

Editorial

Sprachen lernen

Programmieren gilt bis heute meist als eine Fertigkeit, deren Erlernen vor allem mathematisch oder technisch talentierten Menschen besonders leicht fällt. Das mag eine der Ursachen dafür sein, dass Mädchen und Frauen um Programmieren meist einen größeren Bogen machen, da sie sich selbst seltener als mathematisch oder technisch begabt einschätzen.

Zwar dreht sich ein Teil der Tätigkeit eines Programmierers tatsächlich um mathematische Logik: Die Nutzung von Bedingungen und Schleifen setzt eine vorab erfolgte gedankliche Beschreibung des Lösungswegs in abstrakter Logik voraus. Aber das ist sicherlich nicht der Teil, vor dem Menschen zurückschrecken – denn die mathematische Komplexität dahinter ist überschaubar. Was abschreckt, ist die Hürde des Erlernens des Programmierens selbst.

Dabei ist jede Programmiersprache in erster Linie einfach nur eine „Sprache“ – und eine extrem simple obendrein. Ihre Syntax ist stark formalisiert, daher gibt es praktisch keine Ausnahmen; das Vokabular besteht nahezu ausschließlich aus einer sehr endlichen Anzahl von Kommandos. Konjugation, Deklination und Konjunktive sucht man vergeblich – alles wird in direkter Rede und ausschließlich mit Imperativen formuliert.

Daher liegt es eigentlich nahe, eine Programmiersprache als „Fremdsprache“ für die Kommunikation mit einem Computer zu begreifen – und auch so zu unterrichten.

Dirk Fox, Stefan Falk

Dass es beim Erlernen einer Programmiersprache tatsächlich viel mehr auf das Sprachtalent als auf die mathematische oder technische Begabung ankommt, hat vor drei Jahren die [Lernforscherin Chantel Prat](#) vom Institut für Learning and Brain Science der Universität Washington [nachgewiesen](#).

Sie konnte in Experimenten mit Vergleichsgruppen zeigen, dass Menschen mit sprachlicher Begabung beispielsweise die (dem Englischen besonders ähnliche) Programmiersprache Python tendenziell schneller erlernen als Menschen mit geringerer oder ohne sprachliche Begabung – mit keinem anderen Talent konnten die Unterschiede in der Lerngeschwindigkeit erklärt werden (siehe auch ihr [Youtube-Video](#)).

Das bedeutet natürlich nicht, dass für die Entwicklung guter Programme und Algorithmen mathematische und technische Begabungen irrelevant sind – wohl aber, dass Menschen mit sprachlicher Begabung das Programmierhandwerk im Vergleich mit anderen Menschen eher besonders leicht und schnell erlernen.

Eine gute Nachricht für alle, die sich bisher als „informationstechnisch unbegabt“ verstehen. Und eine Einsicht, die zu einem Überdenken unserer Schul-Curricula führen sollte: Coden gehört nicht in die Nerd-Ecke, sondern sollte Teil des Fremdsprachenunterrichts sein.

Beste Grüße,
Euer ft:pedia-Team

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

Inhalt

Sprachen lernen.....	2
Großprojekt Seilbahn (Teil 7): Zusammenspiel	4
fischertechnik-Figuren im Maßstab 1:10	12
POD-Antrieb für ein fischertechnik-Schiff.....	22
Ein Bild von einem Modell (Teil 3): Betonmischer 1967.....	27
Die erste Lokomotive	32
Die Odyssee mit den Steckern.....	35
Rutschen und Rappeln.....	37
Rotationsencoder (Rotary Encoder) KY-040	38
Quadratur-Encoder für fischertechnik-Encoder-Motoren	41
Fahrtregler (2): Vom Widerstandsdraht bis zur PWM	47
Digitale Uhr mit Echtzeitgenerator.....	53
Silberlinge: Original oder Nachbau (Teil 11).....	57
Sensoren	65
V. I. P. – Ein I ² C-nach-Computing-Interface-Umsetzer (Teil 5)	70
Die Beispielprogramme von ROBO Pro Coding für den TXT 4.0 im Überblick.....	77
Laserprojektion (2): Lissajous-Figuren	89
Balancier-Plattform	95

Termine

Was?	Wann?	Wo?
Clubdag Schoonhoven	28.10.2023	Schoonhoven
Südconvention 2023	21.- 22.10.2023	Fördertechnik- Museum Sinsheim

Impressum

<http://www.ftpedia.de>

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

Autoren: Florian Bauer, Gerhard Birkenstock, Axel Chobe,
Arnoud van Delden, Stefan Falk, Dirk Fox, Hans-Christian
Funke, Peter Krijnen, Rüdiger Riedel, Tilo Rust, René Trapp.

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

Großprojekt Seilbahn (Teil 7): Zusammenspiel

Tilo Rust

Diese Serie begleitet das Großprojekt „Kuppelbare Einseilumlaufbahn / Doppelmayr (10-MGD)“ im Fördertechnik-Museum Sinsheim von Anfang bis zur Fertigstellung und Ausstellung auf der BUGA 2023 in Mannheim. Diesmal beleuchten wir einige Aspekte, bei denen viele Teile und Menschen miteinander gut zusammenspielen mussten.

Fast alle wichtigen Komponenten des Modells sind nun fertiggestellt und funktionieren für sich gesehen ausreichend gut. Die Reifenförderer beider Stationen laufen sauber und können ein Fahrzeug durch den Umlauf fördern. Prinzipiell funktionieren

die Kuppelstrecke und die Türöffnung, auch wenn bei letzterer noch der Mechanismus an der Kabine optimiert werden muss. Prinzipiell funktioniert auch der Hauptantrieb, wenn er Strom bekommt, und kann die Seilscheibe und das Seil bewegen. Auch



Abb. 1: Alle Einzelkomponenten sind an der Anlage angebracht und funktionieren für sich selbst genommen zumindest in den Tests. Doch nun geht es an die Feinarbeit der Abstimmung aller Komponenten aufeinander, denn erst wenn diese perfekt zusammenspielen, kann das Modell zeigen, was es draufhat.

die Hauptsteereinheit funktioniert im Labor einwandfrei und kann die Reifenförderer mit der Seilgeschwindigkeit synchronisieren (siehe Video). Das eingehängte Mittelseil trägt das Datenkabel, welches die zwei Stationen verbindet und wurde mit Kollisionsleuchten (roten Blink-LED) versehen. Ebenso wurde der Steuerstand mit Datenanbindung und Strom versorgt und das Steuerpult von Doppelmayr an unser System angepasst. Auch die Swarms haben bereits einige Grundfunktionen gezeigt und die grafische Oberfläche (Connect) zeigt, was sie soll. Die Klemmen der Gondeln halten zuverlässig, und nachdem die erste Rollenbatterie gefertigt ist und die Stützen neu konstruiert wurden, waren die ersten Tests vielversprechend. Die Grundgestelle sind stabil und lassen sich mit Staplern bewegen, die Verschaltungen sind zumindest teilweise vorhanden und die Montage klappt wie vorgesehen sehr schnell und sauber. Selbst die Dekoration, wie die Verkleidung des Hauptantriebs und einige optische Gimmicks sind sauber integriert.

Doch wie all diese Komponenten zusammenspielen und ob das Gesamtsystem funktioniert – das ist die große Frage.

