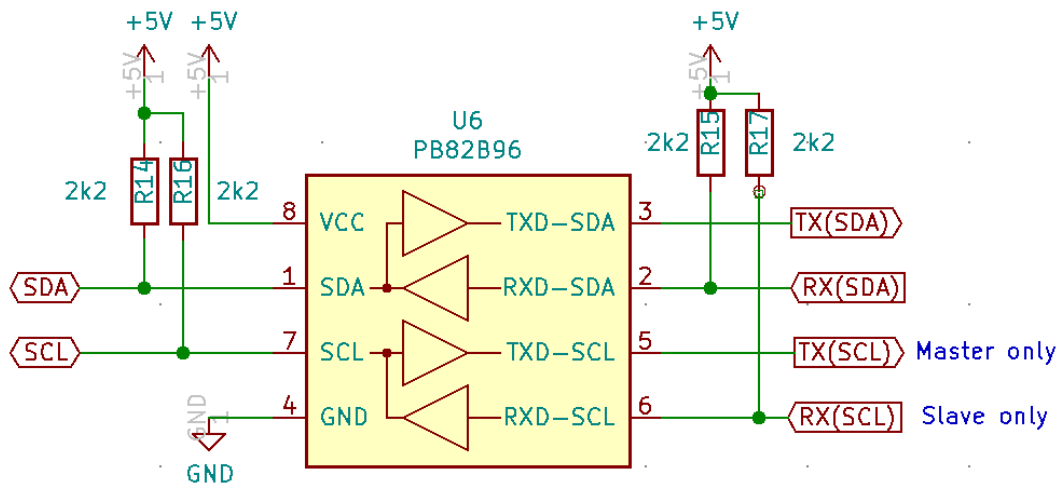


Abb. 6: Aufbau für bidirektionale Datenübertragung über einen Lichtleiter mit Master-Transceiver (oben) und Slave-Transceiver (unten)



└J2C-RXTX-Splitter_Option_2

Abb. 7: Splitter mit P82B96 für I²C zum Auftrennen in Transmit- und Receive-Kanal

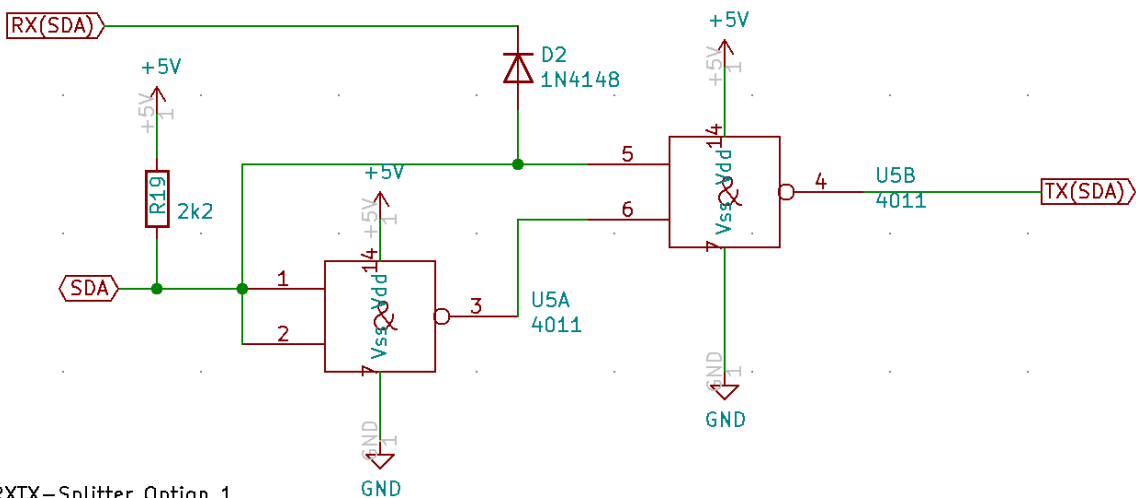
Signal mit der roten und empfängt das zurückgesendete grüne SDA-Signals des Slaves.

Der Slave-Transceiver hat zwei Empfänger und einen Sender. Er empfängt das blaue Clock-Signal des Masters und das rote SDA-Signal vom Master. Wenn der Slave antwortet, sendet er mit der grünen LED.

Bei galvanischer Ankopplung von Master und Slave würde die SDA-Leitung in beiden Richtungen verwendet. Die SDA-

Leitung wird über Pull-Up-Widerstände nach High gezogen und vom Master oder Slave nach Masse gezogen. Bei der optischen Übertragung müssen Hin- und Rück-Kanal aufgetrennt werden. Dafür gibt es den I²C-Transceiver-Baustein P82B96 (siehe Abb. 7, [24]).

Wer diesen IC nicht zur Verfügung hat oder wegen seines SO8-Formfaktors nicht verwenden will, kann die Trennung auch mit der kleinen Logik-Schaltung mit einem



└J2C-RXTX-Splitter_Option_1

Abb. 8: Variante mit CMOS-NAND 4011: Splitter für I²C zum Auftrennen in Transmit- und Receive-Kanal

Standard-CMOS-IC, die in Abb. 8 dargestellt ist, realisieren. Die Schaltung hat einen SDA-Anschluss, der wechselseitig von der SDA-Leitung des lokalen oder des entfernten I²C-Partners (RX) auf Low gezogen werden kann.

Über die Diode D2 wird das vom entfernten Partner empfangene SDA-Signal ohne Kurzschlussgefahr auf die SDA-Leitung des lokalen Partners geführt. Um zu verhindern, dass der lokale Partner sendet, wenn ein Signal vom entfernten Partner anliegt, wird das Senden über TX (SDA) unterdrückt, wenn ein Low-Signal an RX (SDA) vom entfernten I²C-Partner anliegt. Denn durch das zweite NAND-Gatter wird der SDA-Low-Impuls vom sendenden Partner nur dann auf TX (SDA) weitergegeben, wenn RX (SDA) High ist.

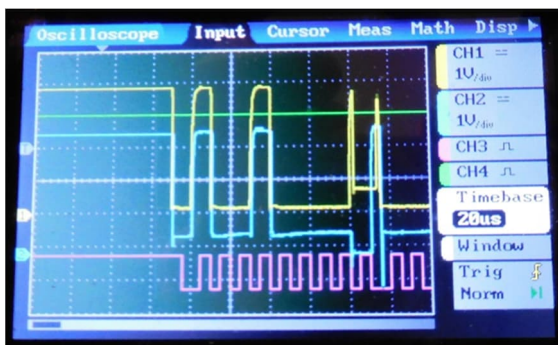


Abb. 9: Oszillogramm der SDA-Signale am Master (gelb) und Slave (cyan) mit Clock-Signal SCL (magenta). Sowohl beim Master als auch beim Slave wird das Empfangssignal der Gegenstelle über eine Diode an die SDA-Leitung geführt. Deshalb geht das Signal beim Empfang nur bis auf ca. 0,6 V herunter. Beim Senden zieht der Partner die Leitung vollständig auf Low.

Abb. 9 zeigt den zeitlichen Verlauf der SDA-Signale von Master und Slave auf dem I²C-Bus. Man erkennt sehr schön, wie der Slave, ein ADS1115, nach Übertragung der Bitfolge für die I²C-Adresse die SDA-Leitung auf Low zieht, um ein Acknowledge zu signalisieren. Auf der Master-Seite wird dieses Acknowledge-Signal registriert und die dortige SDA-Leitung wird über die Diode D2 bis auf die Diodenspannung von

0,6 V gegen GND gezogen. Dass das SDA-Signal vom Master vor diesem Acknowledge-Signal noch kurz auf High ist, wird durch Laufzeitverzögerung des Signals in der Schaltung verursacht. Mit der verwendeten Schaltung beträgt diese etwas unter 1 μ s, was offenbar noch tolerabel ist.

Aufbau mit fischertechnik

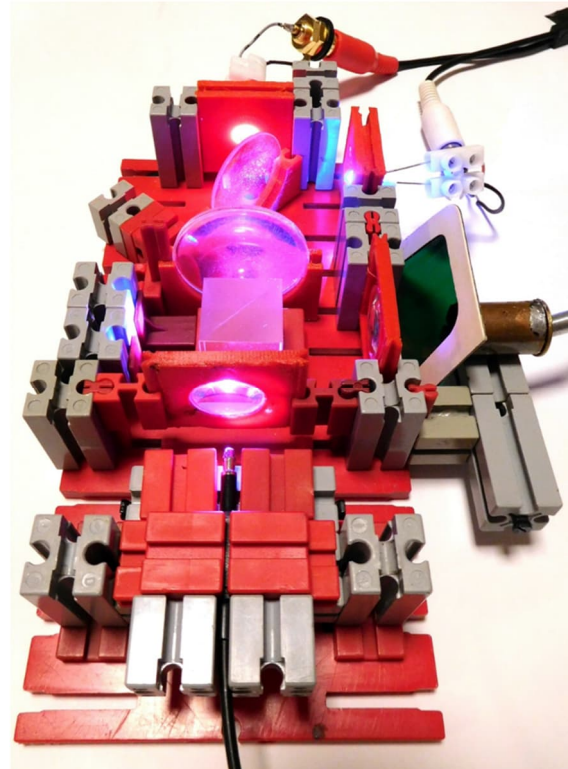


Abb. 10: Master-Transceiver mit roter Sender-LED für SDA, blauer LED für SCL und Empfänger-Diode mit Grünfilter davor für den Empfang des SDA-Signals vom Slave

In Abb. 10 ist der fischertechnik-Aufbau des Master-Transceiver gezeigt. Man erkennt (von oben nach unten) die Sender-LED für SDA (rot), den Strahlteiler und senkrecht dazu die Sender-LED für SCL (blau). Danach folgt der von Sammellinsen umgebene Signaltrennungs-Block mit dem Strahlteiler.

Am Eingang befindet sich eine große Kollimationslinse, am Ausgang die Sammellinse für die Einkopplung der Master-Signale (rot und blau) in den Wellenleiter und Kollimation des austretenden Rück-Signals des

Slaves (grün). Um den Strahldurchmesser auf die Größe der Detektorfläche der Empfangsdiode zu bringen, ist eine weitere Sammellinse platziert. Der Detektor für das SDA-Signal des Slaves befindet sich in dem abgeschirmten DIY-Kupfer-Gehäuse hinter einer grünen Filterfolie. Diese ist wichtig, da sonst ungewollte Reflexionen des blauen SCL-Signals an den Linsen und am Strahlteiler durchschlagen. Der Detektor muss auch in drei Achsen justierbar sein. Wenn das rote SDA-Signal optimal eingestellt ist, kann das SCL-Signal durch Höhenverstellung der blauen LED und Verdrehung des ersten Strahlteilers auf das Ende des Lichtleiters justiert werden.

Unten ist der Justierblock für das Lichtwellenleiterkabel zu sehen, dessen Aufbau zur Verdeutlichung separat in Abb. 12 gezeigt ist. Mit dem Justierblock kann der

Lichtwellenleiter auf der Bauplatte in drei Achsen verschoben werden, um optimale Einkopplung zu erreichen.

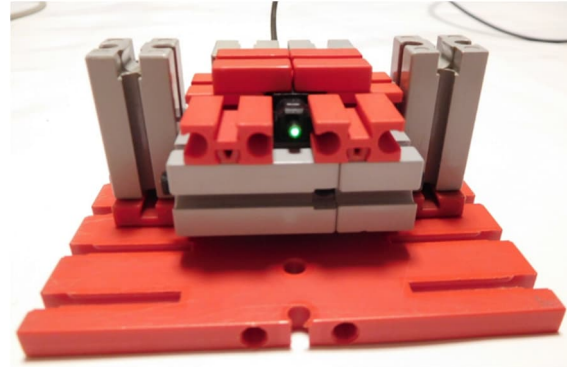


Abb. 12: In drei Achsen verstellbarer Justierblock für Lichtwellenleiter (TOS-Link-Leitung)

Der Aufbau des Slave-Transceivers ist in Abb. 11 zu sehen. Oben sieht man die grüne LED für das SDA-Signal des Slaves, der

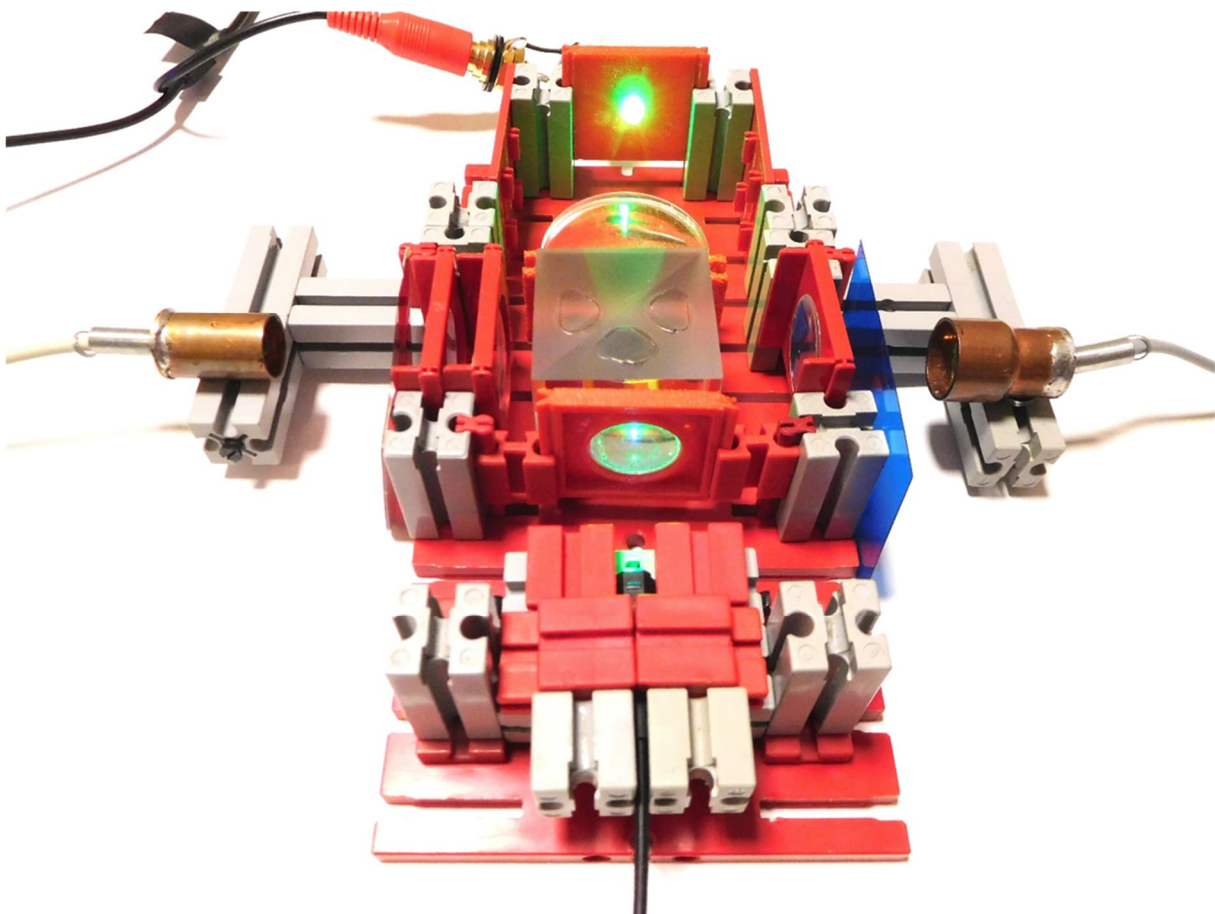


Abb. 11: Slave-Transceiver mit SDA-Empfänger (links), SCL-Empfängerdiode (rechts) und SDA-Transmitter (grüne LED)

durch eine große Kollimations-Linse in den vorher beschriebenen dichroitischen Strahlteilerwürfel einstrahlt. Der Würfel trennt die beiden Signale vom Master in das rote SDA-Signal nach links und das blaue SCL-Signal nach rechts. Vor den in drei Achsen verschiebbaren Empfangsdioden sind wieder Linsen und Farbfilterfolien platziert.

Linse	f	Durchmesser
L1	50 mm	30mm
L2	25 mm	18mm
L3	20 mm	18mm
L4	50 mm	30mm
L5	25 mm	18mm
L6 (ft)	35 mm	23mm
L7 (ft)	2 × 70 mm ~ 35 mm	23mm

Tabelle 2: Brennweiten f der verwendeten Linsen; Bezeichnungen gemäß Abb. 6

Wie in den Abbildungen zu erkennen ist, wurden neben den optischen Teilen ein paar Halterungen für Linsen und LEDs mit dem 3D-Drucker gedruckt. Die Parade dieser Teile ist in Abb. 13 zu sehen. Das parametrische Design wurde mit FreeCAD erstellt und die Quellen und STL-Dateien werden im fischertechnik-Forum zum Download zur Verfügung gestellt.

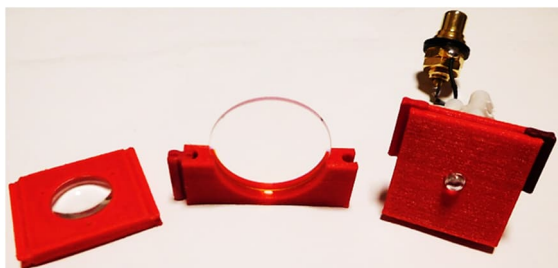


Abb. 13: Halterungen für Linsen und LEDs aus dem 3D-Drucker

Anwendungsbeispiel: Auslesen eines ADS1115 mit dem Arduino

Als Anwendungsfall wurde der AD-Wandler ADS1115 an einen Arduino UNO angeschlossen. Das Programm initialisiert den AD-Wandler, liest die Spannungen kontinuierlich aus und gibt sie über die serielle Schnittstelle aus.

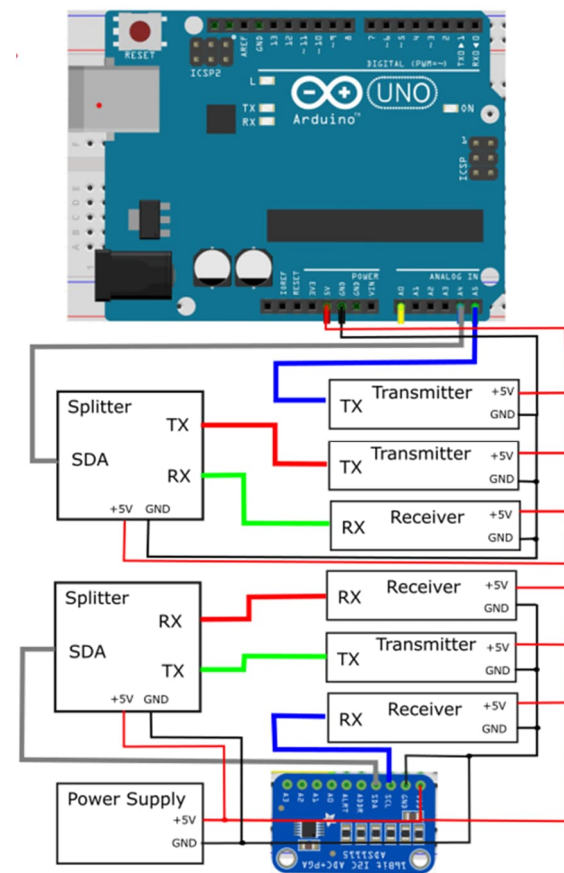


Abb. 14: Verdrahtungsplan der Übertragungsschaltung zwischen Arduino und ADS1115.

Für die Receiver wird noch jeweils eine virtuelle Masse (siehe Abb. 5) benötigt, die wegen der Übersichtlichkeit in der Abbildung weggelassen wurde.

Justage

Im Folgenden wird die Einstellung des Aufbaus beschrieben. Je nach verwendeten Linsen und Emissionswinkeln der LEDs können sich andere Abstände des Aufbaus ergeben, die man durch Ausprobieren ermitteln muss. Die Justage besteht aus

einer rein optischen Vorjustierung ohne Messgerät, gefolgt von einer Optimierung der elektrischen Signale. Hier ist ein Oszilloskop sehr hilfreich. Wenn man die LEDs mit einem Arduino im PWM-Betrieb ansteuert, kann man die Signale am Ausgang der Verstärkerstufen auch mit einem AC-Voltmeter messen.

Optische Vorjustierung

1. LEDs im Dauerbetrieb leuchten lassen
2. Am Master wird mit dem Justierblock auf maximale Einkopplung von „Rot“ justiert. Man beobachtet dabei das Licht, das aus der anderen Seite des Lichtleiters herauskommt.
3. Danach erfolgt die Justage der maximalen Einkopplung von „Blau“. Hierbei dürfen nur der Strahlteiler und die Höhe der LED verändert werden.
4. Am Slave wird mit dem Justierblock, der Position der LED und der Linse auf maximale Einkopplung von „Grün“ justiert.
5. Am Slave bringt man den Spot des blauen Signals auf Receiver-Fotodiode durch Drehen des Strahlteilers; Verschieben der Höhe der Linse und der Detektorposition.
6. Der Spot des roten Signals wird auf die Receiver-Fotodiode gebracht. Verstellt werden dürfen dabei nur Linse und Detektorposition.
7. Zurück zum Master: Der Spot des grünen Signals wird auf Receiver-Fotodiode für „Grün“ gebracht: Nur durch Drehung des Strahlteilers, Verschiebung der Linse und der Detektorposition.

Feinabstimmung mit Elektronik

8. Das Programm zum Auslesen des AD-Wandlers (*Continuous.ino* in *examples/Continuous* [19]) wird in den Arduino geladen.
9. Der grüne Kanal wird unterbrochen und das Trimpotentiometer für den

grünen Kanal so eingestellt, dass die rote und blaue LED blinken.

10. Durch feinere optische Justage: Signale für rot und blau an den Ausgängen der Verstärkerstufen optimieren, sodass ein Signalpegel von mind. 200 mV erreicht wird (Auf Spikes achten!).
11. Trimpotentiometer der Komparatoren so einstellen, dass die gesendeten Signale repliziert werden. Wenn „Rot“ und „Blau“ richtig eingestellt sind, sollte die grüne „LED“ im Slave hell blinken.
12. Der Strahlengang für „Grün“ wird freigegeben und das Potentiometer für den Empfänger im Master vorsichtig eingestellt, bis die Schaltung „zündet“ und alle LEDs stroboskopisch zu flackern beginnen. Wenn dann in der Arduino-Konsole noch richtige Werte angezeigt werden, kann man sich freuen und das Glücksgefühl genießen, dass es läuft. Dreht man das Potentiometer zu weit oder liegen Störungen vor, hängt sich die Schaltung auf. Sollten sich die Betriebsbedingung nicht erreichen lassen, kann dies auf Spikes, Timing-Probleme, auf zu schwache oder zu starke Signale zurückzuführen sein.
13. Alternativ kann man auch die SDA-Signale an Slave und Master mit einem Oszilloskop, sofern zur Hand, beobachten und so wie in Abb. 9 gezeigt einstellen.

Fazit

Dieser Beitrag zeigt, wie man eine bidirektionale optische Übertragungsstrecke mit fischertechnik realisieren kann. Das fischertechnik-Baukastensystem hat sich für das schnelle Ausprobieren von Prototypen als sehr flexibel und nützlich erwiesen. Eine asynchrone Zwei-Kanal-Übertragung lässt sich mit wenig Aufwand realisieren. Hier reichen Mikroskop-Objekttträger als Strahlteiler aus.

Die synchrone Übertragung von I²C-Signalen bis 100 kHz erfordert drei Kanäle und

eine schnellere Elektronik und ist deshalb anspruchsvoller. Die notwendigen Konzepte, Schaltungen und Aufbauten wurden beschrieben. Die Signal-Verzögerung liegt bei unter einer Mikrosekunde.

Meine „Lessons Learnt“ bei diesem Projekt habe ich in Tab. 3 zusammengestellt.

Ausblick

Wünschenswert wäre ein fischertechnik-Transceiver-Baustein (Silberling), der einen Empfänger und einen Sender enthält,

vorzugsweise mit Spannungs-Kompatibilität mit dem fischertechnik-System und dem Arduino. Dafür bräuchte man ein entsprechendes kompaktes Platinen-Layout für die zahlreichen elektronischen Komponenten in SMD-Ausführung.

Als Justierhilfe könnte ich mir ein Arduino-Programm vorstellen, mit dem man ohne Messgerät auskommt. Der Arduino müsste die LEDs pulsen und die empfangenen Signale auswerten. Das Programm könnte die Signalstärke und die Verzögerung

Probleme	Maßnahmen
Optik	
optische Verluste durch Vignettierung	größere Kollimatorlinsen nahe der LEDs LEDs mit kleinem Emissionskegel (20°)
Reflexionsverluste an Linsen und Strahlteilern	(Dichroitische) Strahlteiler LEDs mit großer Lichtstärke
Einkopplungsverluste	Linsen mit kleiner Brennweite nahe Lichtleiter
Übersprechen der Signale	Unterschiedliche LED-Farben und Farbfilter
Elektronik	
Delay des übertragenen Signals	Einsatz schneller Operationsverstärker Vermeiden von Übersteuerung
Entstehung von Störsignalen	Abschirmung der Leitungen für die LEDs
Einstreuung von Störsignalen	Abschirmung der Detektoren und der Zuleitungen Trennung Digital- und Analogteil Sauberer elektrischer Aufbau (keine fliegenden Leitungen)
Ringling und Spiking	Hysteresis beim Komparator durch Feedback-Widerstand zum nicht-invertierenden Eingang Block-Kondensatoren für Spannungsquellen
Fremdlicht-Empfindlichkeit	Kapazitive Kopplung der Verstärkerstufen, Farbfilter, Blenden und Einhausung
Schwingungen der Verstärker	Kondensatoren parallel zu den Feedback-Widerständen Widerstand am Eingang nach Masse
Stabilität der Schaltung	Justage auf große Signalpegel von mind. 200 mV
Mechanik	
schlechte Fixierung der LEDs und Linsen	Konstruktion spezieller 3D-Druckteile
Justierbarkeit des Aufbaus	Justierblock für LWL
Stabilität des Aufbaus	Verstärkung mit Riegeln

Tabelle 3: Probleme und Maßnahmen zur Lösung

messen und auch, ob das Signal korrekt repliziert wird oder ob Drop-Outs oder Spikes vorhanden sind.

Für den Einsatz als Datenübertragung in ein rotierendes System ist der Aufbau mit Lichtwellenleiter noch zu unhandlich. Eventuell ließe sich, wie im Forum vorgeschlagen, mit recycelten Bauteilen aus einem CD-RW-Laufwerk eine kompaktere Lösung entwickeln.

Meines Erachtens wäre es aussichtsreich, wenn man statt des Lichtwellenleiters eine Freifeld-Übertragung wählen und die LED und Empfangsdioden mit vorgeschalteten Filtern nahe der Achse platzieren würde. Dann könnte man sich Strahlteiler und Linsen sparen, hätte nicht Verluste durch die Einkopplung und die Justage wäre unproblematischer. Die optoelektronischen Bauelemente könnten auch direkt auf eine kleine Platine gesetzt werden, was auch schaltungstechnisch von Vorteil wäre.

Es würde mich freuen, wenn dieser Beitrag etwas von meiner Begeisterung für die optischen Experimente mit fischertechnik vermitteln konnte und er Anregung für weitere Projekte in diesem Bereich ist.

Quellen

- [1] Wikipedia: [Kollimation](#).
- [2] Wikipedia: [Dichroismus](#).
- [3] Wikipedia: [Interferenzfilter](#).
- [4] Wikipedia: [Spiking](#).
- [5] Wikipedia: [Überschwingen](#).
- [6] Wikipedia: [Photophon](#).
- [7] Karel Kulhavy: [Ronja](#). Projekt zur optischen Datenübertragung.
- [8] Tsai-Chen Wu, Yu-Chieh Chi, Huai-Yung Wang, Cheng-Ting Tsai & Gong-Ru Lin: *Blue Laser Diode Enables Underwater Communication at 12.4 Gbps*. Auf [nature.com](#).
- [9] Wikipedia: [Optischer Richtfunk](#).
- [10] RP Photonics Encyclopedia: [Acceptance Angle in Fiber Optics](#).
- [11] Robert Keim: *Design Tips for Photodiode Amplifiers*. Auf [All About Circuits](#), 2021.
- [12] Christof Ermer: *Photo DIODEN Strom-Spannungswandler = Transimpedanz Verstärker*. [Uni Regensburg](#), 2022.
- [13] Bonnie Baker: *Design Transimpedance Amplifiers for Precision Opto-Sensing*. [Digi-Key Electronics](#), 2017.
- [14] Miro Samek: *Opto-electrical Isolation of the I2C-Bus*. [embedded.com](#).
- [15] SG Micro Corp: *SGM80581/SGM80582/SGM80584 220MHz, Rail-to-Rail I/O, CMOS Operational Amplifiers*. [Datenblatt](#), 2019.
- [16] Texas Instruments: *LFx5x JFET Input Operational Amplifiers*. [Datenblatt](#), 2015.
- [17] Vishay: *BPW34, BPW34S Silicon PIN Photodiode*. [Datenblatt](#), 2011.
- [18] Wolfgang Ewald: *ADS1115 – A/D-Wandler mit Verstärker*. Auf [Wolles Elektronik](#), 2020.
- [19] Wolfgang Ewald: *ADS1115_WE*. Arduino-Library zum Auslesen eines ADS1115 über I²C, [GitHub](#), 2022.
- [20] Kingbright: *T-1 3/4 (5mm) SOLID STATE LAMP*. Datenblatt zu verwendeter LED, bei [Reichelt](#), 2013.
- [22] Luckylight: *5mm Round With Flange Type Blue LED Technical Data Sheet*. Datenblatt zu verwendeter LED, [Reichelt](#), 2009.
- [23] Sloan AG: *Series L5 / Ø 5mm, T-1 3/4, Colour: green*. Datenblatt zu verwendeter LED, [Reichelt](#), 2011.
- [24] NXP Semiconductors: *P82B96 Dual bidirectional bus buffer*. [Datenblatt](#), 2021.

Elektronik

Elektronikmodule (Teil 9): Direkter Vergleich von Multifunktionsmodul und Grundbaustein

Hans-Christian Funke

In Teil 8 meiner Beitragsreihe habe ich beim Vergleich zwischen Silberlingen und Elektronikmodulen festgestellt, dass speziell beim Multifunktionsmodul und dem Grundbaustein der reine Vergleich der Ein- und Ausgänge in der Praxis nicht wirklich weiterhilft. Vielmehr wären reale Beispiele aus Modellen oder Schaltungen nützlich, um die Umsetzung und die Unterschiede zu verstehen und nachzuvollziehen. Dieser Erkenntnis soll mit diesem Beitrag Rechnung getragen werden.

Der Differenzverstärker

Der Grundbaustein h4G und das Multifunktionsmodul 70015 beinhalten beide einen Differenzverstärker und können somit grundsätzlich die gleichen Aufgaben übernehmen und austauschbar eingesetzt werden.

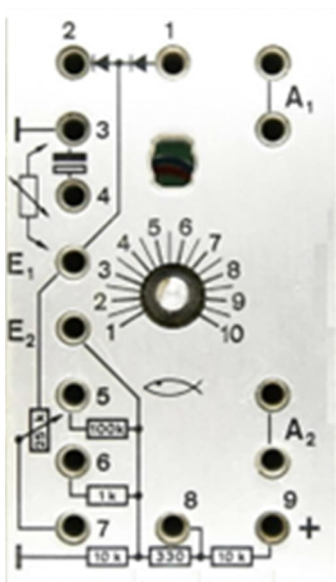


Abb. 1: h4GB

Die Schaltung

Vergleicht man die Schaltung der beiden Bausteine (Abb. 3 und 4) stellt man fest, dass die beiden Schaltungen sehr ähnlich

aufgebaut sind, was der Grund für die Austauschbarkeit ist.



Abb. 2: Multifunktionsmodul

Während die Verbindungen für die verschiedenen Funktionen beim h4G gesteckt werden müssen, können die meisten beim 70015 durch DIP-Schalter hergestellt werden. Darüber lassen sich Funktionen schnell aktivieren, aber die Flexibilität wird etwas eingeschränkt, wie es bei einigen der praktischen Beispiele zu sehen ist.

Praktische Beispiele

Als Grundlage für die praktischen Beispiele habe ich auf das Buch von fischertechnik hobby 4 [3] zurückgegriffen. Darin finden sich zahlreiche Beispiele und Modelle sowie Beschreibungen der einzelnen Funktionen des Grundbausteins.

In den nachfolgenden Zeichnungen sind Seitenzahlen aus dem Buch hobby 4 angegeben. Auf diesen Seiten wird die Funktion der abgebildeten Schaltung beschrieben. Ist in den Zeichnungen der Drehknopf bei einem Elektronikbaustein grün gekennzeichnet, dann kann mit dem Drehregler die Funktion der Schaltung beeinflusst werden (siehe Abb. 5). Schwarze DIP-Schalter sind eingeschaltet und befinden sich in der Position ON (Hebel nach unten).

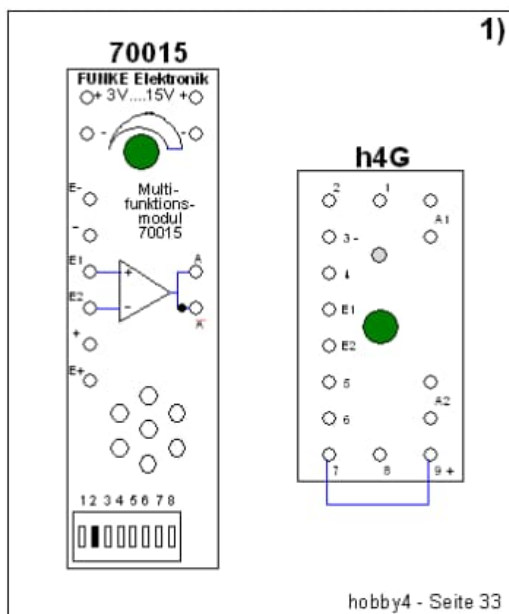


Abb. 5: Schaltung mit aktivem Drehregler und eingeschalteten DIP-Schalter 2 am 70015.

Schwellenwertschaltung für Foto- und NTC-Widerstand

In Abb. 6 und 7 sind Schwellenwertschaltungen abgebildet. Die Verbindung von Ausgang A2 über 100 kΩ an Eingang E2 wird beim 70015 mit dem DIP-Schalter 5 hergestellt.

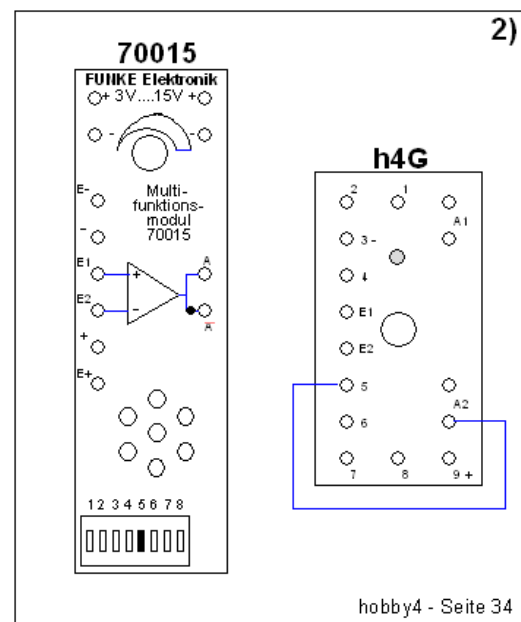


Abb. 6: Schwellenwertschalter

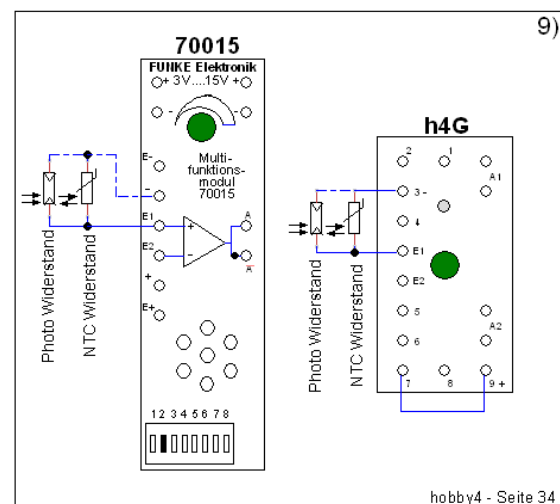


Abb. 7: Schwellenwertschalter mit großem Schaltabstand

In Abb. 7 wird der Drehregler mit eingebunden und der Schwellenwert kann über diesen verändert werden. Der Drehregler wird beim h4G dafür mit Plus verbunden; beim 70015 muss zur Herstellung dieser Verbindung der DIP-Schalter 2 eingeschaltet werden.