Abb. 2 zeigt die Verkleidung des Hauptantriebs, stellvertretend für einen Doppelmayr Direct Drive. Während die Pneumatikschläuche die Kabel sein sollen (hier fehlt noch ein „Schaltschrank“), gibt es auch ein Sicherheitsgeländer, denn der Antrieb ist wie im Original um $3,5^\circ$ gekippt. Darunter befindet sich ein 24-V-Industriemotor der Firma Ott und zwei große Hauptkugellager der Firma Rogalla. Die Hauptwelle von 40 mm Durchmesser und die Edelstahlbleche mit gelaserten Ausschnitten sind ebenfalls von Nebensponsoren

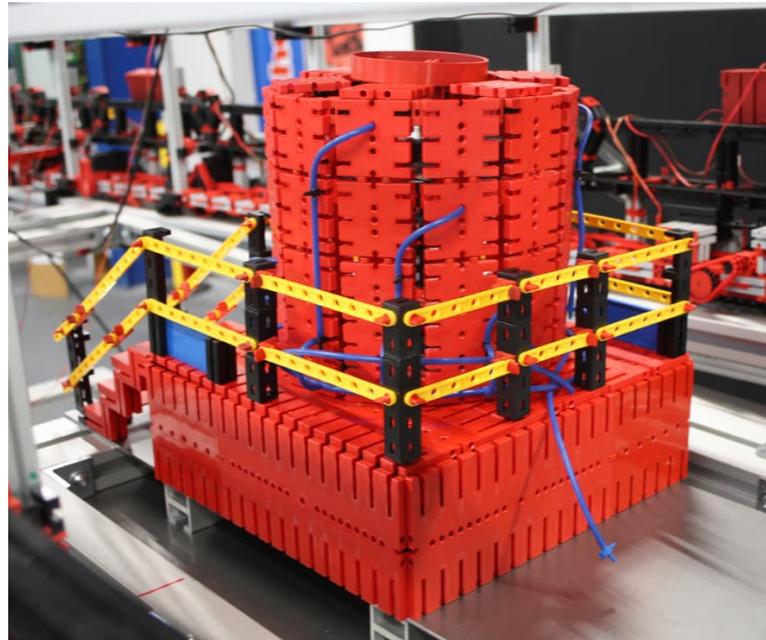


Abb. 2: Die Verkleidung des Hauptantriebs stellt einen Doppelmayr Direct Drive dar

gefertigt worden. Vielen Dank für das gute Zusammenspiel bei der Konstruktion – wir berichten mit Video noch ausführlich darüber.

Stützpunkt

Kurz vor der Abreise auf die BUGA23 stellten wir in einem Test fest, dass die Konstruktion der Stützen nicht den tatsächlichen Anforderungen entspricht und diese sich übermäßig verbiegen. Auch die Rollenbatterien selbst waren nicht richtig verschränkt und verzogen sich bereits nach ca. 60% der Nennbelastung so stark, dass das Seil entgleiste (siehe Video „Bautage“ [2]). Im Eilverfahren wurden daher eine komplette Neukonstruktion erdacht und die Prototypen gebaut. Der Zeitdruck, zur Eröffnung der BUGA23 wenigstens etwas zu zeigen, war enorm.

Dank des perfekten Zusammenspiels zwischen dem Techniker des Fördertechnik-Museums, einem Werkstudenten und unserem Team hatten wir binnen 24 Stunden alles neu geplant und die Teile



Abb. 3: Gesamtansicht

bestellt bzw. sogar schon aus der Werkstatt besorgt. Der Techniker machte Überstunden und schraubte mit dem Studenten die Stützen zusammen, betonierte das Fundament und fand Lösungen zur Höhenverstellbarkeit. Im Zusammenspiel mit dem fischertechnik-Designer und den fischertechnik-Werken wurden neue Alu-Profile bestellt und auf den Weg gebracht.

Ob dies aber funktioniert, das konnten wir frühestens nach dem Zusammenbau von über 4000 Teilen während der Ausstellung sehen – die sind bis heute allerdings noch nicht ganz fertiggestellt.

Abb. 3 zeigt links im Bild die Werkstattpalette mit Kaffeemaschine und Sponsortassen, natürlich auch mit sortierten Bausteinen, Werkzeug, Licht und Stromversorgung, Bauplänen und allem, was wir sonst brauchen. In der Bildmitte die neue Stütze 1 mit der neuen Rollenbatterie. Dahinter die Antriebsstation „Spinelli“ mit ihrer neuen Verkleidung des Untergestells. Rechts im Hintergrund der Steuerstand.

Transport

Einer der vorläufigen Höhepunkte unseres Giga-Projektes war der Transport des kompletten Modells auf die BUGA23. Prinzipiell haben wir dafür die Untergestelle aus Industrie-Aluminium gebaut, so dass die Anlage stabil in wenigen Modulen transportiert werden kann. Stützen und Steuerstand sowie Werkstatt basieren auf Europaletten. Die Idee war, für alle Module eine Verschalung aus Holz herzustellen, die später auch als Stellwand in der Ausstellung verwendet werden kann. Hiervon sind wir (leider) abgerückt, da die BUGA23 diese nicht aufstellen wollte und somit ein Rücktransport nötig gewesen wäre. Gleichzeitig war geplant, die Anlage auf einem geschlossenen LKW zu transportieren, was eine Verschalung nicht unbedingt erforderlich macht. Zudem sparte uns das im Vorfeld viel Arbeitszeit und schonte das knapp werdende Budget.

Doch es kam alles anders. Der Logistikunternehmer konnte den LKW nicht bereit-

stellen und wir mussten die Module privat mit Anhängern nach Mannheim bringen. Dann schlug das Wetter zu und wir erwischten die einzige Sturmnacht in diesem Frühjahr. Und da wir keine Verschalung gebaut hatten, war unsere Anlage ungeschützt.

Nach einer abenteuerlichen Fahrt über die Autobahn und einigem Bruch (siehe Video auf unserem YouTube-Kanal) erhöhten viele Leute unseren Hilferuf und eilten zusammen. In einer Blitzaktion wurde die zweite Verschalung gebaut (die letztlich doch nicht benötigt wurde), wurden Transporter organisiert und privat mit Hänger unter der Woche für viele Pendelfahrten bereitgestellt. Ein improvisiertes Zusammenspiel dutzender Hände. Dann, nach einigen Tagen Reparatur durch ein Team, das Hand in Hand arbeitete, war es so weit: Wir waren auf der BUGA23.

Am ersten Eröffnungswochenende leisteten wir 107 Stunden Standdienst und führten in 40 Führungen über 450 Besucher in die Details der Seilbahn ein.

Abb. 4 zeigt die Antriebsstation „Spinelli“. Um den Unterbau der Station sind mit Hilfe von starken Magneten Dibond-Platten mit der Skyline von Mannheim, Seilbahn und

den Sponsorenlogos angebracht. Sie können leicht abgenommen werden, denn darunter verbergen sich Technik und Lagerraum. Später soll das Motiv die Verwendung von Seilbahnen im urbanen Umfeld promoten.

In Abb. 5 seht ihr die Umlenkstation „Luisenpark“. Sie zeigt sich auf dem 2 m hohen Unterbau in Hochglanz. Er ist so gebaut, dass man an zwei Stellen unter das Modell treten und durch eine Plexiglas-scheibe (zur Sicherung gegen Eingriff) aus der Subjektiven die Technik sehen kann. Nach der BUGA23 wird dieser Bereich noch vergrößert, aktuell ist aber der zweite Teil durch die Original-Klemme belegt und daher nicht betretbar. Die Sponsorentafeln stammen von der BUGA23, die vor dem Modell hängende Fahne ist ärgerlich – aber das ist ein anderes Thema.

Schöner präsentiert

Die darauffolgenden Tage wurden genutzt, um die Anlage noch schöner zu gestalten. Die Fahrzeuge wurden dekoriert, LEDs angebracht, diverse Details an Deko gebaut und viele Schilder mit technischen Details aufgestellt.



Abb. 4: Die Antriebsstation „Spinelli“ mit vorgelagerter Stütze 1



Abb. 5: Umlenkstation „Luisenpark“

In einer guten Zusammenarbeit mit unseren drei Hauptsponsoren gestalteten die Grafikabteilungen eine stilisierte Skyline für die Umhausung des Unterbaus, die dann auf Dibond gedruckt und montiert wurde. Es ist wirklich bemerkenswert, dass die Grafikerrinnen von Doppelmayr und fischertechnik sogar in der Freizeit Daten austauschten, um den Termin zu halten. Vielen Dank für das perfekte Zusammenspiel.