Schwellenwertschaltung für Feuchtheitsfühler bzw. mit höherem Schaltabstand

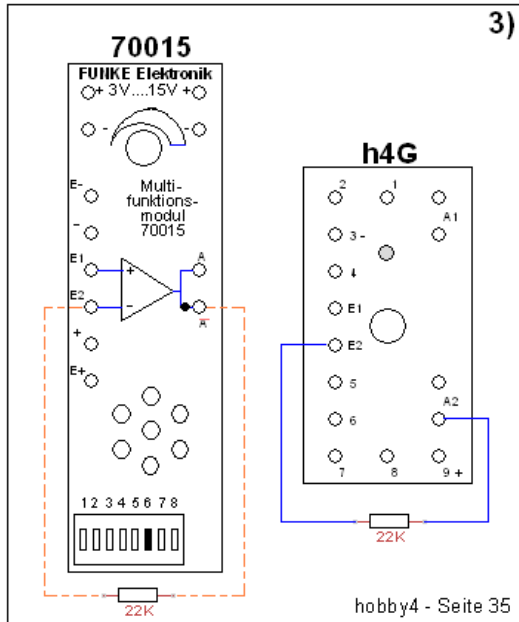


Abb. 8: Schwellenwertschalter mit höherem Schaltabstand

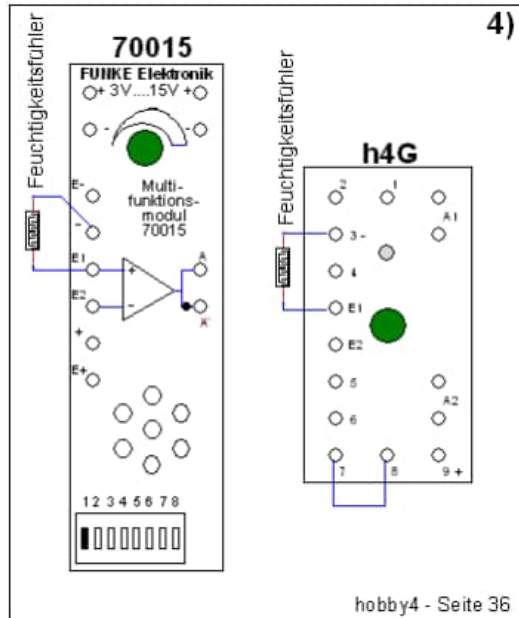


Abb. 9: Schwellenwertschalter für einen Feuchtheitsfühler

Um den Schaltabstand zu erhöhen, wird eine Rückkopplung des invertierenden Ausgangs über einen Widerstand nach E2 hergestellt (Abb. 8). Beim 70015 stehen dafür die drei DIP-Schalter 5 (100 kΩ), 6 (10 kΩ)

und 7 (1 kΩ) zur Verfügung. Alternativ kann auch jeder beliebige Widerstandswert direkt zwischen die Anschlüsse E2 und \bar{A} geschaltet werden; dann darf aber kein DIP-Schalter 5 – 7 eingeschaltet sein, weil sonst eine Parallelschaltung der Widerstandswerte erfolgt.

Für den Feuchtheitsfühler wird die Schaltung in Abb. 9 empfohlen – hier wird nur die halbe Versorgungsspannung an den Drehregler angelegt. Beim 70015 muss dafür der DIP-Schalter 1 eingeschaltet werden.

Schwellenwertschaltung für Kleinstsignale

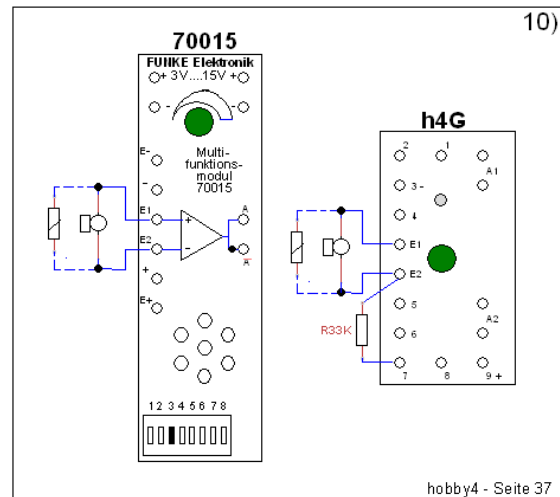


Abb. 10: Schwellenwertschalter für Kleinstsignale

Für die Auswertung von Kleinstsignalen, wie sie vom Mikrophon beim Auftreffen von Schallwellen erzeugt werden oder von Magnetspulen bzw. Elektromagneten im bewegten Magnetfeld, kann die in Abb. 10 gezeigte Schaltung verwendet werden.

Der Widerstand erhöht die Sensibilität des Differenzverstärkers und kann im Wert variiert werden. Im Leitfadensystem der Elektronikbausteine von fischertechnik [4] wird ein 22-kΩ-Widerstand verwendet. Zur Herstellung dieser Verbindung im 70015 muss der DIP-Schalter 3 mit einem Widerstandswert von 33 kΩ eingeschaltet werden. Die Verbindung zwischen E2 und

Drehregler könnte auch nur über eine Drahtbrücke realisiert oder weggelassen werden. Die Praxis hat gezeigt, dass der Einsatz eines 22-kΩ- bzw. 33-kΩ-Widerstandswertes die besten Ergebnisse geliefert hat.

Differenzverstärker in einem Zustand festsetzen

Bei diesen Schaltungen (Abb. 11 und 12) darf bei der Aktivierung über die Eingänge E1 und E2 nur mit Minus (low) gearbeitet werden: Bei Kontakt mit Plus (high) kann es zu unbeabsichtigtem Statuswechsel am Ausgang kommen – also mechanische Umschalter nur mit Minus beschalten. Die Schaltung liefert ein gutes Beispiel zur Verwendung der Buchsen (1) und (2) des h4G. Der 70015 verfügt über vergleichbare Eingänge – die Buchse (1) am h4G entspricht dem Anschluss E+ am 70015 und die Buchse (2) des h4G dem Anschluss E- am 70015.

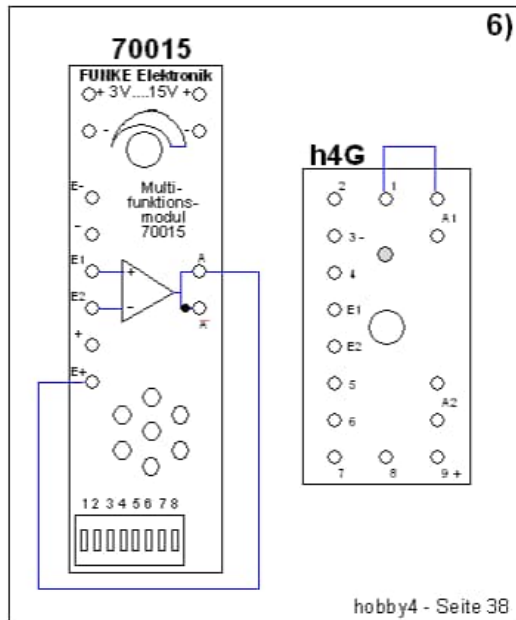


Abb. 11: Selbstsperrung

Im Prinzip befindet sich der Differenzverstärker in der Grundstellung, in der dieser als RS-Flip-Flop (RS-FF) fungieren kann – siehe hierzu die Schaltungsbeispiele des 70015 [2]. Um ein Schalten des RS-FF sicher zu verhindern bzw. in einen Zustand

festzusetzen, kann ein Gatter (AND, OR oder EXOR) verwendet werden.

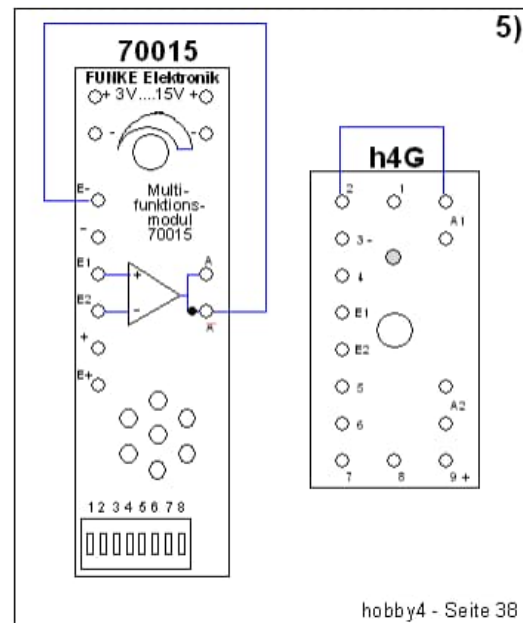


Abb. 12: Selbsthaltung

Die Verbindung in Abb. 12 muss tatsächlich vom invertierenden Ausgang am 70015 erfolgen.

Taktgenerator (Taktgeber) und Tongenerator

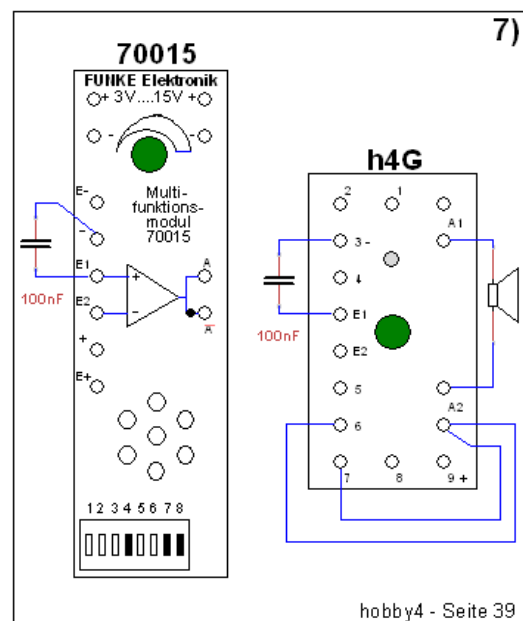


Abb. 13: Tongenerator (hohe Frequenzen)

Um den Differenzverstärker als Taktgenerator (Taktgeber) oder Tongenerator

einzusetzen wird auf jeden Fall ein Kondensator benötigt. Es können auch andere Kondensatorwerte als die Werte in den Abb. 13 und 14 verwendet werden. Beim 70015 stehen sechs verschiedene Kondensatorwerte aus dem mitgelieferten Kondensator-Set zur Auswahl.

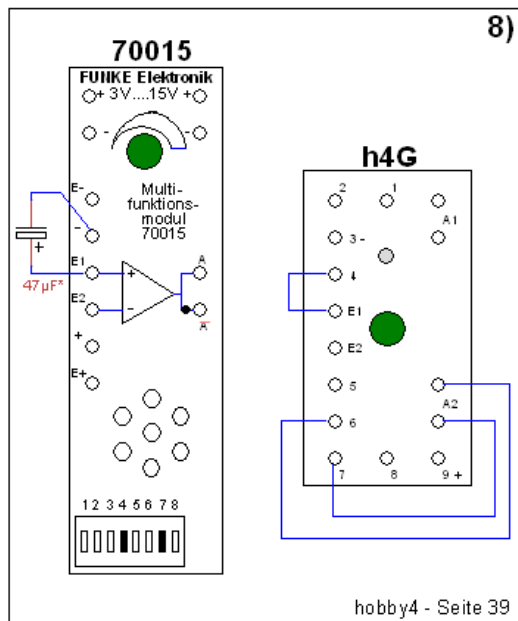


Abb. 14: Taktgenerator (niedrige Frequenzen)

Die Rückkopplung vom invertierten Ausgang zum Drehregler wird durch Einschalten des DIP-Schalters 4 hergestellt und für die Rückkopplung zum Eingang E2 können die DIP-Schalter 5–7 verwendet werden. Der DIP-Schalter 8 dient zur Aktivierung des internen Mini-Lautsprechers des 70015.

Taktgenerator (Taktgeber) gesteuert

Mit diesen beiden Schaltungen kann über den veränderlichen Widerstand der Sensoren die Taktfrequenz verändert werden.

In der Schaltung aus Abb. 15 muss ein Kondensator gewählt und der DIP-Schalter 7 für die Rückkopplung über den 1-k Ω -Widerstand zum Eingang E2 eingeschaltet werden.

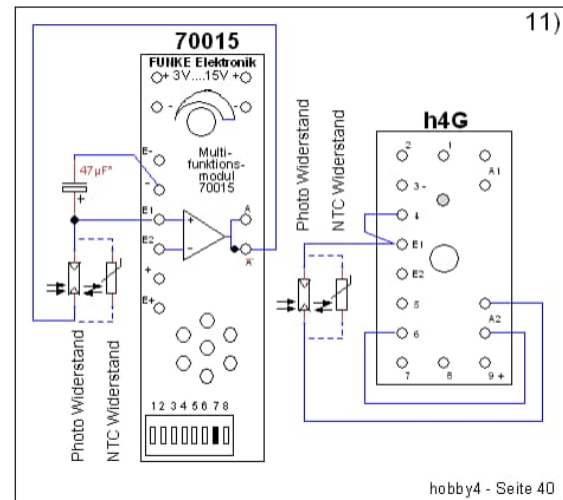


Abb. 15: Taktgenerator gesteuert über Foto- oder NTC-Widerstand

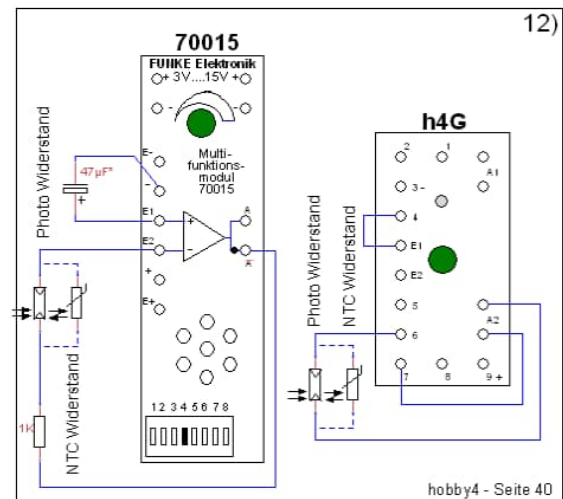


Abb. 16: Taktgenerator gesteuert über Foto- oder NTC-Widerstand und Drehregler

In der Schaltung aus Abb. 16 soll die Rückkopplung über eine Reihenschaltung des Sensors mit dem 1-k Ω -Widerstand erfolgen. Dies kann beim 70015 so nicht realisiert werden; hier muss die Reihenschaltung extern z. B. über ein Experimentiermodul aufgebaut und zwischen dem invertierenden Ausgang und E2 eingebracht werden. Der DIP-Schalter 4 aktiviert den Drehregler, der nun ebenfalls Einfluss auf die Taktfrequenz hat.

Schaltverzögerung für mechanische Schalter

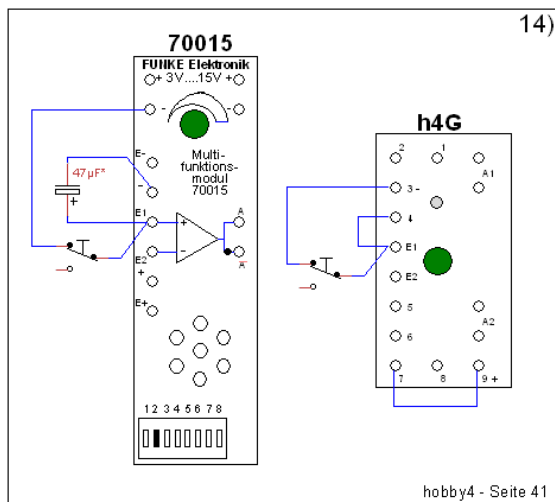


Abb. 17: Verzögerung beim Einschalten eines Tasters/Schalters

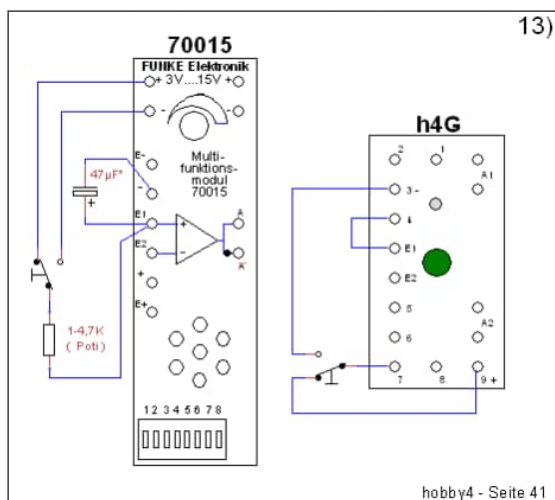


Abb. 18: Verzögerung beim Ein- und Ausschalten eines Tasters/Schalters

Mit dem Einschalten des DIP-Schalters 2 (Abb. 17) wird die Verbindung zwischen Plus und dem Drehregler hergestellt. Es wird wieder ein beliebiger Kondensator aus dem mitgelieferten Set benötigt, der zwischen die Anschlüsse Minus und E1 gesteckt werden muss. Mit dem Drehregler kann nun die Einschaltverzögerung des Schalters reguliert werden.

In der zweiten Schaltung (Abb. 18) wird zwar beim Einschalten eine regulierbare Verzögerung erreicht, aber die Ausschalt-

verzögerung ist sehr kurz. Um diese Schaltung zu realisieren, muss der Schalter in Reihe mit dem Drehregler gebracht werden. Dies ist beim 70015 nicht ohne weiteres möglich. Dazu muss ein separates Poti verwendet werden, z. B. vom Messmodul 70022, oder es wird ein fester Widerstand zwischen 1 k Ω und 4,7 k Ω eingesetzt.

Zusatzfunktionen

Das Multifunktionsmodul ist mit weiteren nützlichen Zusatzfunktionen ausgestattet worden:

- einem Minilautsprecher, der einen separaten Mikrofon-Lautsprecher-Baustein h4MF ersetzt bzw. überflüssig macht (DIP-Schalter 8);
- einem 33-k Ω -Widerstand, der zur Erhöhung der Messsensibilität bei Bedarf zwischen die Eingänge E1 und E2 hinzugeschaltet werden kann (DIP-Schalter 3);
- einem zusätzlichen Rückkopplungswiderstand von 10 k Ω als Ergänzung zu den Rückkopplungswiderständen mit 1 k Ω und 100 k Ω (DIP-Schalter 6).

Darüber hinaus wurde dem Ausgang ein Gegenkopplungsverstärker spendiert, wodurch eine Schaltleistung von bis zu 500 mW (Leistungsgrenze der Transistoren) zur Verfügung steht.

Schlusswort

Ergänzend sollten die Schaltungsbeispielen aus den „Grundlagen der Elektronikmodule“ herangezogen werden, weil hier z. B. gezeigt wird, wie ein Fototransistor am 70015 angeschlossen werden muss. Dieses Dokument enthält alles Wissenswerte über die Elektronik- und Zusatzmodule. Das Schriftstück war bis zur Süd-Convention 2022 noch unvollständig, wurde aber vor kurzem fertig gestellt und kann gerne bei mir angefordert werden [2].

Das Dokument ist um ein Vielfaches erweitert worden mit Informationen, Schaltungen

gen, Bildern und Beispielen zu allen Elektronikmodulen und Zusatzmodulen – zu den veröffentlichten und den nicht veröffentlichten. Aus diesem Grund kann ich nur empfehlen, die alte Version, die noch vor September 2022 verschickt wurde, durch die neue, aktuelle Version zu ersetzen.

Mit diesen Ausführungen hoffe ich nun jeden in die Lage versetzt zu haben, den 70015 als Ersatz für den h4G oder diesen vollumfänglich in seiner Funktion für die eigenen Projekte einzusetzen. Wer weitere Fragen zu den Elektronikmodulen oder zu Schaltungen mit Silberlingen hat, kann diese per E-Mail [2] an mich richten. Gerne bin ich bei der Lösungsfindung und der Umsetzung von Problemstellungen behilflich.

Informationen zu den Elektronikmodulen, den Zusatzmodulen, Laboren und Zubehör sind auch über den Online-Shop von Franz Santjohanser erhältlich [1].

Quellen

- [1] santjohanser: [Spielen. Lernen. Technik](#). Autorisierter fischertechnik-Fachhändler.
- [2] Hans-Christian-Funke: *Grundlagen der Elektronikmodule* (70099). Erhältlich als PDF auf Anfrage beim Autor: elektronik@funke4you.de
- [3] Fischer-Werke: [fischertechnik hobby 4](#) (35804), 1971.
- [4] Fischer-Werke: [Leitfaden für die fischertechnik-Elektronik-Bausteine](#) (39504), 1976.

Elektronik

Silberlinge: Original oder Nachbau (Teil 9)

Peter Krijnen

In diesem letzten Beitrag in der Reihe der Silberlinge werde ich mich mit den Modulen aus hobbylabor 1 und dem Mikrofon-Lautsprecher-Baustein befassen. Abschließend möchte ich dann noch meine Nachbauten vorstellen.

hobbylabor 1

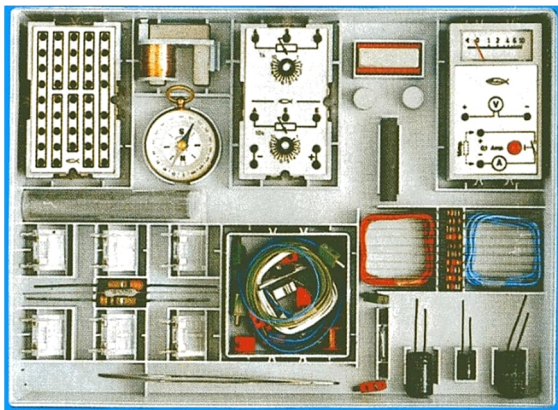


Abb. 268: hobbylabor 1 [30626/30850](#)

hobbylabor 1 wurde 1974 als Neuheit auf der Nürnberger Spielwarenmesse vorgestellt. Diese Kasten erhielt die Artikelnummer [30626](#). Warum diese Nummer 1975 auf [30850](#) geändert wurde, ist mir nicht klar.

In einem Katalog 1975/1976 wird auch angekündigt, dass ein hobbylabor 2 „Halbleiterschaltungen“ in Vorbereitung sei. Es blieb bei der Ansage; leider wurde dieser Kasten nie produziert. 1979 wurden jedoch die Kästen „Elektronik-Praktikum“ ([30629](#)) und „IC-Digital-Praktikum“ ([30630](#)) eingeführt. Mit beiden Kästen lassen sich „Halbleiterschaltungen“ bauen. Beide wurden 1981 durch „Elektronik“ ([30253](#)) ersetzt. Auf die Module in den Kästen [30630](#) und [30253](#) gehe ich in einem späteren Beitrag ein. 1980 endete das hobbylabor 1.

Ich selbst besitze kein hobbylabor 1. In einer großen Menge fischertechnik, die ich mal bei einer Schule gekauft habe, waren die Module aus diesem Kasten enthalten.

Neben einem „Potentiometer-Baustein“ PB ([37158](#), Abb. 269) und einem „Experimentierfeld“ EF ([37140](#), Abb. 273) befand sich auch ein „Volt-Ampere-Meter“ ([37142](#)) im Karton. Diese wurden auch separat verkauft.

Beide Module aus hobbylabor 1 scheinen die gleichen zu sein wie die ersten Module aus hobby 4. Es wurden jedoch drei Änderungen vorgenommen:

- Erstens sind die Anschlussbuchsen in einer 7,5×7,5 mm-Matrix (9×5 Buchsen) anstelle der 12×8 mm-Matrix (8×3 Buchsen) angeordnet.
- Zweitens bestehen die Anschlussbuchsen nicht mehr aus Bundhülsen, sondern aus gerolltem Blechmaterial (Abb. 270).
- Da die Enden offen und flach sind, können die Anschlussbuchsen nicht mehr auf die Platine gelötet werden (Abb. 274).

Der Vorteil der flachen Enden der Anschlussbuchsen liegt darin, dass man auch die Anschlussdrähte eines Widerstands, Kondensators oder einer Diode direkt in die Buchsen stecken kann, ohne vorher Stecker aufschrauben zu müssen.

Potentiometer-Modul PB

Zwei Potentiometer sind im Potentiometer-Baustein eingebaut: Eines mit 1 kΩ und eines mit 10 kΩ. Das 1K-Potentiometer hat noch einen Vorwiderstand von 150 Ω und das 10K einen von 1 kΩ. Außerdem gibt es je zwei Anschlüsse für „+“ und „-“. Ich gehe davon aus, dass das „A“ für Ausgang und das „E“ für Eingang steht. Warum sie gewechselt wurden, ist mir nicht klar. „S“ steht dann für „Schleifer“. Wie das alles funktioniert ist auf dem Schaltbild in Abb. 271 zu sehen. Wie es unter dem Deckel aussieht zeigt Abb. 270, das Layout Abb. 272.

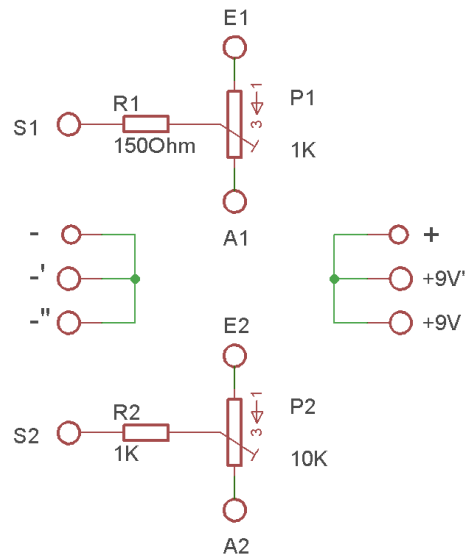


Abb. 271: Schaltbild des PB

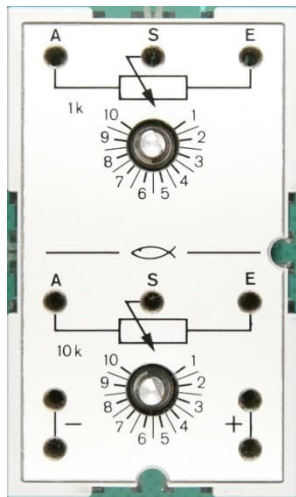


Abb. 269: Potentiometer-Baustein PB (37158)

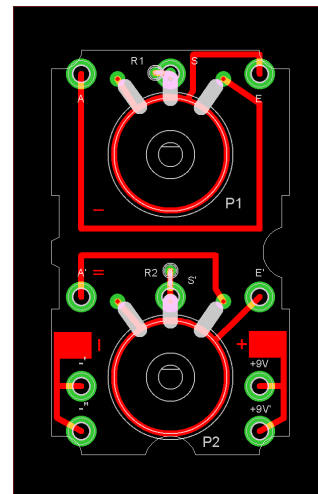


Abb. 272: Layout des PB



Abb. 270: Leiterbahnseite des PB

Experimentierfeld EF

Das Experimentierfeld besteht aus fünf Reihen oder Spalten (je nach Betrachtungsweise) mit jeweils neun Buchsen. Da in den Modulen Platz benötigt wird, um die Module untereinander zu verbinden, gehen unten und auf der rechten Seite je ein Anschluss verloren.

Insgesamt stehen also 43 Anschlussbuchsen verteilt auf sieben Leisten zur Verfügung: 1 von 9, 1 von 8, 2 von 5 und 4 von 4. Es gibt keine interne Verbindung mit „+“ oder „-“. Abb. 274 gibt einen Blick auf die Unterseite des Deckels, Abb. 275 zeigt das Layout.



Abb. 273: Experimentierfeld EF (37140)



Abb. 274: Leiterbahnseite des EF

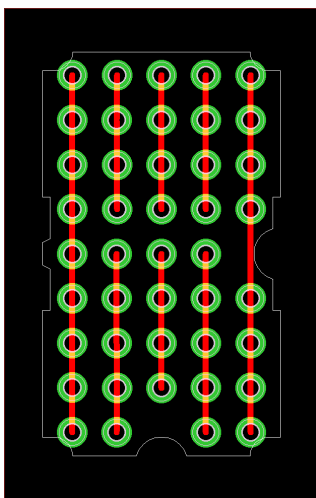


Abb. 275: Layout des EF

Mikrofon-Lautsprecher-Modul ML

Der Mikrofon-Lautsprecher verfügt über ein „Kristall-System“. Dieses System wandelt einerseits elektrische Energie in akustische Schwingungen um und umgekehrt akustische Schwingungen in elektrische Energie [1]. Dies wird in [2] ab Seite 32 weiter erklärt. In Abb. 277 ist zu erkennen, dass die vier Anschlussbuchsen untereinander vernetzt sind. Es ist klar, dass dieser Baustein jeweils nur eine Funktion handhaben kann. Laut [1] liegt der Frequenzbereich bei 50 – 7000 Hz.

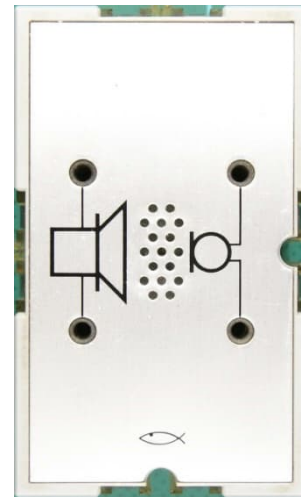


Abb. 276: Mikrofon-Lautsprecher

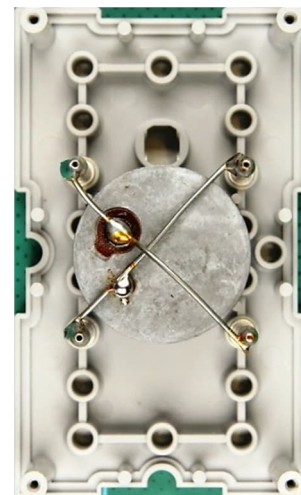


Abb. 277: Unterseite des Deckels des ML

Nachbau

Zum Nachbau des Potentiometer-Bausteins verwende ich auch meine universellen Leiterplatten. Die 45×75-Version ist auf Abb. 278 zu sehen. Im Vergleich zum Original habe ich die „+“- und „-“-Verbindungen in eine logischere Position gebracht. Ich habe auch die Anschlüsse der Potentiometer verschoben. Da an „+“ oder „-“ keine Bauteile angeschlossen sind, könnte man die Platine auch um 180° drehen. Die „S“-Anschlüsse befinden sich dann auf der rechten Seite. Es spielt keine Rolle, welchen Anschluss der Potentiometer man „A“ oder „E“ nennen möchte. In Abb. 279 sehen wir die Komponentenanzordnung und in Abb. 280 das Layout.

Wir sind nicht verpflichtet, uns beim Nachbauen an das Original zu halten. Das heißt, wir können jeden gewünschten Wert für die Potentiometer nehmen. Für ein Projekt, an dem ich arbeite, brauchte ich zwei 10K-Potentiometer – dachte ich. 10K erwiesen sich aber als zu niedrig. Also habe ich ein drittes Modul gebaut, aber jetzt mit zwei 25K-Potentiometern.

Die Abb. 281 bis 283 geben einen Blick auf die 60×60-Version, Abb. 284 bis 286 die 30×60-Version.

Für den Nachbau des Experimentierfeldes empfehle ich das in Abb. 275 gezeigte Layout. Dies habe ich jedoch nicht genutzt. Ich denke, dass der experimentelle Bereich in Bezug auf die Möglichkeiten etwas zu begrenzt ist. Wie in Abb. 287 und 288 verwende ich ein kleines Steckbrett [3]. So kann ich auch ICs verwenden. Auch Widerstände, Kondensatoren und Dioden haben einen besseren Kontakt.

Ich sah keinen Sinn darin, den Mikrofon-Lautsprecher-Baustein neu zu erstellen. Erstens, weil ich ihn so gut wie nie benutze, und wenn, dann nur noch als Lautsprecher. Auch ein „Kristall-System“ kann ich nirgendwo finden.

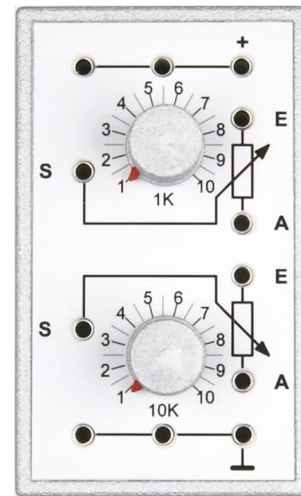


Abb. 278: Nachbau 2: Frontplatte des PB für das 45×75-Gehäuse

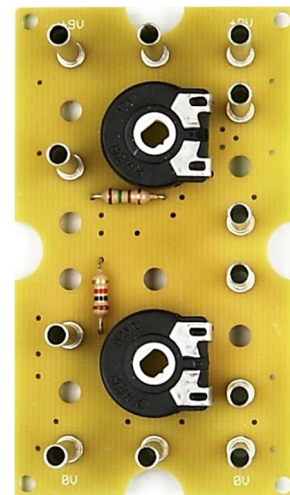


Abb. 279: Nachbau 2: Platine des PB für das 45×75-Gehäuse

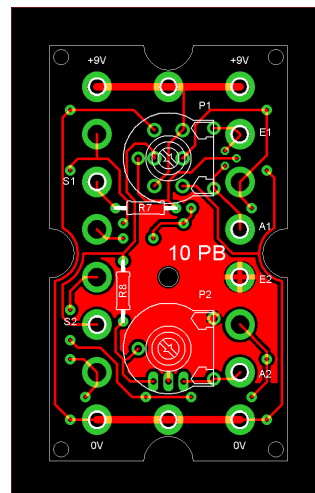


Abb. 280: Nachbau 2: Layout des PB für mein 45×75-Gehäuse

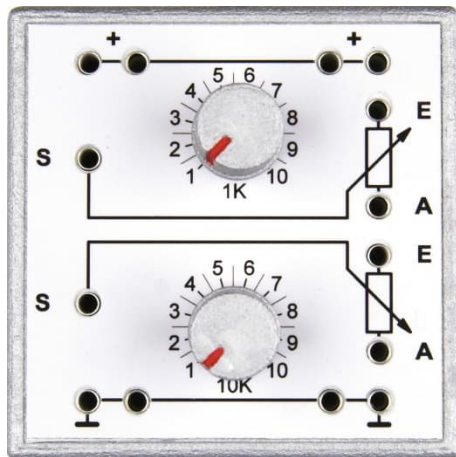


Abb. 281: Nachbau 3: Frontplatte des PB für das 60×60-Gehäuse

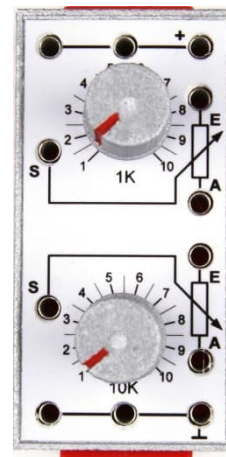


Abb. 284: Nachbau 4: Frontplatte des PB für das Batteriegehäuse ([32263](#))

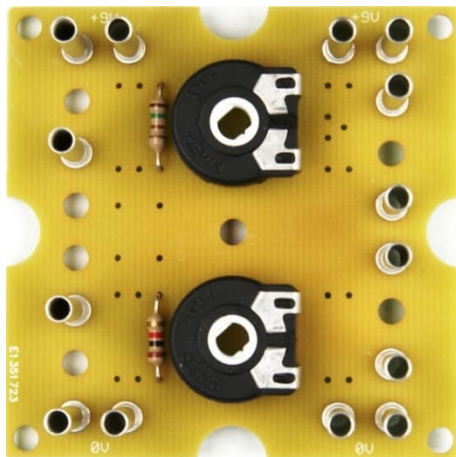


Abb. 282: Nachbau 3: Platine des PB für das 60×60-Gehäuse

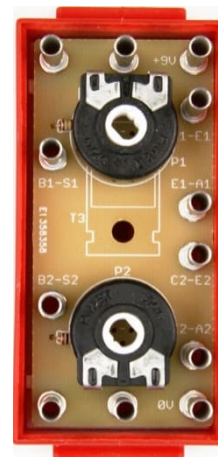


Abb. 285: Nachbau 4: Platine des PB für das 30×60-Gehäuse

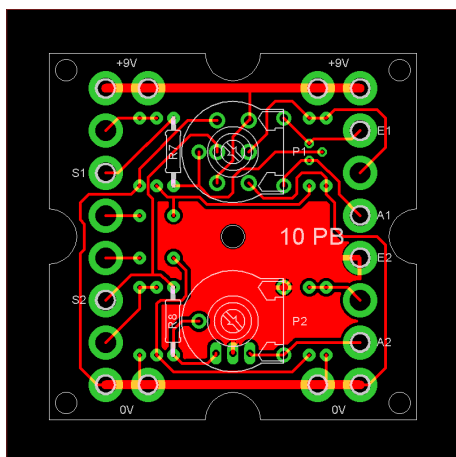


Abb. 283: Nachbau 3: Layout des PB für die 60er-Kassette ([32076](#))

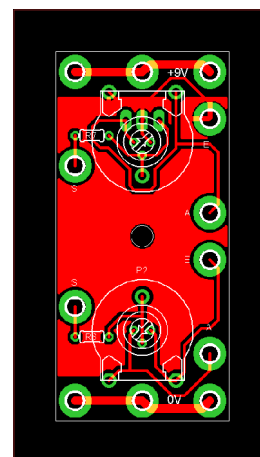


Abb. 286: Nachbau 4: Layout des PB für das Batteriegehäuse ([32263](#))

Alternativ könnte man den Lautsprecher (Abb. 289) von „Elektronik“ ([30253](#)) verwenden. Eine weitere Alternative könnten die kleinen Lautsprecher sein, die im Modellbau zum Beispiel für den Sound von Lokomotiven verwendet werden. Ob das in Sachen Widerstand und Leistung mit den Ausgängen der Silberlinge kompatibel ist, muss man jedoch prüfen.