Schöner präsentieren

Ein weiteres Zusammenspiel fand zwischen unseren Mitgliedern, freiwilligen Helfern und dem RaumZeitLabor (RZL) statt. Nach einem Besuch der Ausstellungseröffnung (s. u.) von fischertechnik steuerte Marc Schrag zusätzliches Budget bei, mit dem wir einheitliche Poloshirts besorgen konnten. Im RZL halfen an mehreren Abenden die Experten beim hochwertigen Folieren der Polos mit unserem Logo. Ab sofort

können wir also schöner präsentieren – genau passend zum Termin der Gondelgala (s. u.).

Groß und klein

Dann ein Anruf von Doppelmayr: Wir erhalten eine originale (große) Totpunkt-klemme, wie sie in der BUGA23 Seilbahn verwendet wird. Auch hier ein perfektes Zusammenspiel zwischen der Doppelmayr-Werkstatt, die daraus ein Ausstellungsstück fertigte, dem Logistikunternehmer und der Stadtgärtnerei, die uns beim Abladen und Transport in die Ausstellungshalle halfen, sowie einem jungen Mitglied, das in den Schulferien früh morgens mit mir die Lieferung empfing und einbaute. Dazu wurde von der Umlenkstation „Luisenpark“ ein Teil des Unterbaus auseinandergenommen, die Klemme eingeschoben und direkt danach wieder zusammengebaut. Nun ist sie definitiv ein Eyecatcher und zeigt

zudem, wie nah wir mit unserem Modell am Original sind.

Auch unser Zusammenspiel mit den Forummitgliedern und unserem Hilferuf nach Unterstützung war vorbildlich. Und so gilt hier ein Dank allen Menschen, die uns mit hunderten von 3D-Druckteilen geholfen haben. Diese passen, funktionieren hervorragend, bisher ohne Bruch, und sehen auch noch klasse aus.



Abb. 6: Die Original-Doppelmayr-Klemme

Das Original der Klemme wird gerne von den Besuchern rundherum betrachtet. Es wiegt über 240 kg. Die Funktion wurde im Modell nachgebildet, natürlich im Rahmen des Möglichen im fischertechnik-Raster oder mit 3D-Druck in PETG.

Gondel-Gala

Nach Beginn der BUGA23 im April fand im Mai eine Veranstaltung zur Eröffnung der Ausstellung in der Indoorhalle (Gebäude 10) statt, in der unser Modell steht. Ein gutes Zusammenspiel unserer Mitglieder sorgte für eine tolle Präsentation der Anlage, auch wenn wir in diesem Sinne nur Zaungäste waren.

Dennoch besuchte uns Herr Schrag von fischertechnik und konnte so zum ersten Mal das Modell im Ganzen sehen.

Knapp einen Monat später dann unsere eigene Veranstaltung: Die „Gondel-Gala“, auf der unser Modell feierlich mit allen Sponsoren in Betrieb gehen sollte. Leider hat es die BUGA23 bisher nicht geschafft, in unserem Gewächshaus die Wärmeabzugsluken zu öffnen und die Verschattung zu schließen. Bei Temperaturen über 45°C auf Kopfhöhe ist Weiterarbeiten an der Anlage kaum möglich. Selbst der Betrieb ist problematisch, da wir an die kritischen Temperaturen der ungekühlten Elektronik kommen.

Ein zweites Hindernis war ein dreister Diebstahl mit Sachbeschädigung an der Steuerung. Zwei verbaute TXT wurden aus der Anlage geklaut, die Stecker herausgerissen. Unabhängig vom Sachschaden, der uns bereits durch Gönner ersetzt wurde, wirft uns dies mindestens drei Monate im Zeitplan zurück. Der moralische Schaden jedoch ist nicht zu beziffern. Dennoch: Ein phänomenales Zusammenspiel unserer Mitglieder am Wochenende vor der Gala gab uns einen großen Schub nach vorne. Rollenbatterien wurden im Akkord gefertigt. Mit sechs Leuten wurde das Seil eingehängt und zum Laufen gebracht. Die grafische Oberfläche wurde aufgespielt und konnte so unseren Sponsoren gezeigt werden.

Und so war die Gala ein voller Erfolg. Als Keynote-Speaker konnten wir den Erfinder und Gründer des Fördertechnik-Museums, Norbert Axmann, gewinnen, der über die Innovationskraft und -förderung in Deutschland referierte. Unsere Hauptsponsoren konnten sich den zahlreichen Besuchern präsentieren, die Nebensponsoren wurden mit einer Kabine bedacht, die in den nächsten Wochen frei gestaltet werden kann. Die Gönner und vor allem das Team wurden vorgestellt.

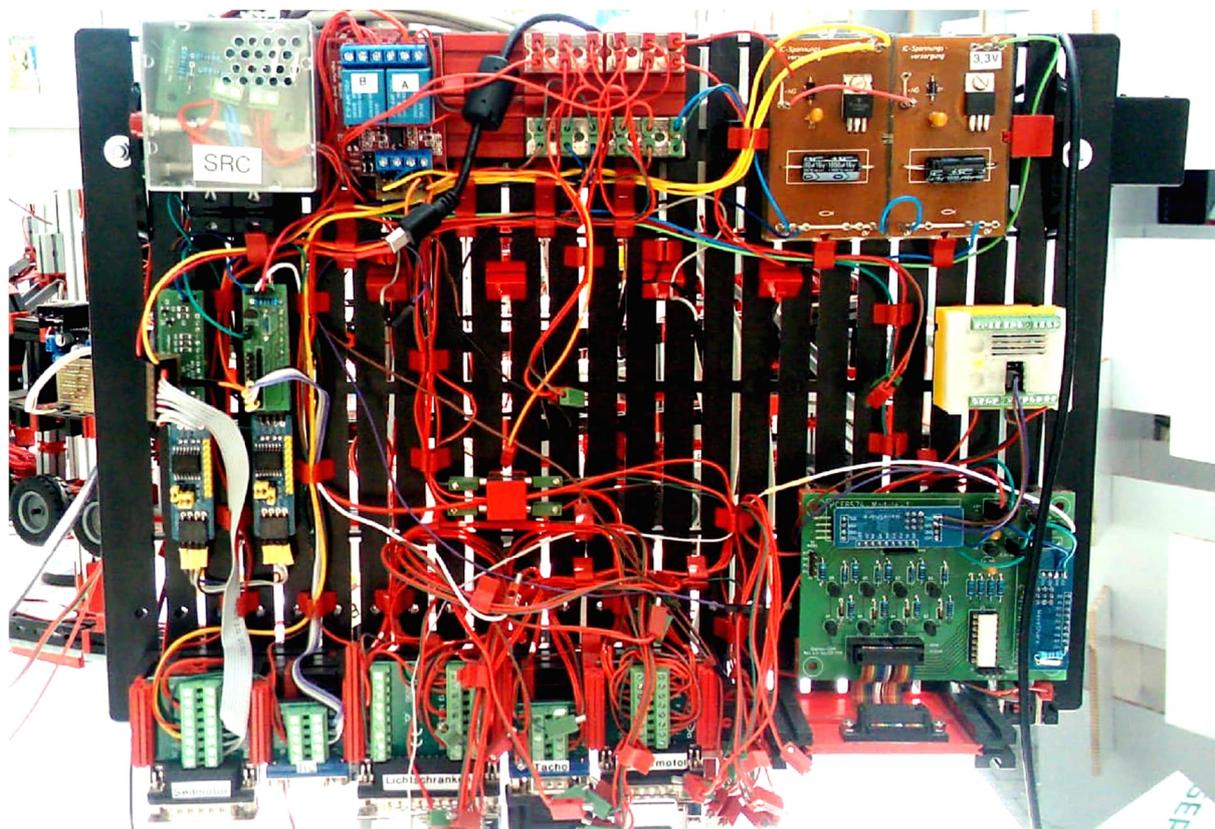


Abb. 7: Die Hauptsteuerung in der Antriebsstation in einem traurigen Zustand: Zwei TXT wurden daraus entwendet; die Kabel hängen ungeordnet herunter. Dabei hatte Tom in monatelanger Kleinarbeit die Kabelführung und den sauberen Aufbau bis in die Perfektion getrieben, um ein repräsentatives Ausstellungsstück zu haben.