Abb. 287: Nachbau 2: Ein Breadboard als Alternative zum Experimentierfeld für das 45×75-Gehäuse

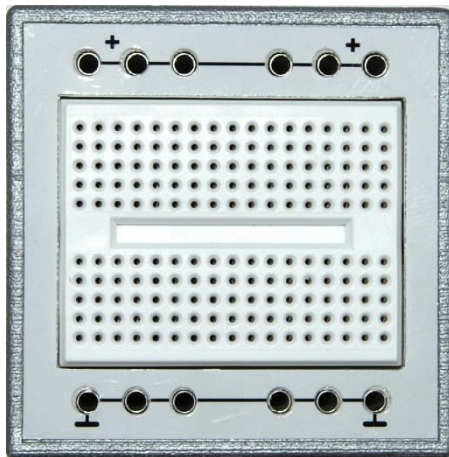


Abb. 288: Nachbau 2: Ein Breadboard als Alternative zum Experimentierfeld für das 60×60-Gehäuse



Abb. 289: Lautsprecher ([36936](#)) aus dem Elektronikkasten ([30253](#))

Ist dies das Ende meiner Besprechungen über die Silberlinge? Was die Silberlinge betrifft: Ja. Hinsichtlich der Module: Nein. Es gibt nämlich noch die Module des „IC-Digital-Praktikums“ ([30630](#)) und die der „Elektronik“ ([30253](#)).

Quellen

- [1] Fischer-Werke: [fischertechnik h4 ML Mikrofon-Lautsprecher](#) (39586).
- [2] Fischer-Werke: [fischertechnik hobby 4 Band 2](#) (39542), 5/1973.
- [3] Conrad Electronic SE: [TRU COMPONENTS Steckplatine Weiß Polzahl gesamt 170 \(L x B x H\) 45.72 x 35.56 x 9.40 mm.](#)

Computing

Einführung in ftScratch (4): Der Morsetelegraf

Dirk Fox

Scratch ist eine für Einsteiger und Schulen besonders geeignete Programmiersprache. Die Entwicklungsumgebung ftScratch3 unterstützt mit entsprechenden Erweiterungen die fischertechnik-Controller TXT und BT Smart [1]. In dieser Serie führen wir in die Programmierung mit ftScratch ein – mit kleinen Modellen und Aufgaben.

Der Morsetelegraf

Die Erfindung des Morsetelegrafen ist ein Meilenstein der Technikgeschichte [2, 3]: Er markiert den Beginn der Echtzeit-Kommunikation über große Distanzen. Zwar spielt im Zeitalter des Internet die Nachrichtenübermittlung mit Morsezeichen keine nennenswerte Rolle mehr. Dennoch ist das Morsen auch ein Meilenstein der Informationstechnik: Der von *Alfred Vail* (1807-1859) im Jahr 1838 entwickelte Morse-Code ist der erste weltweit standardisierte Zeichencode für den Nachrichtenaustausch. Er ist zugleich sehr effizient, da die Länge des Morsezeichens eines Buchstabens davon abhängt, wie häufig dieser in der

englischen Sprache auftritt: Die Morsezeichen für häufige Buchstaben (wie „e“ oder „t“) sind sehr kurz, die für seltenere Buchstaben (wie „x“ oder „y“) um ein Vielfaches länger, wie man am Code-Baum sehr schön erkennen kann (Abb. 1).

Modelle von Morsetelegrafen finden sich schon im Jahr 1975 in fischertechnik-Anleitungen – als Club-Modell und in der Anleitung zum Baukasten Elektromechanik [4, 5]. Unser Morsetelegraf kommt mit weniger Mechanik aus – für ihn benötigen wir neben dem BT Smart oder TXT Controller nur einen fischertechnik-Taster oder einen einfachen „Zeichengeber“ (Abb. 2, 3), eine Linse-LED und eine Fotodiode.

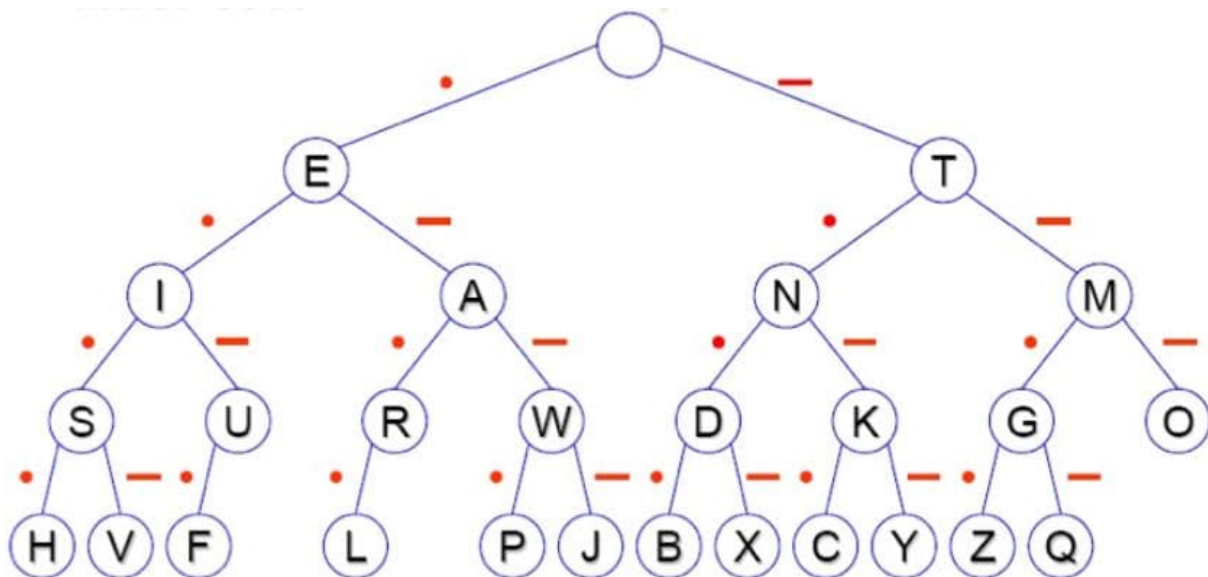


Abb. 1: Morsealphabet

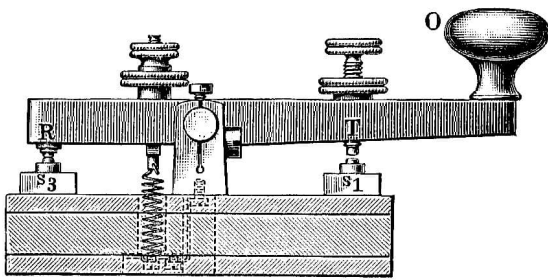


Abb. 2: Morsezeichen-Geber
(aus: Meyers Konversationslexikon, 1909)

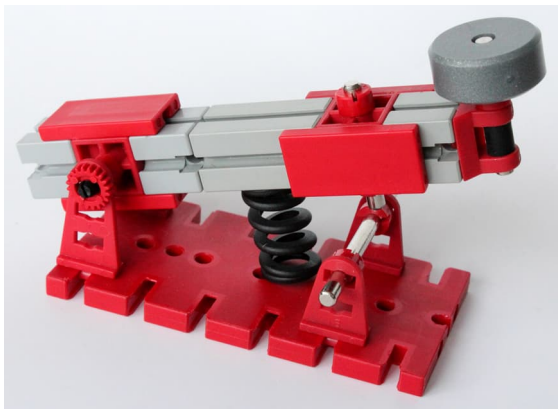


Abb. 3: Morsezeichen-Geber [3]

Mit den folgenden Aufgaben tasten wir uns Schritt für Schritt an die Funktionsweise der Datenkommunikation heran: Wir beschäftigen uns mit Kodierung, lernen Kommunikationskanäle kennen und berechnen deren Bandbreite. Als „Krönung“ programmieren wir ein kleines Chat-Programm – auf Morse-Basis.

Sensoren und Aktoren

Die Aufgaben können sowohl mit dem TXT als auch mit dem BT Smart Controller gelöst werden. Dafür benötigen wir

- eine Fotodiode
- eine Linse-LED (oder Linse-Lampe)
- eine weitere LED zur Signalanzeige

Die Sensoren und Aktoren werden wie folgt angeschlossen (jeweils Polung beachten):

- I1: Fotodiode
- M1: Linse-LED
- M2: Anzeige-LED

Aufgaben

Für die Übertragung von Informationen benötigen wir zweierlei: Erstens einen Code zur Darstellung der Informationen, auf den sich Sender und Empfänger zuvor einigen müssen (in unserem Fall den international standardisierten Morsecode), und zweitens einen Kommunikationskanal, über den wir die kodierten Daten übertragen können.

Als Kommunikationskanal kommen grundsätzlich verschiedene Möglichkeiten in Betracht: Für die Übertragung über große Distanzen eignen sich elektromagnetische Wellen besonders gut, für kürzere Distanzen Schallwellen (Töne), Licht (bspw. Infrarot oder sichtbares Licht) und Spannung (kabelgebundene Übertragung).

Wir verwenden in unserem Modell eine Übertragung mit Licht. Im Modell der Knickschranke [6], beim Münzzähler [7] und beim Barcodeleser [8] hast du eine Linse-LED und eine Fotodiode als Lichtschranke eingesetzt. Eine solche „Licht-Strecke“ werden wir nun zur Übertragung von Signalen verwenden.

Aufgabe 1

Mit der Linse-LED soll ein Lichtsignal ausgesendet werden. Schließe die Linse-LED dazu zunächst an 9V/GND an und schalte den Morse-Geber dazwischen, so dass die LED leuchtet, wenn der Taster gedrückt ist. Richte die Linse-LED auf die Fotodiode (Abb. 4).

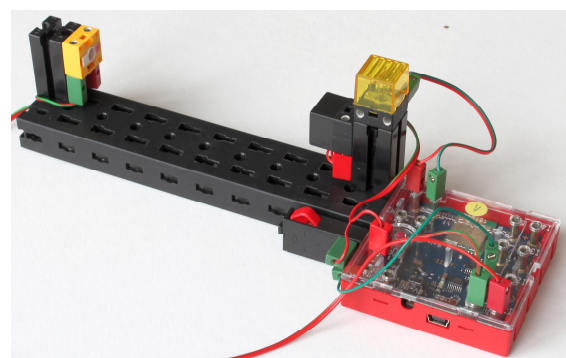


Abb. 4: Sender und Empfänger

Empfängt die Fotodiode das Lichtsignal, soll die Anzeige-LED (an M2) eingeschaltet werden. Die empfangenen Lichtsignale sollen zusätzlich gezählt und auf der „Bühne“ angezeigt werden. Schreibe ein entsprechendes Scratch-Skript und experimentiere mit dem Geber. Was kannst du beobachten?

Aufgabe 2

Nun wollen wir uns zunächst auf das Senden konzentrieren. Verbinde dazu die Linsen-LED (ohne zwischengeschalteten Geber) direkt mit dem Motorausgang M1.

Für die Darstellung der zu sendenden Signale verwenden wir einen simplen Binär-code aus kurzen und langen Signalen: Ein „-“ steht für ein kurzes Signal („dit“) und ein „-“ für ein langes Signal („dah“). Eine Signalfolge kodieren wir also als eine Zeichenfolge aus Punkten und Strichen.

Damit der Empfänger kurze und lange Signale unterscheiden kann, benötigen wir neben einer Pause zwischen zwei aufeinander folgenden Signalen außerdem einen „Takt“. Den geben wir durch die Länge der Signale und Pausen vor: Ein „dah“ soll dreimal so lang sein wie ein „dit“, die Pause zwischen zwei Signalen soll der Dauer eines „dit“ entsprechen. Genau dies sind die Eigenschaften des Morsecodes.

Als Basiszeiteinheit, die der Länge eines „dit“ entspricht, legen wir 300 Millisekunden (0,3 Sekunden) fest.

Schreibe nun ein Scratch-Skript, das die Signalfolge `-.-.---.---.---.---` mit der Linsen-LED an die Fotodiode übermittelt, und lass‘ dir das Empfangsergebnis auf der „Bühne“ anzeigen. (*Tip*: Arbeite mit Unterprogrammen und vergiss‘ die Pausen zwischen den Signalen nicht!)

Aufgabe 3

Für unseren Morsetelegrafen kodieren wir jetzt das gesamte Morsealphabet als Punkt-Strich-Zeichenfolgen und speichern es in

einer Liste. Um euch die Arbeit des Abtippens zu ersparen, könnt ihr dafür die Textdatei „Morsealphabet.txt“ aus dem [Download-Bereich](#) zu dieser Ausgabe der ft:pedia in eine Listenvariable laden: Lasst euch die Listenvariable „Morsealphabet“ auf der „Bühne“ anzeigen und öffnet über die rechte Maustaste die txt-Datei zum Import der Daten.

Passen dein Programm aus Aufgabe 2 nun so an, dass es nacheinander alle 26 Zeichen des Morsealphabets sendet. Das Ende eines Morsezeichens wird dabei durch eine Pause der Länge von insgesamt drei „dit“ signalisiert.

Aufgabe 4

Nun soll dein Skript lernen, eine beliebige Buchstabenfolge zu morsen. Dazu benötigst du eine zweite Listenvariable mit den Buchstaben des Alphabets. Auch die kannst du aus dem [Download-Bereich](#) als Textdatei herunterladen und sie per Mausklick in die Listenvariable „Alphabet“ übernehmen („Alphabet.txt“).

Erweitere dein Skript aus Aufgabe 4 nun so, dass es eine beliebige Buchstabenfolge morst und teste es mit dem folgenden Pan-gram (= einem Satz, der alle Buchstaben des Alphabets mindestens einmal enthält): „thequickbrownfoxjumpsoverthelazydog“.

Aufgabe 5

Jetzt wenden wir uns dem Empfang der Morsezeichen zu. Während die Linsen-LED sendet, soll die Fotodiode die gesendeten Zeichen erkennen und anzeigen. Das Empfangsprogramm kannst du entweder als einen parallelen Task programmieren oder, wenn dir ein zweiter Rechner zur Verfügung steht, auf einem separaten „Empfangsgerät“.

Beginne zunächst mit einem Empfangs-Thread, der die in der Lösung zu Aufgabe 2 gesendeten Signale empfängt, jedes empfangene Signal über die LED anzeigt und

die Signale als Punkt-Strich-Zeichenfolge auf der „Bühne“ ausgibt.

Aufgabe 6

Nun soll der Empfänger-Prozess die 26 Zeichen des Morse-Alphabets empfangen, dekodieren und auf der „Bühne“ ausgeben können, die mit deiner Lösung zu Aufgabe 3 gesendet werden.

Aufgabe 7

Jetzt können wir einzelne (Morse-) Zeichen separat empfangen – und es ist nun kein großer Schritt mehr zu dem Empfang beliebiger Morse-Nachrichten.

Erweitere dein Empfangs-Skript aus Aufgabe 6 zu einem solchen Morsezeichen-Empfänger und teste ihn mit dem Pangram, das deine Lösung zu Aufgabe 4 aussendet.

Du wirst bemerken, dass ein zuverlässiger Empfang aller gesendeten Zeichen gar nicht so einfach zu erreichen ist. Das hat mehrere Ursachen:

- Die Kommunikation mit dem Controller zur Abfrage der Inputs und Ansteuerung der Outputs kostet Zeit (siehe Aufgabe 1)
- Scratch schaltet nur in Wartezeiten zwischen parallelen Threads hin und her
- Die Fotodiode kann (abhängig von Störlicht und der Entfernung von der Linsen-LED) fehlerhaft „schalten“

Alle drei Punkte erschweren die Synchronisation von Sender und Empfänger. Dennoch wirst du feststellen, dass sich Sender und Empfänger nach einem Übertragungsfehler wieder von selbst synchronisieren, sobald der Empfänger eine Pause zwischen zwei Zeichen erkennt.

Aufgabe 8

Bisher hast du den zu übermittelnden Text fest vorgegeben. Erweitere deinen Sendethread jetzt noch um eine Texteingabe über die Tastatur: Damit wird aus deinem Morse-telegraf ein vollständiger Chat-Client.

Wenn du die Eingabe besonders komfortabel gestalten willst, ergänze die Texteingabe um die Möglichkeit, eine Fehleingabe zu korrigieren.

Aufgabe 9

Für Scratch gibt es eine Erweiterung (Extension), die die Übersetzung eines Textes in eine andere Sprache vornimmt (Abb. 5).

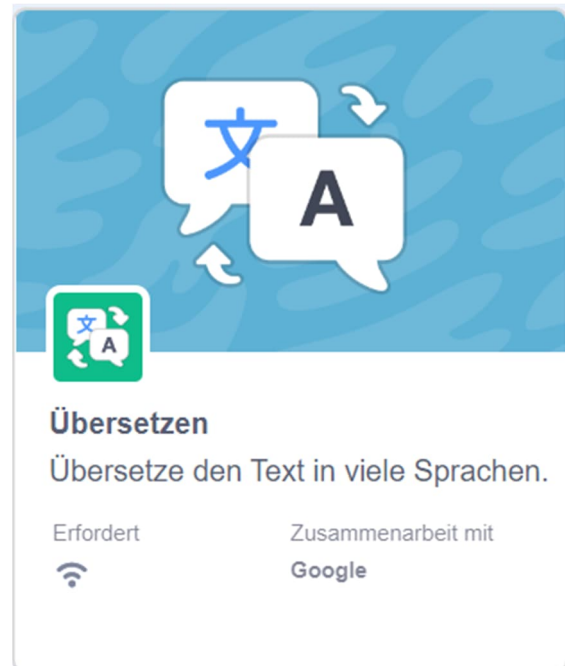


Abb. 5: Erweiterung „Übersetzen“

Binde diese Erweiterung ein und erweitere dein Skript aus der Lösung zu Aufgabe 8 so, dass ein zu übertragender Text in eine andere Sprache – bspw. ins Englische – übersetzt wird, bevor Scratch ihn morst.