Geheimsache

Im Vorfeld der Gondel-Gala hatte ich mir aber lange Gedanken gemacht, wie ich meinem Team für die 5.000 Arbeitsstunden der vergangenen 30 Monate danken könnte. Und so kam es, dass ich ein Geheimprojekt mit Arnoud „WhizzBizz“ van Delden startete – siehe seinen Artikel „fischertechnik-Figuren im Maßstab 1:10“ in dieser Ausgabe [4]. In neun Monaten perfekten Zusammenspiels gestalteten wir das Seilbahn-Team als fischertechnik-Männchen (und -Weibchen) im Maßstab 1:10, passend zu unserem Modell. Zusätzlich fanden wir einen Weg, jede Figur durch Kleidung, Frisur und Gesichtszüge zu individualisieren und als entsprechendes Mitglied erkennbar zu machen. Staunende Gesichter bei der Übergabe waren das Ergebnis.

Nach dem o. g. Diebstahl werden wir die Figuren nicht wie ursprünglich beabsichtigt auf der BUGA23 im Modell ausstellen, sondern sie erst im Fördertechnik-Museum in Sinsheim platzieren – ein Verlust wäre nicht zu ersetzen.

Quellen

- [1] [BUGA23 – Bundesgartenschau in Mannheim](#)
- [2] [YouTube-Kanal](#) des Seilbahn-Projekts
- [3] [Fördertechnik-Museum Sinsheim](#), (direkt neben dem Technik-Museum)
- [4] Arnoud van Delden: *fischertechnik-Figuren im Maßstab 1:10*. ft:pedia 2/2023 (in diesem Heft).
- [5] Kontakt zur Projektgruppe „Seilbahn“: ft.seilbahn@gmail.com



Abb. 8: Strahlende Gesichter in Groß aus Kunststoff, passend im Modellmaßstab 1:10 und natürlich in noch größer, Maßstab 1:1 bei unseren Projektmitgliedern. Arnoud überrascht uns alle bei 45°C.

3D-Modellentwurf

fischertechnik-Figuren im Maßstab 1:10

Arnoud van Delden

Die originale fischertechnik-Figur ([30460](#)) gibt es seit 1982. Obwohl sie seitdem in verschiedenen Farbvarianten und mit möglicher Kopfbedeckung aufgetaucht ist, weist sie keine große Vielfalt in Aussehen, Alter, Geschlecht oder Hautfarbe auf. Bisher gibt es nur zwei Gesichtsausdrücke und Frisuren. Mit Hilfe eines 3D-Druckers lässt sich dagegen etwas unternehmen, dachten sich im Jahr 2022 mehrere Leute im ftc-Forum [1]. Wie diese Suche schließlich zu einem unerwarteten Beitrag zum Seilbahnprojekt für die Bundesgartenschau Mannheim (BUGA23) führte, könnt ihr hier lesen.

3D-Drucken

Die klassischen fischertechnik-Figuren und die selbstgedruckten Varianten in Abb. 1 sind 8,5 cm hoch. Bei einer durchschnittlichen menschlichen Körpergröße von 170 cm haben sie daher einen Maßstab von etwa 1:20.



Abb. 1: Bunte Figurenvielfalt

Die Forumdiskussion führte schließlich zu viel Kreativität und verschiedenen zusätzlichen Figuren. Im Clubmagazin des niederländischen fischertechnikclubs [2] habe ich zusammen mit Jeroen Regtien beschrieben, wie wir dazu inspiriert wurden, Arme, Körper und Frisuren zu drucken, um die Vielfalt der Figuren zu erhöhen. Ich selbst ging gerne noch weiter, als nur einen Haar-

schnitt zu ersetzen, und habe zum Beispiel auch einen Clown, jemanden mit gebrochenem Bein, einen Menschen im Rollstuhl und ein paar Figuren mit spezifischer Berufskleidung oder Attributen angefertigt.

3D-Modellierung

Abb. 2: Ein Bild aus dem Video mit Mr. Lemniskate

Zufälligerweise habe ich, bevor die Diskussion über das Fehlen einer weiblichen fischertechnik-Figur im Forum aufkam, bereits nützliche „3D-Vorarbeiten“ geleistet. Vor einiger Zeit wollte ich diese Figur in der 3D-Animation in einem Video [3] zur Unterstützung eines Artikels über Lemniskaten zeigen [4]. Zu diesem Zweck hatte ich damals bereits alle Einzelteile der origina-

len fischertechnik-Figur selbst am Computer modelliert. Obwohl das Designprogramm „fischertechnik designer“ [5] bereits maßgenaue 3D-Modelle enthielt, erwiesen sich diese in der Praxis als unzureichend detailliert oder für die Animation in Blender [7] nicht nutzbar. Im Video musste beispielsweise den Schultern seitlich etwas mehr Bewegungsfreiheit gegeben werden als im Original.



Abb. 3: Die einzelnen 3D-Teile sind unendlich vielfältig

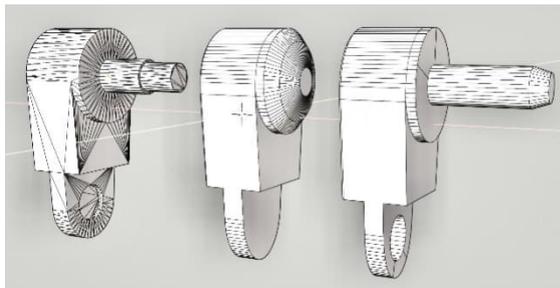


Abb. 4: Die verschiedenen Anwendungen haben unterschiedliche Designanforderungen

Als ich die Figur später tatsächlich aus dem 3D-Drucker ausrollen lassen wollte, hatten die für die Animation verwendeten Modelle erneut in einigen Punkten Mängel. So verfügten die Oberarme beispielsweise noch nicht über die Drehzapfen und auch das digitale Modell der Hüftpartie, die in der Animation lediglich am Unterkörper befes-

tigt wurde, musste verfeinert werden. Abb. 2 zeigt das Modell des „fischertechnik Designers“ des Oberarms, mein eigenes Animationsmodell und die 3D-druckbare Variante.

Das Seilbahnprojekt

Seit März 2010 hält uns Tilo Rust im ftc-Forum [8] und in Artikeln in der ft:pedia [9, 10, 11, 12, 13, 14] über den Fortschritt des fischertechnik-Seilbahn-Projekts auf dem Laufenden. Nachdem er die mit dem 3D-Drucker erzielten Ergebnisse gesehen hatte, veröffentlichte er eine Anfrage im Forum: Schließlich wäre es toll, wenn das fischertechnik-Modell der Seilbahn auf der BUGA23 [15] mit 3D-gedruckten Figuren ausgestattet werden könnte. Da die Seilbahn jedoch im Maßstab 1:10 gebaut ist, müssten die Proportionen der Figuren dafür verdoppelt werden.



Abb. 5: Kopf und Hals im Maßstab 1:10

Mein ursprüngliches Angebot, einige Deko-Figuren für das Seilbahn-Projekt im Maßstab 1:10 zu drucken, nahm bald eine unerwartete Wendung. Bereits vor unserem ersten persönlichen Treffen auf der Südconvention in Sinsheim im Jahr 2022 äußerte Tilo den Wunsch, jedes Mitglied seines Teams mit einer personalisierten Figur zu überraschen. Eine tolle Idee, aber auch eine ziemliche Herausforderung, denn neben der Kleidung sollten vor allem die Gesichter und die Frisur jeder Figur möglichst gut erkennbar sein. Das bedeutete, dass jedes Mitglied des Teams zu einer

einzigartigen fischertechnik-Figur geformt werden musste. Die Figuren sollten als Überraschung bei der feierlichen Präsentation des Modells Anfang Juni 2023 präsentiert werden. Bis dahin wäre absolute Geheimhaltung geboten.

Obwohl mir die Größe meines möglichen Beitrags recht schnell klar wurde, war ich so begeistert, dass ich sofort zustimmte, die 3D-Modelle im Maßstab 1:10 zu vergrößern und loszulegen. Auf der Südconvention konnte ich Tilo daher bereits, natürlich unter größter Geheimhaltung, einige frühe Probeglieder und (noch unverzierte) Köpfe zeigen.