Aufgabe 10

Die Geschwindigkeit eines Morsesenders wird meist in „Wörtern pro Minute“ (WpM) angegeben. Dabei wird das Wort „Paris“ (inklusive Wortende-Signal) als Maß verwendet. Wie schnell ist dein Morse-Sender/-Empfänger in WpM?

Lösungsbeispiele

Die folgenden Lösungsbeispiele wurden für den BT Smart Controller programmiert. Die entsprechenden Scratch-Skripte und alle Hilfsdateien findet ihr im [Download-Bereich](#) zu dieser Ausgabe der ft:pedia. Die Anpassung der Skripte an den TXT Controller (Ersetzung der entsprechenden Blöcke in ftScratch) ist sehr einfach.

Aufgabe 1

Den Signalzähler kannst du als parallelen Thread realisieren: Verwende dafür das Schließen der „Lichtschranke“ als Auslöser (*Trigger*).

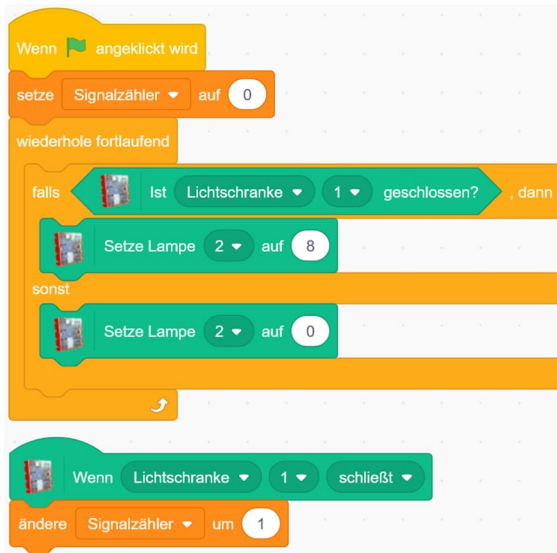


Abb. 6: Empfangen von Lichtsignalen

Du erkennst, dass die Anzeige-LED etwas verzögert reagiert. Diese Verzögerung wird vor allem durch die Abfrage der Eingänge des Controllers über die USB- oder BLE-Verbindung verursacht. Daher dürfen die Signale nicht zu kurz sein, da sie sonst von der Fotodiode nicht erkannt werden.

Aufgabe 2

Die Aussendung der langen und kurzen Signale „dit“ und „dah“ können wir in je einen eigenen Block auslagern; damit wird das Programm übersichtlicher (Abb. 7). Im Hauptprogramm wird jetzt nur noch die vorgegebene Zeichenfolge nacheinander ausgesendet (Abb. 8).

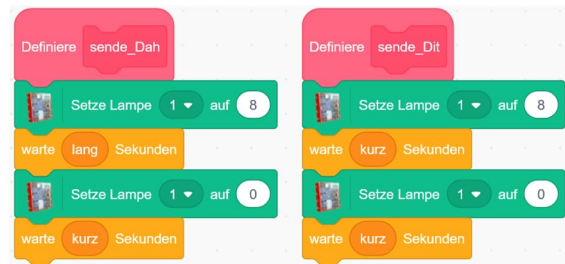


Abb. 7: Senden der kurzen und langen Signale

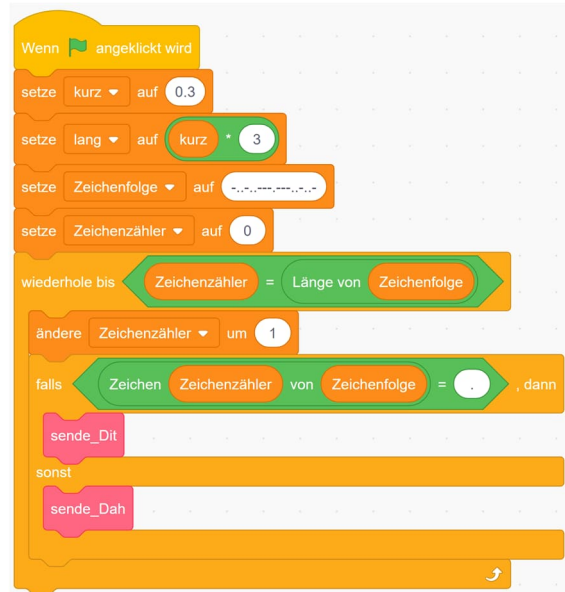


Abb. 8: Schleife zur Aussendung der Signale

Aufgabe 3

Die Sendefunktion aus dem Lösungsbeispiel von Aufgabe 2 kann man in einen Block auslagern, der ein einzelnes Morsezeichen sendet (Abb. 9).

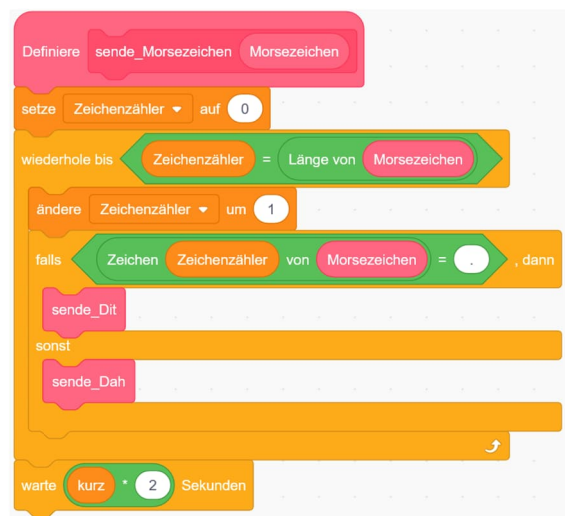


Abb. 9: Senden eines Morsezeichens

Damit beschränkt sich das Hauptprogramm auf eine Schleife, die ein Morsezeichen nach dem anderen aus dem Morsealphabet auswählt und an den Sende-Block übergibt (Abb. 10).



Abb. 10: Sendeschleife

Aufgabe 4

Für die Ausgabe eines ganzen Textes müssen wir nur wenige Anpassungen an dem Skript aus Aufgabe 3 vornehmen: Aus dem Hauptprogramm wird ein Buchstabe nach dem anderen aus dem vorgegebenen Text (statt eines Elements aus der Liste des Morsealphabets) gesendet (Abb. 11).

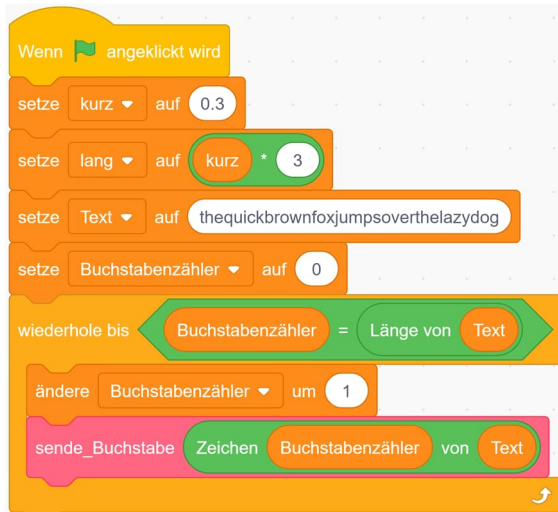


Abb. 11: Senden der Buchstaben eines Textes

Mit einer einzigen raffinierten Programmzeile wählen wir das zu dem jeweiligen

Buchstaben gehörige Morsezeichen aus: Wir bestimmen die Position des Buchstabens im Alphabet und nehmen diese als Index für die Auswahl des entsprechenden Morsezeichens aus dem Morsealphabet (Abb. 12).

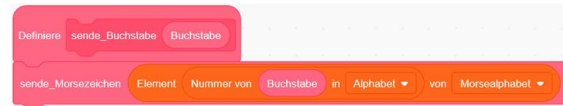


Abb. 12: Selektion des Morsezeichens

Aufgabe 5

Als Trigger für den Empfang eines Signals verwenden wir wieder das Schließen der Lichtschranke. Anschließend genügt es, etwas länger als ein „dit“ zu warten, um festzustellen, ob ein „-“ (Länge: drei „dit“) oder ein „.“ gesendet wurde (Abb. 13).⁸



Abb. 13: Empfangs-Thread

Aufgabe 6

Die Auswertung eines einzelnen Signals können wir aus der Lösung zu Aufgabe 5 übernehmen und als eigenen Block definieren (Abb. 14).

Um mehrere Morsezeichen empfangen zu können, darf der Empfangsprozess nicht mit einem Trigger gestartet werden – anderenfalls wird bei jedem Signal ein neuer Thread gestartet.

⁸ Wenn du statt einer Linsen-LED eine Linsen-Lampe verwendest, musst du die Wartezeit etwas länger wählen, da die Lampe „nachglüht“.

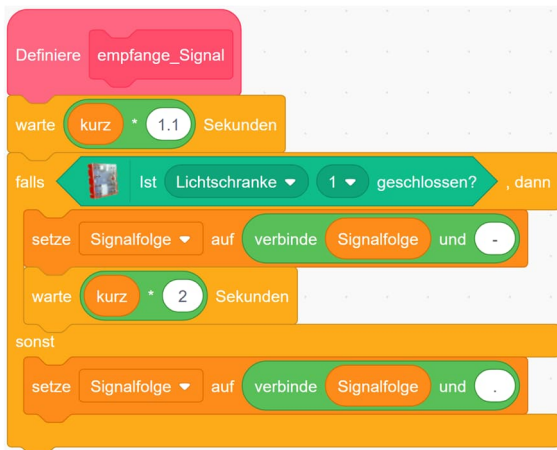


Abb. 14: Auswertung eines einzelnen Signals

Außerdem müssen wir die Zeit zwischen zwei Signalen messen, um festzustellen, ob eine empfangene Zeichenfolge bereits beendet ist.

Das machen wir, indem wir prüfen, ob direkt auf das Pausenzeichen nach einem Signal ein weiteres Signal oder eine längere Pause (von insgesamt drei „dit“ Dauer) folgt. Dafür „tasten“ wir den Signalverlauf etwas verzögert ab. Abb. 15 zeigt schematisch den Signalverlauf beim Sender und die Abtastung beim Empfänger.

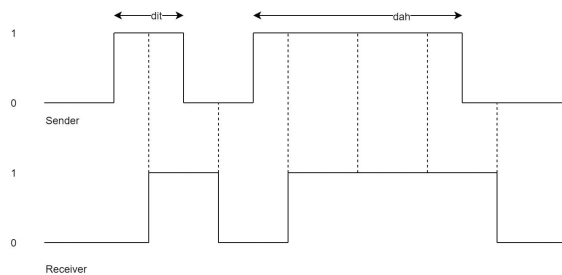


Abb. 15: Signalverlauf beim Sender, etwas verzögerte Abtastung durch den Empfänger

Mit der Länge der Verzögerung müsst ihr vielleicht ein wenig experimentieren: Da auch die anderen Operationen etwas Zeit benötigen, darf die Anfangsverzögerung nicht zu lang gewählt werden, sonst könnte es passieren, dass ein kurzes „-“-Signal übersprungen wird.

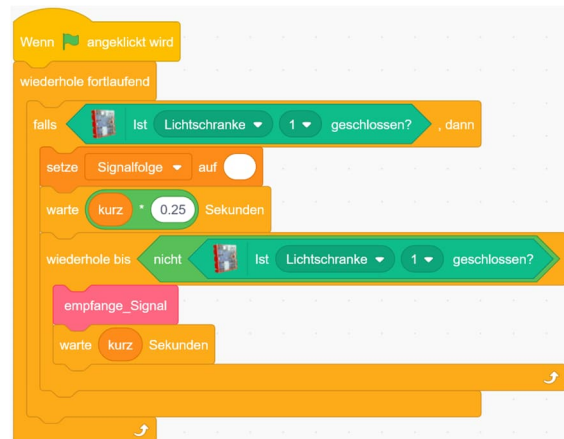


Abb. 16: Empfangs-Thread für mehrere Zeichen (Signalfolgen)

Aufgabe 7

Wieder kann die Schleife aus dem Hauptprogramm in einen Block verschoben werden (Abb. 17). Die Verzögerungszeit am Anfang des Blocks muss ggf. etwas angepasst werden, damit keine Signale verloren gehen oder einem falschen Morsezeichen zugeordnet werden.

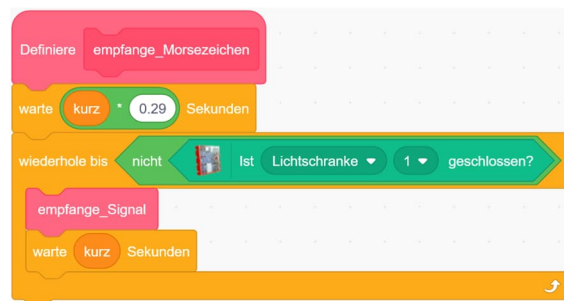


Abb. 17: Empfang eines Morsezeichens als Block

Im Empfangs-Thread erfolgt dann die Dekodierung des Morsezeichens zu einem Buchstaben des empfangenen Textes ähnlich wie im Sende-Thread (Abb. 18).

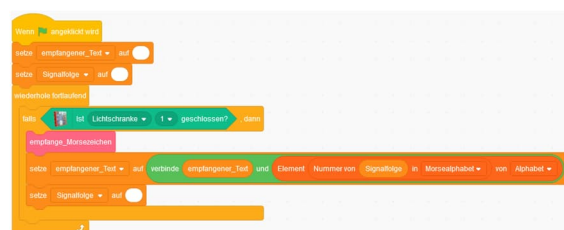


Abb. 18: Empfang und Dekodierung eines längeren Textes aus Morsezeichen

Aufgabe 8

Die Texteingabe kannst du in einen Block auslagern, der die Tastatur abfragt und den Buchstaben der jeweils gedrückten Taste in eine Text-Variable schreibt (Abb. 19).

Zwei Punkte musst du dabei beachten:

- du musst eine „Abschluss“-Taste festlegen, mit der die Eingabe beendet werden kann (im Lösungsbeispiel in Abb. 20 ist das die Leertaste), und
- du musst nach jedem Tastendruck eine kurze Pause vorsehen, um die Taste zu entprellen.

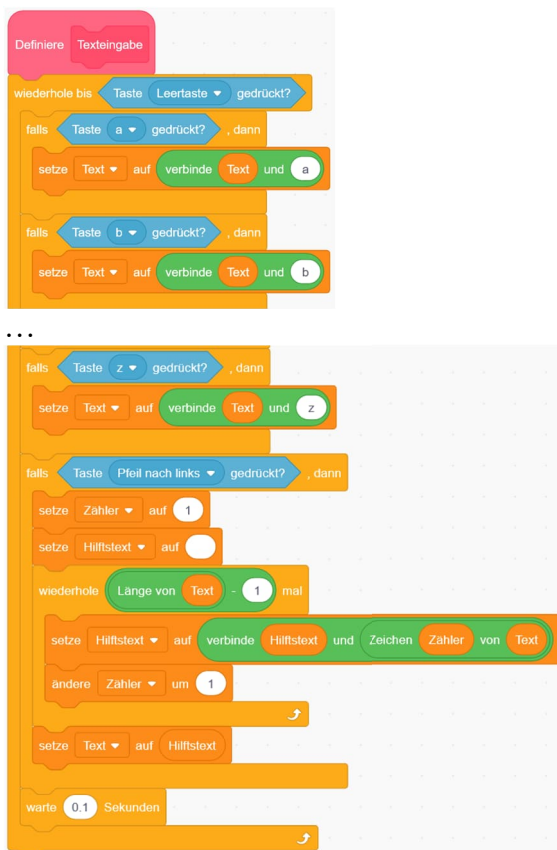


Abb. 19: Tastatureingabe der Nachricht

Etwas knifflig ist die Implementierung einer „Zurück“-Taste zum Löschen des letzten Zeichens (im Lösungsbeispiel in Abb. 19 ist das der Pfeil nach links).

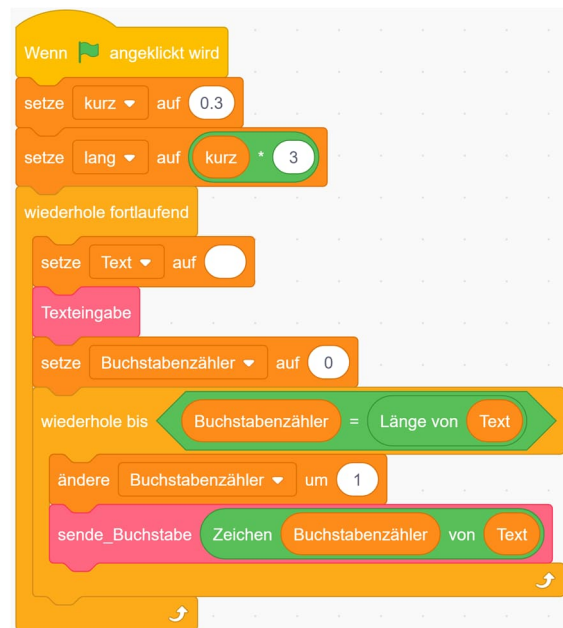


Abb. 20: Textkommunikation

In Scratch kann man das nur durch ein Kopieren der einzelnen Zeichen aus dem eingegebenen String umsetzen.

Mit einer geringen Anpassung wird so aus dem Sende-Thread aus der Lösung zu Aufgabe 7 ein einfacher Chat-Client (Abb. 20).

Aufgabe 9

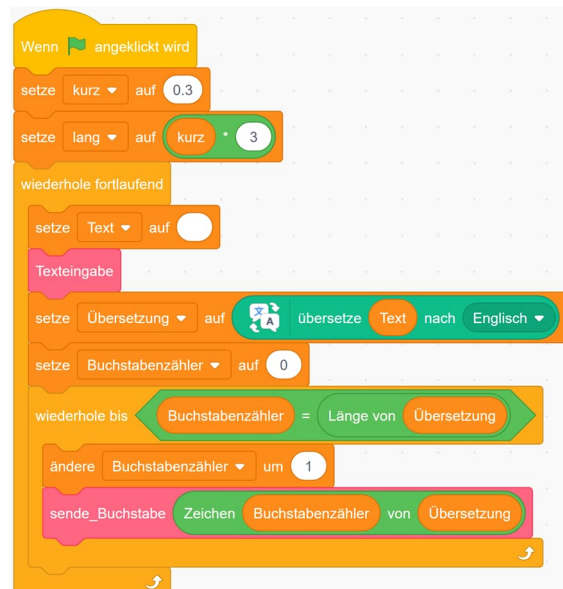


Abb. 21: Übersetzung des Textes

Die Erweiterung der Lösung aus Aufgabe 8 ist minimal.

Aufgabe 10

Der Morsecode von „Paris“ besteht aus 50 „dit“. Somit erreicht unser Morse-Sender mit „dit“ = 0,3 s gerade einmal 4 WpM. Das entspricht einem Drittel der Morsegeschwindigkeit, die Funkamateure in ihrer Prüfung nachweisen müssen. Der Morse-Rekord liegt bei 88 WpM.

Weiterentwicklung des Modells

Scratch bietet eine Vielzahl von Möglichkeiten, um deinen Morse-Client zu erweitern. So kannst du die gesendeten oder die empfangenen Morse-Zeichen statt über die Empfangs-LED auch akustisch anzeigen (wie bei einem echten Funkempfänger).

Wenn du den Empfangs-Thread bei einem anderen Modell (wie beispielsweise der Schranke aus [6]) ergänzt, kannst du aus der Ferne Kommandos an das Modell übertragen – z. B. Wartungseingriffe wie das Öffnen oder Schließen der Schranke oder Texte, die an der Schranke (ggf. in verschiedenen Sprachen) über Lautsprecher ausgegeben werden sollen.

Mit zwei Controllern, die du optisch oder mit Kabeln miteinander verbindest, erhältst du eine Vollduplex-Kommunikationsverbindung, bei der auf beiden Seiten gleichzeitig gesendet und empfangen werden kann.

Oder du schließt einen Taster oder den Geber aus Abb. 3 an und baust deinen Morsecode-Empfänger zu einem Morse-Trainer mit Fehlerstatistiken aus.

Referenzen

- [1] Dirk Fox: *Scratch mit fischertechnik – Update 2022*. [ft:pedia 1/2022](#), S. 86–92.
- [2] Dirk Fox: *Morsetelegraf*. [ft:pedia 1/2014](#), S. 4–10.
- [3] Dirk Fox, Thomas Püttmann: *Technikgeschichte mit fischertechnik*. dpunkt-Verlag, 2015.
- [4] Fischerwerke: *Bauanleitung Morsegerät*. Club-Modell 1975-1, 1975.
- [5] Fischerwerke: *Ein Schreibgerät für Morsezeichen*. In: *Anleitungsbuch Elektromechanik (em2)*, Tumlingen 1975, S. 29–31.
- [6] Dirk Fox: *Einführung in ftScratch (1): Die Schranke*. [ft:pedia 1/2022](#), S. 93–97.
- [7] Dirk Fox: *Einführung in ftScratch (2): Der Münzautomat*. [ft:pedia 2/2022](#), S. 119–122.
- [8] Dirk Fox: *Einführung in ftScratch (3): Der Barcodeleser*. [ft:pedia 3/2022](#), S. 85–92.

Computing

Hard- und Softwareübersicht zur Ansteuerung von fischertechnik

Axel Chobe

Eine der besonderen Merkmale von fischertechnik ist die Möglichkeit, Modelle anzusteuern – mit Mikrocontrollern und Fernsteuerungen. Der Beitrag gibt eine aktuelle Übersicht.

Seit Mitte der 1980er Jahre beschäftigt sich fischertechnik auch mit Computing. Jeweils dem Stand der Technik entsprechend entwickelte fischertechnik passende Hard- und Software zur Ansteuerung von Modellen.

Es begann mit dem Parallel-Interface und endet z. Z. mit dem TXT 4.0. Auch die Software passte sich den Entwicklungen der Betriebssysteme an: von DOS über alle möglichen Windows-Versionen bis schließlich auch für Android und Linux gab bzw. gibt es passende Entwicklungsumgebungen, von LLWIN bis ROBO Pro Coding [1].

Daneben entwickelte auch die Fan-Gemeinschaft eigene Hard- und Software. Die Motivation ist zum einen sicherlich die Verbesserung der Software, um deren Möglichkeiten zu erweitern. Zum anderen soll die Hardware preisgünstig sein und ebenfalls erweitert genutzt werden können. An dieser Stelle dafür von mir herzlichen Dank




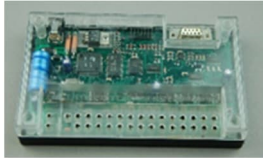



an alle Beteiligten für das solcherart erweiterte „Spielvergnügen“.









Drittens gibt es zusätzliche Hardware-Adapter für verbreitete Mikrocontroller wie z. B. den micro:bit, den Calliope oder den Arduino. Aber auch verbreitete Programmiersprachen wie z. B. Scratch wurden an fischertechnik-Schnittstellen angepasst bzw. passende Treiber entwickelt.

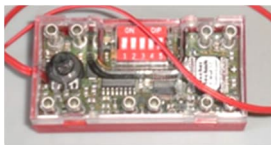



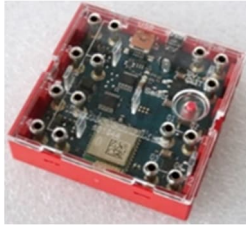
Vor diesem Hintergrund wird klar, dass das inzwischen ein komplexes Thema ist, und es schon eine Herausforderung darstellt, hier den Überblick zu behalten. Aber auch bei einem historischen Dachbodenfund von fischertechnik-Bauteilen ist es sicherlich interessant, diesen passend einzuordnen zu können.





Im Folgenden wird zuerst die Entwicklung der Mikrocontroller dargestellt; anschließend werden die historische Entwicklung und die Möglichkeiten der Software beschrieben.

Hardware zur Steuerung von fischertechnik-Modellen





Beschreibung	Artikel	Handbuch
<p>Parallel-Interfaces (30520)</p> <p>Die ersten Interfaces datieren aus den Jahren 1984/85 und waren für die damaligen Home-Computer von Atari, Commodore und Schneider vorgesehen. Mit der Verbreitung der IBM-PCs setzten sich Interfaces mit Centronics-Schnittstelle durch.</p> <p>Software: Basic, Maschinensprache</p> <p><i>Heute noch aktuell; siehe [2].</i></p>		
<p>Flip-Flop (36121)</p> <p>Einführungsjahr: 1991</p> <p>Software: keine (Steuerung über 1 DIP und 2 Potis)</p> <p>Verbindung: keine</p> <p>Eingänge: 2 universal</p> <p>Ausgänge: 2 (Motor, Lampe, Summer)</p> <p><i>Baustein ist das Herzstück des Sensoric Baukastens; kann als „komplexer Silberling“ verstanden werden.</i></p>		
<p>Intelligent Interface (30402)</p> <p>Einführungsjahr: 1997</p> <p>Software: LLWin</p> <p>Verbindung: serielle Schnittstelle (SubD9)</p> <p>Eingänge: 8 digital (TTL), 2 analoge Widerstände, Fotozelle</p> <p>Ausgänge: 4 Motoren oder 8 Lampen</p> <p><i>Einfache Abfragen können mit Python realisiert werden. Sehr selten und daher teuer.</i></p>		
<p>E-Tec-Modul (91083)</p> <p>Einführungsjahr: 2003</p> <p>Software: keine (8 Programme über DIP-Schalter)</p> <p>Verbindung: keine (läuft autark)</p> <p>Eingänge: 3 digital</p> <p>Ausgänge: 1 Motor oder 2 Lampen</p> <p><i>Neben den DIP-Schalter-Programmen können auch Grundlagen der Digitaltechnik erlernt werden [3].</i></p>		






Beschreibung	Artikel	Handbuch
<p>ROBO Interface (93293)</p> <p>Einführungsjahr: 2004 Software: ROBO Pro, Basic, Python [4] Verbindung: seriell 9 polig, USB Eingänge: 8 digital (TTL), 2 analoge Widerstände, 2 analoge Spannungen, 2 Abstandssensoren Ausgänge: 4 Motoren oder 8 Lampen Fernbedienbar: IR-Control Set (30344), Data Link (93295/112833)</p> <p><i>Erster Controller mit ROBO Pro. Selten und daher teuer.</i></p>		
<p>ROBO I/O Extension (93294)</p> <p>Einführungsjahr: 2006 Software: ROBO Pro Verbindung: USB für direkte Nutzung, 10 pol. Flachbandkabel an ROBO Interface Eingänge: 8 digital (TTL), 1 analoger Widerstand Ausgänge: 4 Motoren oder 8 Lampen</p> <p><i>Läuft selbstständig oder als Erweiterung zum ROBO Interface.</i></p>		
<p>ROBO TX Controller (500995) [5]</p> <p>Einführungsjahr: 2009 Software: ROBO Pro, C Verbindung: USB, Bluetooth, I²C (5V) Eingänge: 8 universal, 4 schnelle Zähleringänge Ausgänge: 4 Motoren oder 8 Lampen Fernbedienbar: mit App RoboTXdroid (via Bluetooth) Display: 128×64 Pixel monochrom</p> <p><i>Weitere Infos zu I²C unter [6, 7].</i></p>		
<p>ROBO LT Controller (139308/147225)</p> <p>Einführungsjahr: 2010 Software: ROBO Pro Light Verbindung: USB Eingänge: 3 universal Ausgänge: 2 Motorausgänge</p> <p><i>Der 147225 kann auch über USB mit Strom versorgt werden.</i></p>		

Beschreibung	Artikel	Handbuch
<p>Electronics-Modul (152063)</p> <p>Einführungsjahr: 2014 Software: keine, 8 Programme über DIP-Schalter Verbindung: keine, läuft autark Eingänge: 3 digital Ausgänge: 2 Motoren oder 4 Lampen</p> <p><i>Zweite Variante des E-Moduls, mit Potentiometer und höheren Eingangsfrequenzen.</i></p>		
<p>TXT Controller (153513)</p> <p>Einführungsjahr: 2014 Software: ROBO Pro, C, Python, Scratch Verbindung: USB, WLAN, Bluetooth, I²C (3,3 V) Eingänge: 8 universal, 4 schnelle Zählengänge, USB-Kamera inkl. Mikrophon Ausgänge: 4 Motoren oder 8 Lampen, Lautsprecher Fernbedienbar: Plus Control-Set (500881/136581), App TXTCamControl (iOS), TXTCamdroid (Android) Display: 6,1 cm, 320×240 Pixel, farbig, touch</p> <p><i>Auslaufmodell; z.T. günstig zu bekommen.</i></p>		
<p>Robotics MiniBots-Modul (156499)</p> <p>Einführungsjahr: 2016 Software: keine, 8 Programme über DIP-Schalter Verbindung: keine, läuft autark Eingänge: 3 digital (Schalter, Spurensucher) Ausgänge: 2 Motoren</p> <p><i>Für Kinder ohne Programmierkenntnisse.</i></p>		
<p>ROBOTICS BT Smart Beginner (161944)</p> <p>Einführungsjahr: 2017 Software: ROBO Pro Light, Scratch Verbindung: USB, Bluetooth Eingänge: 4 universal Ausgänge: 2 Motoren oder 4 Lampen Fernbedienbar: ROBOPro Smart für Android und iOS</p> <p><i>Sehr gut für Einsteiger geeignet; erstmals mit Bluetooth.</i></p>		







Beschreibung	Artikel	Handbuch
<p>E-Tronics-Modul (182323)</p> <p>Einführungsjahr: 2021 Software: keine, 8 Programme über DIP-Schalter Verbindung: keine, läuft autark Eingänge: 3 digital Ausgänge: 2 Motor oder 4 Lampen</p> <p><i>Dritte Variante des E-Moduls.</i></p>		
<p>TXT 4.0 Controller (560166) [8]</p> <p>Einführungsjahr: 2022 Software: ROBO Pro Coding, C, Python Verbindung: USB, WLAN, Bluetooth, I²C (5 V) Eingänge: 8 universal, 4 schnelle Zählergänge Ausgänge: 4 Motoren oder 8 Lampen, Lautsprecher, 3 Servoausgänge Fernbedienbar: ja Display: 320×240 Pixel, kapazitives Touch-Display</p> <p><i>Komplettkasten leider teuer, sonst top.</i></p>		



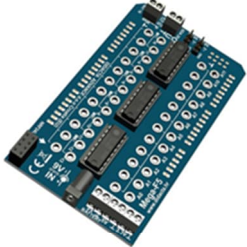

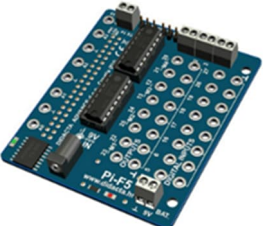
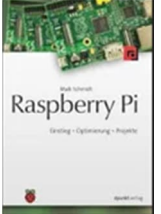
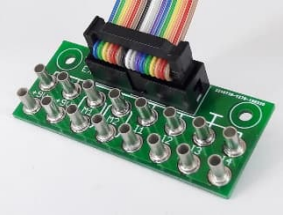
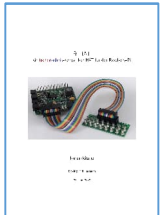
Fernsteuerungen

Beschreibung	Artikel	Handbuch
<p>Funkfernsteuerung (30270), Fernlenkset (30375)</p> <p>Einführungsjahr: 1983 Zweiter Fernsteuerungsempfänger möglich. Motorblock mit 3-Gang-Schaltung inkl. Differenzial und Gelenkwelle sowie einem Servomotor.</p> <p><i>Sehr selten und darum teuer.</i></p>		
<p>IR Control Set (30344)</p> <p>Einführungsjahr: 1998 Empfänger: kann auch direkt vom ROBO Interface gesteuert werden; es kann ein zweiter Empfänger angesprochen werden 3 Motorausgänge Sender: IR, Raupenfunktion möglich</p> <p><i>Sehr selten und darum teuer.</i></p>		

Beschreibung	Artikel	Handbuch
<p>ROBO RF Data Link (93295/112833)</p> <p>Einführungsjahr: 2004 Empfänger: muss in ROBO Interface verbaut werden. Sender: mit USB-Typ-B an PC</p> <p><i>Nicht mehr zu erhalten.</i></p>		
<p>Control Set (500881/136581)</p> <p>Einführungsjahr: 2008 Empfänger: kann auch vom TXT gesteuert werden. Es kann ein zweiter Empfänger angesprochen werden. Ausgänge: 3 Motorausgänge, 1 Servoausgang Sender: IR, 2 Frequenzen, Raupenfunktion möglich</p> <p><i>Ideale Ergänzung zum TXT, einfache Programmierung.</i></p>		
<p>Bluetooth Control Set (540585)</p> <p>Einführungsjahr: 2017 Ausgänge: 3 Motorausgänge, 1 Servoausgang Sender: Raupenfunktion und Tempomat-Funktion möglich; auch über Bluetooth Control App für Android und iOS mit Smartphone oder Tablet</p> <p><i>Fernbedienung über Bluetooth.</i></p>		
<p>Plus Control Set – Bluetooth-Steuerung</p> <p>Einführungsjahr: 2022 Ausgänge: 3 Motorausgänge, 1 Servoausgang Sender: Bluetooth Control App für Android und iOS mit Smartphone oder Tablet</p> <p><i>Preiswerte Möglichkeit einer Bluetooth-Steuerung, da kein Sender benötigt wird.</i></p>		

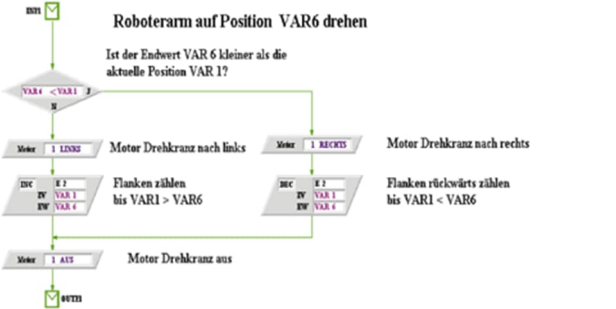
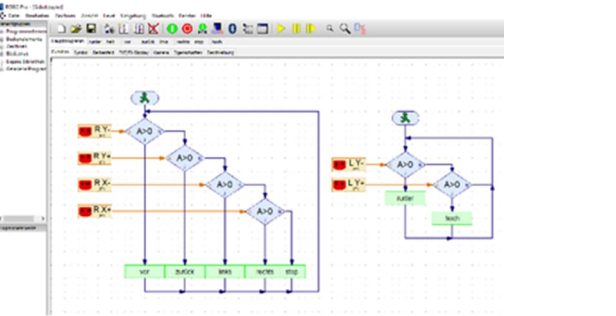
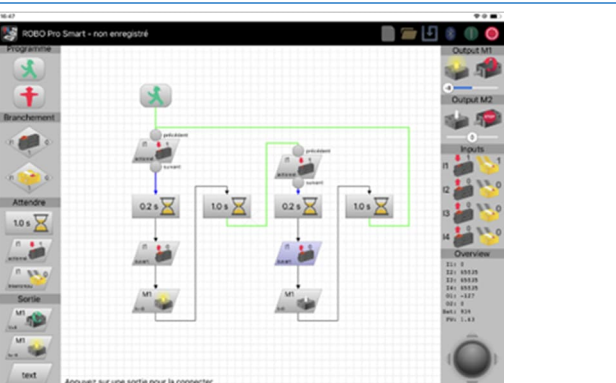
Controller und Adapter, die nicht von fischertechnik stammen