Natürlich waren Fotos notwendig, um die persönlichen Eigenschaften jedes Teammitglieds bestmöglich darzustellen. Tilo hat eine Methode gefunden, diese einzusammeln, ohne Verdacht zu erregen. Er wollte versuchen, die Referenzfotos jedes Teammitglieds unter dem Vorwand zu erhalten, dass diese für die Zugangskarten zum BUGA23-Gelände notwendig seien.

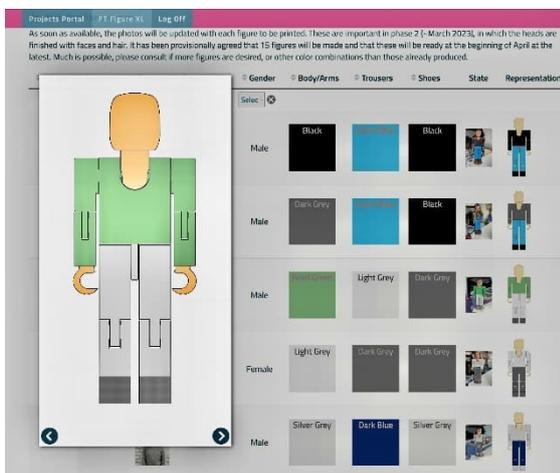


Abb. 6: Konfigurationsschnittstelle mit Benachrichtigung über den Produktionsfortschritt

Mittlerweile hatte ich in einem geschützten Bereich hinter meiner Website eine spezielle Projektseite für die interne Kommunikation aufgebaut. Hier konnte der Farbkonfigurationsvorschlag für jedes Teammitglied konkretisiert und der Fortschritt

überwacht werden. Um die Überraschung nicht vorzeitig zu stören, wurden diese Informationen selbstverständlich mit einem Passwort geschützt. In den folgenden Monaten machte Tilo Vorschläge für die Farben der Kleidung und weitere spezifische persönliche Eigenschaften jeder Figur, während ich meine Sammlung an Farbfilamenten [16] für den 3D-Drucker erweiterte und die „Produktionsdruckbarkeit“ der 3D-Modelle verbesserte.

Die 3D-Designs

Verdoppelt man alle Maße der originalen fischertechnik-Figur ergibt sich eine 3D-druckbare Figur von 17 cm Höhe. Insgesamt benötigt mein Drucker ohne die manuelle Nachbearbeitung mehr als zehn Stunden, um eine solche Figur zu drucken. Eine so große Puppe im Maßstab 1:10 wiegt schließlich etwa 85 Gramm.

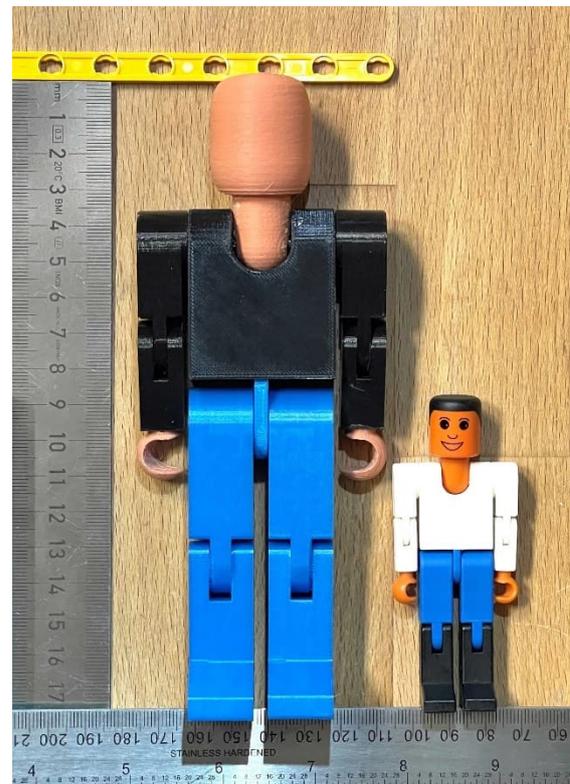


Abb. 7: Groß und Klein brüderlich Seite an Seite

Vor dem Drucken der Figuren habe ich verschiedene Materialien ausprobiert. PLA erwies sich als nützlicher als PETG [6].

Allerdings sind alle verfügbaren Filamente härter und spröder als der Kunststoff der originalen fischertechnik-Figuren. Dadurch unterscheidet sich der Rotationswiderstand in den verschiedenen 3D-gedruckten Gelenken vom Original. Um zu verhindern, dass die Figur beim In-Szene-Setzen unter ihrem eigenen Gewicht zusammenbricht, musste mit den Durchmessern und der Form der Drehpunkte experimentiert werden. Dies war insbesondere beim kombinierten Bedrucken („Print-in-Place“) von Unter- und Oberschenkel wichtig. Abschließend wurde am Kniegelenk einseitig ein leicht ovaler Drehpunkt geschaffen und der Drehzapfen der Schulter am Schaft mit einer Rippe versehen, die bei Bedarf nachträglich mit einer Feile aufgeraut werden konnte.



Abb. 8: Die Serienproduktion kann beginnen

Wie in Abb. 8 zu sehen ist, konnten nach der Skalierung des ursprünglichen 3D-Designs die ersten Experimente mit dem Drucken dieser „großen“ Figuren beginnen.

Die Gesichter

Bei den Figuren im Maßstab 1:20 hatte ich bereits einige davon mit Händen und Köpfen in anderen Hautfarben oder inklusive Frisur gedruckt. Natürlich mussten diese nachträglich mit einem Gesicht versehen werden. Damit die endgültige Figur als fischertechnik-Figur erkennbar bleibt, sollte dieses Gesicht idealerweise grafisch so weit wie möglich mit dem stilisierten Standardgesicht der ursprünglichen fischertechnik-Figur übereinstimmen. Es stellte sich

jedoch heraus, dass das Anbringen eines solchen Gesichts nicht so einfach war.

Aufgrund des schichtweisen Aufbaus der 3D-gedruckten Köpfe habe ich beim Zeichnen mit einem Marker nur wenig erfolgreiche Ergebnisse erzielt. Wegen der Kapillarität zwischen den Schichten und der rauen Oberfläche waren scharfe Linien kaum möglich. Gleichzeitig liefen diese seitlich schnell und unkorrigierbar aus. Aufgrund dieser Oberflächenstruktur ist es im 3D-Druck aber auch nicht möglich, scharfe Linien zu erzeugen, die dann mit Farbe ausgewaschen werden können, wie dies bei gravierten Linien möglich ist. Auch hier machen die Saugfähigkeit der einzelnen Schichten und die raue Oberfläche einen Strich durch die Rechnung. Das Ergebnis ist fleckig und verschwommen. Eine Gravur der Gesichter mit einem Laser ist ebenfalls nicht möglich: Das 3D-Filament schmilzt bei relativ niedriger Temperatur und verfärbt sich nicht schwarz. Letztlich habe ich deshalb die Münder und Augen der früheren Figuren im Maßstab 1:20 mit einem sehr feinen Pinsel gemalt. Sehr arbeitsintensiv und natürlich bei einer Kleinserienproduktion (16 einzigartige Figuren im Maßstab 1:10), wie sie mir aufgetragen wurde, nicht sehr attraktiv.

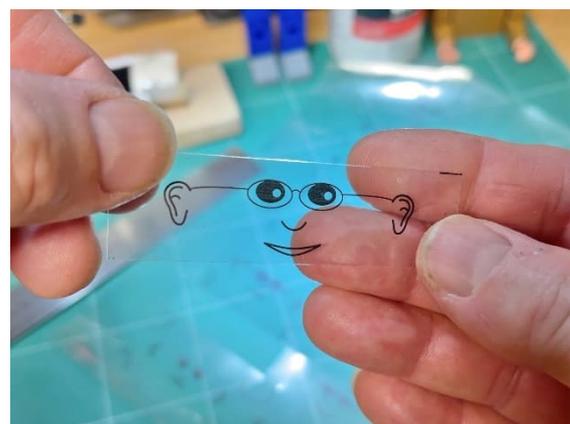


Abb. 9: Experimente zur Bildübertragung aus einer transparenten Overheadfolie

Die nützlichste Lösung wäre ein Weg, um die mit dem Laserdrucker gedruckten Gesichter auf die 3D-gedruckten Köpfe zu

übertragen. Da die originalen fischertechnik-Figuren keine Brille, Bärte, Schnurrbärte etc. haben, wurde auf eine 3D-Anfertigung davon verzichtet. Haare mit Volumen könnten möglicherweise Teil des 3D-Drucks des Kopfes einer Figur werden.

Nachdem ich Körper und Gliedmaßen in den verschiedenen vereinbarten Farben (für jede Person individuell zusammengestellt) gedruckt hatte, beschloss ich, mit der Anpassung der Gesichter zu experimentieren. Dazu habe ich stilisierte Designs der gewünschten Personen (ohne Gesichtshaarung, aber mit eventueller Brille) auf einem A4-Blatt zusammengestellt, das auf dem Laserdrucker ausgedruckt werden konnte.

Anhand von Videos auf YouTube experimentierte ich zunächst mit der Übertragung des lasergedruckten Gesichts über ein Gelmedium. Im Idealfall würde dadurch nur der Toner auf den 3D-Kopf übertragen. Obwohl ich mit verschiedenen Trägermaterialien und Papieren herumgespielt habe (Calque-Papier, Plastik-Overhead-Folie und die wachsartige Oberfläche der Trägerfolie von Klebeetiketten), wurden damit auch keine guten Ergebnisse erzielt. Ich habe sogar den speziellen Clip von Abb. 10 entwickelt, bei dem das Bild fest auf der kreisförmigen Oberfläche gehalten wurde, während es trocknete.



Abb. 10: Experimente mit maßgeschneiderter Klemme während des Trocknungsvorgangs

Die besten Ergebnisse lassen sich natürlich erzielen, wenn sich der Toner leicht vom

Träger lösen lässt. Es wird dann jedoch schwierig sein, das Bild ohne Beschädigung zu positionieren. Darüber hinaus waren die transparenten Transfermaterialien (z. B. Overheadfolien) zwar zu positionieren, aber kaum luftdurchlässig. Dadurch dauerte die Trocknung zu lange und die Gefahr einer Beschädigung beim Entfernen des Trägers stieg. Daher schien dieses Verfahren für die im Auftrag geforderte Halbserienfertigung wenig geeignet zu sein.



Abb. 11: Unbefriedigende Ergebnisse mit „Liquitex Matte Gel Medium“

„Water Slide Decals“

Obwohl die Versuche, den Toner der spiegelverkehrt gedruckten Gesichter mit einem Gelmedium auf die Köpfe zu übertragen, gescheitert waren, wäre eine Lösung ideal, bei der die Gesichter im Voraus gestaltet werden können. Wäre der Kopf, wie das Originalteil, rein zylindrisch, könnte transparentes Aufklebermaterial vielleicht eine Lösung sein. Allerdings war zu erwarten, dass dies aufgrund des Glanzes und der Dicke der Folie kein ansprechendes Endergebnis liefern würde. Außerdem hatte ich den Eindruck, dass der Kleber des Materials mit der Zeit ausdünsten und sich verfärben könnte. Und vielleicht wäre auch die Haftung am 3D-Filament wegen des steifen Materials bei einem so kleinen Durchmesser auf Dauer nicht ausreichend. Dies könnte im wahrsten Sinne des Wortes zu einem Gesichtsverlust führen.

Dann erinnerte ich mich an die Transfers, die früher zu den Plastikflugzeugbausätzen gehörten. Man schneidet sie aus und lässt sie kurz in einer Untertasse mit Wasser einweichen. Nach einiger Zeit konnte man das Bild vorsichtig vom Papier auf das Modell schieben. Aufgrund der wässrigen Leimschicht ließ sich das Bild kurzzeitig recht gut positionieren. Für Modellbauer, die ihre eigenen Abziehbilder herstellen wollen, gibt es spezielles Papier für Laser- und sogar für Tintenstrahldrucker. Diese speziell präparierten Papierbögen bestehen aus einer wasserdurchlässigen Papierträgerschicht, auf die ein wasserlöslicher Leim und eine gelartige Deckschicht aufgetragen werden. Man kann mit einem normalen Laserdrucker direkt auf die oberste Schicht drucken.



Abb. 12: Anbringen eines „Water Slide Decal“

Diese Methode war sofort viel erfolgreicher. Aufgrund der A4-Größe der (relativ teuren) Abziehbilder ist es sinnvoll, möglichst viele Gesichter auf einem Blatt unterzubringen. Anschließend kann jedes gewünschte Gesicht möglichst nah am Motiv ausgeschnitten werden. Gelingt es dabei, nur relativ kleine Löcher in das Blatt zu schneiden, kann es anschließend mühelos erneut durch den Laserdrucker geführt werden, um etwaige „fehlgeschlagene“ Übertragungsversuche oder nachträgliche Verbesserungen erneut zu drucken.

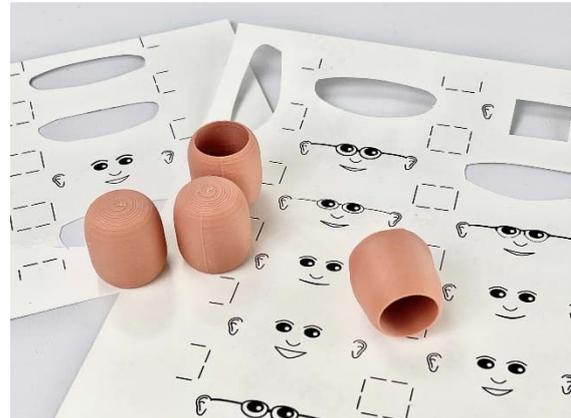


Abb. 13: Transferblatt und Köpfe

Den ursprünglich rein zylindrischen Kopf der fischertechnik-Figur hatte ich bereits zu einem früheren Zeitpunkt etwas stärker gebogen. Die 3D-gedruckten Köpfe hatten daher eine leichte Krümmung, die zu kleinen Falten im „Aufkleber“ führen konnte. Im Handel gibt es einen speziellen „Weichmacher“, mit dem sich die Transfers geschmeidig machen lassen. Nach einigen Recherchen in Modellforen lieferte auch klarer weißer Essig hervorragende Ergebnisse.

Am Ende erwies es sich als am besten, den Kopf leicht mit einem verdünnten Gelmedium zu grundieren und den Transfer in klarem, weißem Essig statt in Wasser einzuweichen. Unmittelbar vor dem Auftragen kann die Oberfläche zusätzlich angefeuchtet werden, um die Positionierung zu erleichtern. Hier ist etwas Vorsicht geboten, da sich der Druckertoner auf der Außenseite des Aufklebers befindet! Das Glätten ging, nach etwas Übung und mit der nötigen Vorsicht, gut mit einem Pinsel mit Essig oder Wasser. Nach dem Trocknen sitzen die Aufkleber fest auf dem Kopf. Eventuell beim Auftragen entstandene Schäden am Bild können noch mit einer Stecknadel mit schwarzer Acrylfarbe (und einer sehr ruhigen Hand) ausgebessert werden. Dann ist der Kopf bereit für den letzten Schritt: das Handbemalen der Frisur und eventueller weiterer Gesichtshaarung.

Die verschiedenen Köpfe

Nachdem es nun eine brauchbare Methode zum Auftragen der verschiedenen Gesichter gab, war es an der Zeit, einen Blick auf die unterschiedlichen Frisuren der Figuren zu werfen. Die Frisuren der verschiedenen Menschen variierten natürlich erheblich und es musste der Schluss gezogen werden, dass der ursprünglich entworfene Kopf nur für Menschen mit sehr wenigen (oder keinen) Haaren auf dem Kopf verwendet werden konnte – zum Beispiel Tilos. Im Team hingegen gab es mehrere Personen mit sehr ausgeprägten und voluminösen Frisuren. Aufgemalt auf einen flachen Kopf wären diese Frisuren wahrscheinlich nicht erkennbar oder unglaublich.