Beschreibung	Artikel	Handbuch
<p>ftDuino [9]</p> <p>Einführungsjahr: 2017 Software: Arduino IDE, Scratch, Blockly, ROBO Pro (online) [10] Verbindung: Mini-USB, I²C (5 V) Eingänge: 8 universell, 4 schnelle Zählergänge Ausgänge: 4 Motoren oder 8 Lampen Fernbedienbar: ja Display: teilweise mit OLED 128×32 Pixel</p> <p><i>Preisgünstiger als das Original; hoher Lerneffekt, Zugang zur „Arduino-Welt“.</i></p>		
<p>Calliope mini 2.0</p> <p>Einführungsjahr: 2017 Software: MakeCode, OPEN Roberta Lab, Scratch, MInd+ Verbindung: USB Eingänge: 4 analoge Ausgänge: 8-11 Ein-/Ausgangsanschlüsse Display: 5×5 Pixel</p> <p><i>Einsatz hauptsächlich in Schulen, da preisgünstig. fischertechnik-Interface verfügbar [11].</i></p>		
<p>micro:bit (IO F5 Adapter) [12, 13]</p> <p>Einführungsjahr: 2018 Software: MakeCode, Python, Scratch Verbindung: USB Eingänge: 6 Ausgänge: 4 Motoren oder 8 Lampen Fernbedienbar: ja (Bluetooth oder weiteren micro:bits) Display: 5×5 Pixel</p> <p><i>Einfache Programmierung mit MakeCode.</i></p>		

Beschreibung	Artikel	Handbuch
<p>micro:bit (BBCZ INTERFACE) [14]</p> <p>Einführungsjahr: 2018 Software: MakeCode, Python, Scratch Verbindung: USB Eingänge: 4 Ausgänge: 2 Motoren oder 4 Lampen Fernbedienbar: ja (Bluetooth oder weiteren micro:bits) Display: 5×5 Pixel</p> <p><i>Einfache Programmierung mit MakeCode.</i></p>		
<p>Arduino UNO/MEGA (fischertechnik UNO/Mega-F5-Shield)</p> <p>Einführungsjahr: 2018 Software: frei wählbar (C, Scratch, ...) Verbindung: Aufsteckplatine auf Arduino Uno bzw. Arduino Mega Eingänge: 12 digital/analog Ausgänge: 12 Display: keins</p> <p><i>Gute Anleitung im Buch rechts [15].</i></p>		
<p>Raspberry Pi (fischertechnik Raspberry PI-F5)</p> <p>Einführungsjahr: 2018 Software: frei wählbar (C, Scratch, ...) Verbindung: Aufsteckplatine auf Raspberry Pi</p> <p>Eingänge: 8 digitale, 4 analoge, I²C Ausgänge: 4 Motoren od. 8 Lampen Display: keins</p> <p><i>Schlägt eine Brücke zwischen Raspberry Pi und fischertechnik.</i></p>		
<p>Raspberry Pi (TX-Pi-HAT)</p> <p>Einführungsjahr: 2019 Software: frei wählbar (C, Scratch, ...) Verbindung: Aufsteckplatine auf Raspberry Pi</p> <p>Eingänge: 4 digitale, I²C Ausgänge: 2 Motoren Display: optional</p>		

Beschreibung	Artikel	Handbuch
<p>ftSwarm [16, 17]</p> <p>Einführungsjahr: 2021 Software: C (Python in Vorbereitung) Verbindung: Micro-USB, WLAN, I²C (3.3V) Eingänge: 4 universal, Gyro (optional) Ausgänge: 2 Motoren, 1 Servo, bis zu 18 RGB-LEDs Fernbedienbar: ja (WLAN oder weitere ftSwarm/ftSwarmControl) Display: keins</p> <p><i>Klein, preisgünstig, mehrere Controller werden über WLAN vernetzt.</i></p>		
<p>ftSwarmControl [16, 17]</p> <p>Einführungsjahr: 2021 Software: C (Python in Vorbereitung) Verbindung: Micro-USB, WLAN, I²C (3,3/5V) Eingänge: 4 digital, 8 Taster, 2 Joysticks, Gyro (optional) Ausgänge: 2 Motoren Fernbedienbar: ja (WLAN oder weitere ftSwarm/ftSwarmControl) Display: OLED 128x64 Pixel</p>		

Programmiermöglichkeiten der Controller

Beschreibung	Screenshot
<p>LLWIN (Lucky Logic for Windows)</p> <p>Das Programm LLWin steuert die fischertechnik-Modelle über das am PC angeschlossene Interface. Mit der Version LLWin 2.1 kann man dabei sowohl das Intelligent Interface (30402) als auch das frühere Universal-Interface (30520) betreiben.</p> <p><i>Heute kaum noch in Gebrauch.</i></p>	
<p>ROBO Pro (für alle fischertechnik-Controller inklusive ftDuino [10])</p> <p>In ROBO Pro werden Programme wie ein grafischer Programmablaufplan aus Kommandos und Verzweigungen zusammengesetzt. Das fertige Programm kann entweder auf dem PC gestartet werden und steuert dann via USB-Schnittstelle den Controller (Online-Mode), oder aber mit einem Klick übersetzt und auf den TX- bzw. TXT-Controller übertragen werden.</p> <p><i>Meistgenutztes Programm, großer Leistungsumfang. Befehlsübersicht unter [18].</i></p>	
<p>ROBO Pro Light (für ROBO LT und BT Smart)</p> <p>Abgespeckte Version von ROBO Pro. Für den BT Smart gibt es ROBO Pro Smart für Android und iOS.</p> <p><i>Geeignet für Einsteiger, geringer Leistungsumfang.</i></p>	

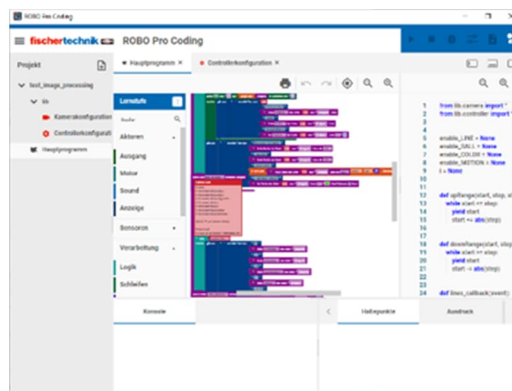
Beschreibung

ROBO Pro Coding (für TXT 4.0)

Die Software ROBO Pro Coding bietet in ihrer Umgebung neben der Möglichkeit zur grafischen Programmierung eine textbasierte mit Python und läuft unter Windows, Linux oder MacOS sowie auf mobilen Endgeräten (Android oder iOS). Selbst erstellte Programme können lokal auf dem Gerät und online in der Cloud gespeichert werden. Über den Interface-Test können Aktoren und Sensoren schnell getestet werden.

Leider zurzeit noch nicht ausführlich dokumentiert [19].

Screenshot

**ftScratch 3 / Scratch 3 (für TXT, BT Smart / ftDuino, Arduino, Calliope, micro:bit)**

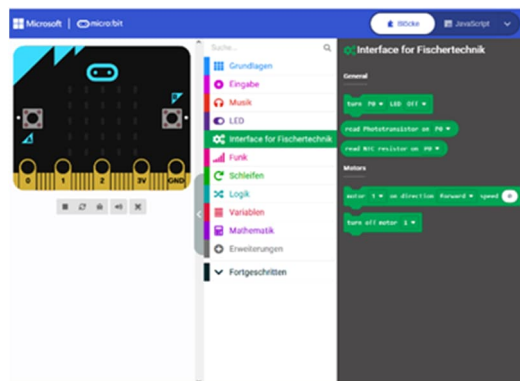
Scratch ist eine grafische Programmiersprache. Statt schriftlicher Anweisungen werden Befehlsblöcke in der richtigen Reihenfolge angeordnet. Syntaxfehler werden durch die Gestaltung der Blöcke verhindert.


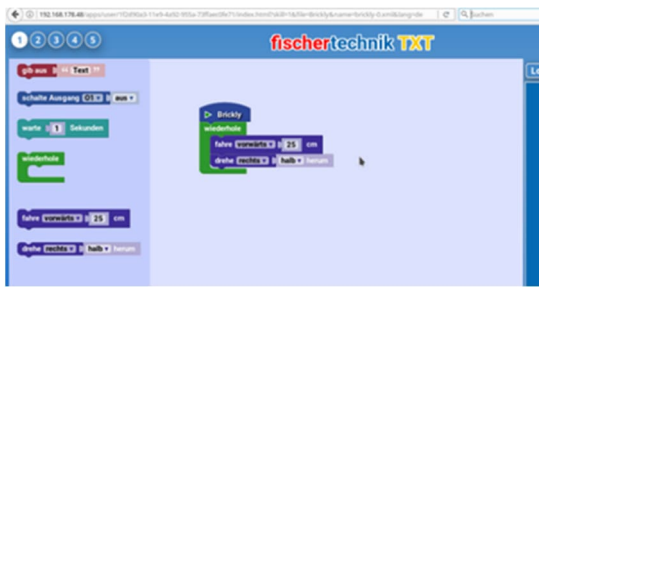
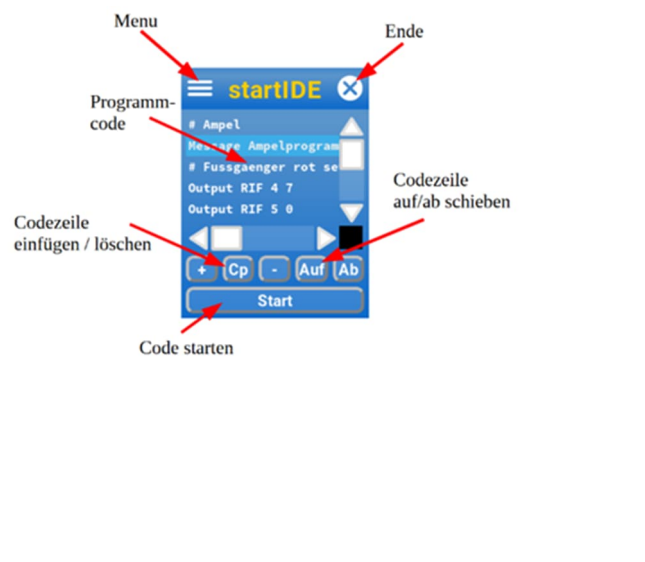
Hervorragende Programmiersprache für Einsteiger [20]. Ausführliche Dokumentation zu Scratch unter [21].

**MakeCode (für micro:bit)**

Die Programmierung erfolgt entweder in einer Blocksprache oder durch Umschalten in der oberen Zeile in JavaScript. Alternativ kann der micro:bit auch über den Python Editor Mu programmiert werden [22].

Webbasierte Oberfläche. Beschreibung unter [12].



Beschreibung	Screenshot
<p>OPEN Roberta Lab (für Calliope Mini)</p> <p>Open Roberta ist ein vom Fraunhofer Institut entwickelter Editor, mit dem sich verschiedene Mikrocontroller und Roboter in der grafischen Programmiersprache NEPO® programmieren lassen [23].</p> <p><i>Webbasierte Programmieroberfläche; basiert auf Blockly.</i></p>	
<p>Brickly [24] (für TXT; erfordert die Community Firmware [25])</p> <p>Mit Brickly kann man im Browser, z. B. auf einem Tablet oder einem PC grafisch Programme erstellen, die auf dem TXT laufen. Der TXT dient dabei als Webserver, auf den man über WLAN zugreifen kann. Die Programme können die Eingänge des TXT auslesen und seine Ausgänge ansteuern.</p> <p><i>Basiert auf Blockly; Vorläufer von ROBO Pro Coding.</i></p>	
<p>startIDE [26] (für TXT; erfordert die Community Firmware [25])</p> <p>startIDE ist eine Programmier-App für die Community Firmware des TXT Controllers (auch für den TX-Pi), mit der sich eine Vielzahl einfacher Steuerungen programmieren lässt.</p> <p><i>Auf dem TXT-Display etwas fummelig. Es besteht auch die Möglichkeit, das Programm in einem Texteditor zu schreiben. Übersicht der Befehle unter [27].</i></p>	

Beschreibung

ftrobopy [28] (für TXT, TXT 4.0)

Ansteuerung des TXT und des TXT 4.0 Controllers in Python. Es wird die Bibliothek „ftrobopy“ benötigt.

Kurzdokumentation unter [29].

Screenshot

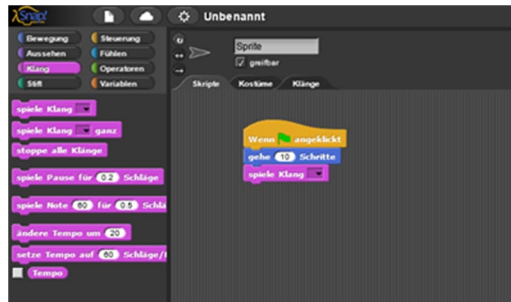
```

test100.py - C:/Users/Axel/AppData/Local/Programs/Python/Python39/test100.py (3.9.6)
File Edit Format Run Options Window Help
import ftrobopy # Bibliothek laden
import time # Bibliothek laden (für Pausefunktion)
txt = ftrobopy.ftrobopy('192.168.178.54') # Verbindung zum TXT
Lampe = txt.output(7) # Erzeugung eines allgemeinen Output-Objekt, Ausgang 8
Lampe.setLevel(512) # Methode des Objektes (Helligkeit zw. 0 und 512)
time.sleep(2) # 2 sek. Pause , Modul time erforderlich
Lampe.setLevel(0) # Lampe aus
  
```

FT-Robo-Snap (für TXT)

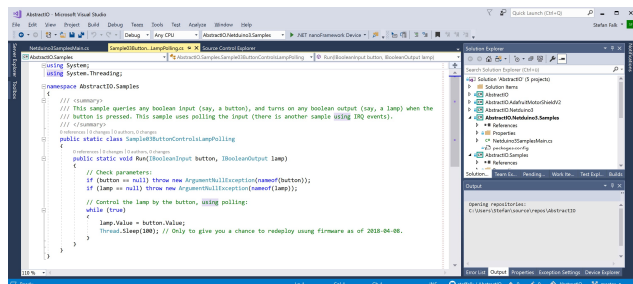
Ein Webinterface und eine IDE für den fischertechnik ROBOTICS TXT Controller, basierend auf Snap! [30]

Snap!, bis zu Version 3.1.1 unter dem Namen BYOB bekannt, ist eine auf Scratch aufbauende und von Scheme und damit auch Lisp inspirierte bildungsorientierte visuelle Programmiersprache.

**nanoFramework [31] (für NetDuino, ESP32, ...)**

Eine .net-Runtime nebst Framework für 32-Bit-Microcontroller wie NetDuino, ESP32 u.v.a. Man programmiert und debugged in C# unter Microsoft Visual Studio (nur Windows) oder Visual Studio Code.

Richtet sich an professionelle Produktentwicklung, ist aber auch für fischertechnik nutzbar. Ggf. wird zusätzlich zum CPU-Board Motortreiber-Hardware benötigt.



Referenzen

- [1] Till Harbaum: *Von Lucky Logic zu RoboPro Coding*. [ft:pedia 1/2021](#), S. 103–109.
- [2] Jens Lemkamp: *Parallel-Interface durch Arduino gesteuert (1)*. [ft:pedia 1/2014](#), S. 24–30.
- [3] Axel Chobe: *E-Tec Analog/Digital*. Übersicht über die DIP-Einstellungen des E-Tec-Moduls, auf [chobe.info](#).
- [4] Erik Andresen: *The fischertechnik Interface for the Rest of us*. [ft:pedia 2/2012](#), S. 32–38.
- [5] Stefan Brunner: *Die Ein- und Ausgänge des TX Controllers*. [ft:pedia 4/2012](#), S. 24–31.
- [6] Axel Chobe: *I²C*. Überblick über I²C mit fischertechnik, auf [chobe.info](#).
- [7] Dirk Fox: *I²C mit TX und Robo Pro – Teil 1: Grundlagen*. [ft:pedia 3/2012](#), S. 32–37.
- [8] Till Harbaum: *TXT 4.0 Controller – ein Blick unter die Haube*. [ft:pedia 2/2021](#), S. 112–116.
- [9] Till Harbaum: *ftDuino – Open-Source trifft Konstruktions-Baukasten*. [ft:pedia 1/2018](#), S. 85–91.
- [10] Dirk Fox: *TX-Simulator für den ftDuino*. [ft:pedia 3/2022](#), S. 110–120.
- [11] André Marcel: *Calliope Mini fischertechnik-Interface*. [ft:pedia 2/2020](#), S. 86–91.
- [12] Axel Chobe: *micro.bit*. Auf [chobe.info](#).
- [13] Axel Chobe: *Micro:Bit*. [ft:pedia 4/2019](#), S. 47–51.
- [14] Axel Chobe: *Micro:Bit (Teil 2)*. [ft:pedia 3/2020](#), S. 90–92.
- [15] Dirk Fox, Thomas Püttmann: *fischertechnik-Roboter mit Arduino*. dpunkt.verlag, 2020.
- [16] Christian Bergschneider, Stefan Fuss: *ftSwarm – Das Prinzip Heinzelmännchen*. [ft:pedia 2/2021](#), S. 107–111.
- [17] Stefan Fuss, Christian Bergschneider: *ftSwarm (2) – Codierst Du noch oder schwärmst Du schon?* [ft:pedia 2/2022](#), S. 134–137.
- [18] Axel Chobe: *Übersicht der RoboPro Befehle für Robo Interface, TX, TXT*. Auf [chobe.info](#).
- [19] Till Harbaum: *ROBO Pro Coding – ein Blick hinter die Kulissen*. [ft:pedia 2/2021](#), S. 117–122.
- [20] Dirk Fox: *Scratch mit fischertechnik – Update 2022*. [ft:pedia 1/2022](#), S. 86–92.
- [21] Axel Chobe: *Scratch – Einführung in die Programmiersprache*. Auf [chobe.info](#).
- [22] microbit.org: [MakeCode](#).
- [23] Open Roberta Lab: [NEPO](#).
- [24] Till Harbaum: *Brickly auf dem TXT: Grafische Programmierung à la Google-Blockly*. [ft:pedia 1/2017](#), S. 92–98.
- [25] Till Harbaum: *Auf zu neuen Ufern: Die Geschichte der „Community-Firmware“ für den TXT*. [ft:pedia 4/2016](#), S. 59–67.
- [26] Peter Habermehl: *startIDE für die Community Firmware – Programmieren direkt auf dem TXT oder TX-Pi*. [ft:pedia 1/2018](#), S. 102–107.
- [27] Beschreibung der fischertechnik TXT Community Firmware auf [chobe.info](#).

- [28] Torsten Stuehn: *Programmierung des TXT mit Python*. [ft:pedia 2/2017](#), S. 58–62.
- [29] Axel Chobe: Kurzdokumentation zu ftrobopy auf [chobe.info](#).
- [30] UC Berkeley: *Snap!*
- [31] .NET nanoFramework: <https://nanoframework.net/>