Da die finalen Figuren im Maßstab 1:10 der vorhandenen, stilisierten fischertechnik-Figur möglichst nahe kommen sollten, schied ein Aufkleben von Draht, Filz oder Watte als Haarersatz aus. Lose 3D-gedruckte „Perücken“ werden schnell sehr dünn und lassen sich nur schwer drucken. Die Schlussfolgerung war, dass es für diese Frisuren besser wäre, einen individuellen Kopf mit zusätzlichem Volumen für die Haare anzufertigen. Für einige (männliche) Figuren genügte ein universelles Grundmodell, das durch die richtige Bemalung hinreichend personalisiert werden konnte. Allerdings sollten die Köpfe von drei Personen in der 3D-Phase individuell angefertigt werden. Selbstverständlich musste bei der Gestaltung berücksichtigt werden, dass nach dem Zusammenbau der gesamten Figur lange Haarsträhnen über die Schultern fielen.

Auf dem Druckbett eines 3D-Druckers können nur ebene Flächen gedruckt werden. Gebogene oder runde Flächen werden mit einer nachträglich abnehmbaren Stützstruktur bedruckt. Diese Oberflächen bedürfen einer Nachbearbeitung und werden meist nicht so glatt. Mit solchen Stützkonstruktionen konnten die maßgefertigten Köpfe oft kopfüber gedruckt werden. Einige XL-

Figuren tragen Deckel. Ich habe diese ohne Stützstruktur flach auf dem Druckbett gedruckt. Den gewölbten Mützenschirm habe ich erhalten, indem ich den flachen Mützenschirm anschließend vorsichtig mit einem kleinen Piezobrenner erhitze und dann von Hand umformte. Im Nachhinein betrachtet hätte diese Technik beispielsweise auch zur Herstellung gebogener Haarsträhnen genutzt werden können.

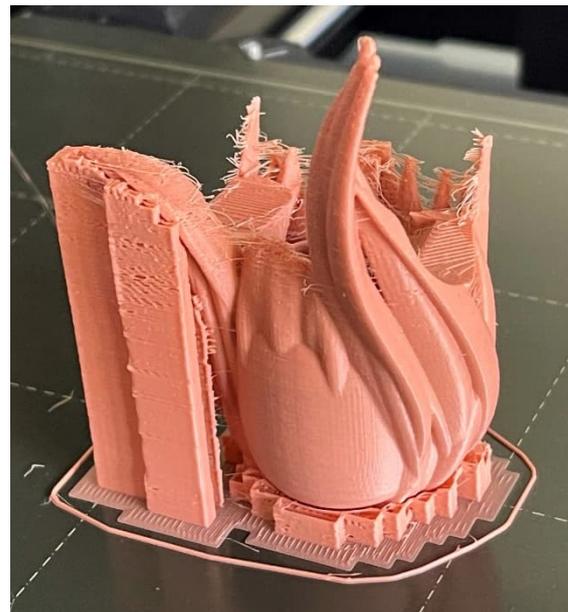


Abb. 14: Unmittelbar nach dem Druck ist der endgültige Kopf kaum noch zu erkennen

Dekoration mit Acrylfarbe

Nach der manuellen Verzierung der Haare und weiterer Gesichtsbehaarung wie Bart und Schnurrbart konnten die Figuren zusammengebaut werden. Anschließend wurde jede Figur mit einem matten transparenten Lack besprüht. Dadurch wurden die Handmalereien fixiert und mögliche Glanzunterschiede der verschiedenen Materialien beseitigt.

Wie oben bereits beschrieben ist 3D-Filament spröder und härter als der Kunststoff, aus dem die originalen fischertechnik-Figuren bestehen. Dies wurde bereits bei der Konstruktion berücksichtigt, allerdings bewegten sich die Gliedmaßen manchmal noch zu leichtgängig. Normalerweise reich-

te es aus, eine kleine Menge Kontaktkleber in die Kniekehle oder in die Öffnungen für die Arme im Nacken aufzutragen, um wieder genügend Reibung zu erhalten. Um die Puppen im Seilbahnmodell auf der BUGA23 zu positionieren habe ich ihnen kleine Fußstützen verpasst.

„Gondel-Gala“

Am Samstag, den 8. Juni 2023, war es endlich so weit. Im Rahmen der angekündigten „Gondel-Gala“ auf der BUGA23 sollten die Figuren den anwesenden Teammitgliedern überreicht werden. Trotz der hohen Temperaturen an diesem Tag hatten zahlreiche Interessenten und Sponsoren den Weg in die Präsentationshalle im Luisenpark gefunden. Nach einer Einführung durch Tilo und einigen Keynotes von Sponsoren (u. a. fischertechnik, Doppelmayr-Garaventa und das Fördertechnik-Museum) wurde Tilo von seinem Team gebührend gedankt. Danach konnten die kleinen Figuren end-

lich von ihren großen Vorbildern adoptiert werden. Die Überraschung darüber war natürlich groß.

Abschluss

Die Herausforderungen, denen ich gegenüberstand, habe ich auf einer Projektseite auf meiner Website zusammengefasst [17]. Hier ist auch ein Link zum Video über die Entstehung der Figuren. Eine weitere Reportage wird auf dem YouTube-Kanal des Seilbahn-Projekts erscheinen.

Das Projekt bot einzigartige Möglichkeiten, mit großformatigen fischertechnik-Figuren zu experimentieren. Mit mehr als 200 Stunden Druck und mindestens 50 Stunden manueller Dekoration war es eine ziemliche Herausforderung. Aber es hat auch viel Spaß gemacht. Verschiedene Probleme wurden nach und nach überwunden und die Lösung für die personalisierten Frisuren und Anwendung der Gesichter ist letztendlich sehr nützlich.



Abb. 15: Haare, Mützen, Bärte und ein einzelner Ohrring



Abb. 16: Tilo mit seinem gesamten Seilbahn-Team



Abb. 17: Tilo mit seinem Figuren-Exemplar

Als ich alle Herausforderungen gemeistert hatte, sah ich, wie die Figuren und das Team wuchsen. Wir halten es für einen Zufall, dass die Vorlaufzeit des Projekts von der Konzeption (Idee) bis zur Geburt (der „Gondel-Gala“) dieser 16 einzigartigen Figuren neun Monate dauerte. Durch den

Besuch einiger fischertechnik-Meetings lernte ich in dieser Zeit verschiedene Teammitglieder kennen. Unnötig zu erwähnen, dass ich bis zur letzten Minute mehrere Male darum gekämpft habe, das Geheimnis zu bewahren.

Die hergestellten Figuren sind zweifellos ein einzigartiges Geschenk. Ich werde mich in Zukunft sicher noch öfter mit der Gestaltung personalisierter fischertechnik-Figuren befassen. Diese Figuren allerdings müssen erst einmal bis zum 8. Oktober den Härte-test auf der BUGA23 bestehen, bevor sie einen ruhigen Lebensabend im Fördertechnik-Museum Sinsheim genießen können.

An dieser Stelle möchte ich mich, nicht im Maßstab 1:10, sondern im Maßstab 10:1, bei Tilo für seine großartige Idee bedanken, seine Begeisterung, mit der wir einige Herausforderungen des Projekts lösen konnten, das Korrekturlesen und Verbessern dieses Beitrags und vor allem für seine

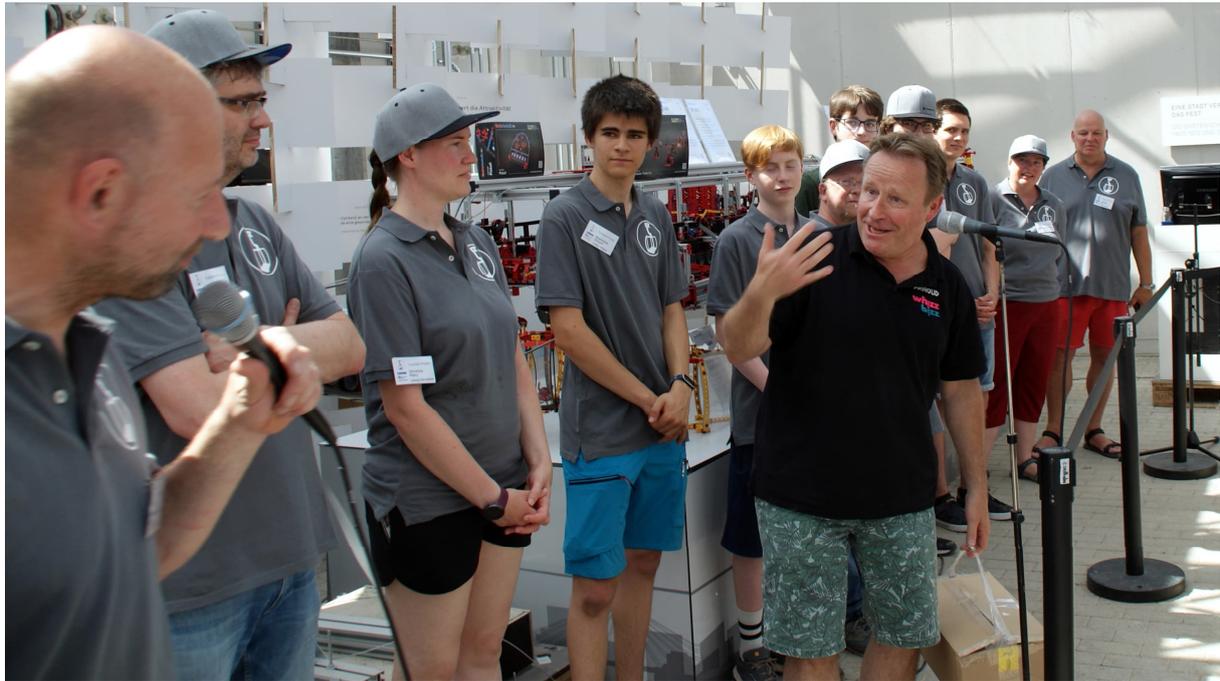


Abb. 18: Arnoud bei der Übergabe an die anwesenden Mitglieder des Seilbahn-Teams

Gastfreundschaft und Unterkunft während meines Aufenthalts in Mannheim.

Quellen

- [1] *fischertechnik Puppensortiment*: Diskussion im [ftc-Forum](#), 2022.
- [2] Clubmagazin des niederländischen fischertechnikclubs, Nr. 2/2022.
- [3] Arnoud van Delden: *The Infinite Lemniscate*. Auf [YouTube](#), 2022.
- [4] Arnoud van Delden: *Die unendliche Lemniskate*. [ft:pedia 2/2021](#), S. 36–44.
- [5] Software „[fischertechnik Designer](#)“:
- [6] Arnoud van Delden: *Neue ft-Teile selbst gemacht – 3D-Druck (7)*. [ft:pedia 1/2023](#), S. 33–42.
- [7] [Blender](#) (3D-Modellierungs- und Animationssoftware)
- [8] *Community-Projekt: Seilbahn*. Im [ftc-Forum](#), 2020.
- [9] Tilo Rust: *Großprojekt Seilbahn (Teil 1): Von der Idee zum Kick-Off*. [ft:pedia 1/2021](#), S. 56–65.
- [10] Tilo Rust: *Großprojekt Seilbahn (Teil 2): Erste Elemente*. [ft:pedia 2/2021](#), S. 45–53.
- [11] Tilo Rust: *Großprojekt Seilbahn (Teil 3): Die Stationen*. [ft:pedia 3/2021](#), S. 28–37.
- [12] Tilo Rust: *Großprojekt Seilbahn (Teil 4): Fundamentale Arbeiten*. [ft:pedia 1/2022](#), S. 37–43.
- [13] Tilo Rust: *Großprojekt Seilbahn (Teil 5): Bewegung*. [ft:pedia 3/2022](#), S. 10–18.
- [14] Tilo Rust: *Großprojekt Seilbahn (Teil 6): Reifenförderer*. [ft:pedia 4/2022](#), S. 41–51.
- [15] Bundesgartenschau Mannheim 2023 gGmbH: [BUGA23](#)
- [16] Arnoud van Delden: *Available filament*. Übersicht der verfügbaren Filamente, auf [whizzbizz.com](#).
- [17] Arnoud van Delden: *Personalized 3D printed figures*. Projektseite auf meiner Website auf [whizzbizz.com](#).

Modell

POD-Antrieb für ein fischertechnik-Schiff

Gerhard Birkenstock

Ein POD- oder Propellergondel-Antrieb ist ein schwenkbarer Propeller für Wasserfahrzeuge und Luftschiffe. Er wurde 1951 von Friedrich Pleuger und Friedrich Busmann erfunden und 1955 patentiert [1]. Mit ein wenig 3D-Druck lässt sich der fischertechnik-Katamaran damit zu einem rasanten und wendigen Wasserfahrzeug „pimpen“.

Der Aufbau

Ein Schiff mit fischertechnik bauen – so richtig, mit Fernsteuerung! Den Rumpf gibt es im Sortiment ([155903](#)). Er besteht aus Styropor und hat die Form eines Katamarans – genau das Richtige für mein Vorhaben und auch noch von guter Qualität. Da die Schiffsschraube unten herausragen wird, musste ich für den Rumpf eine Aufnahme bauen, damit die Schraube und der POD (übliche Bezeichnung für einen Propellergondel-Antrieb) nicht beschädigt werden. Mit ein paar Grundbausteinen und den Streben aus der Statik war das schnell gelöst. Das so entstandene Untergestell ist in Abb. 1 zu sehen.

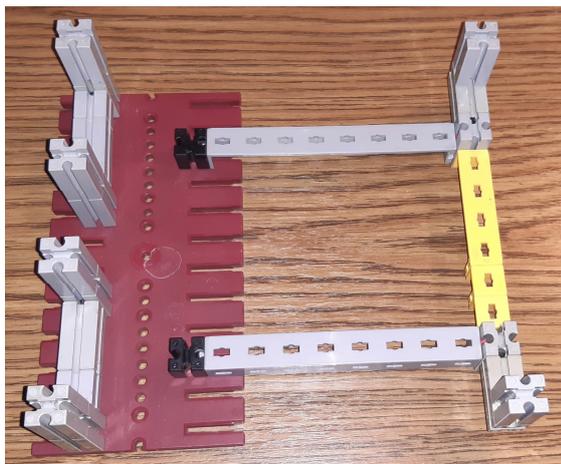


Abb. 1: Rumpf-Ablage

Beim Bau hatte ich noch keinen 3D-Drucker; auch die Software für ein solches Vorhaben war nicht vorhanden. Daher habe

ich mich an einen Freund gewendet. Als ich ihm darstellte, was mir vorschwebte, ging er gleich mit Eifer daran. Es war faszinierend, wie er meine Vorstellung in CAD umsetzte – super schnell (er macht so etwas täglich). Die STL-Datei habe ich von ihm auf den USB-Stick geladen und mitgenommen und nach wenigen Tagen wurden die Teile von einem Dienstleister geliefert. Einfach toll, was heute mit 3D-Druck alles möglich ist.

Trotz der hohen Wiedergabe-Qualität musste ich mit Feile und Bohrer noch einmal Hand anlegen. Das Ergebnis war aber richtig gut. Die Wellen und Zahnräder wurden direkt aus dem fischertechnik-Sortiment genutzt: Es sind die Rastachse 30 mit Kegelzahnrad ([35061](#)), die Rastachse 90 ([35066](#)) mit den steckbaren Enden sowie das Rast-Kegelzahnrad ([35062](#)).



Abb. 2: 3D-Druckteile

Die Schiffsschraube wurde auch per 3D-Druck hergestellt. Obwohl die Schraube nicht die optimale Form bekommen hat, leistet sie im Wasser gute Dienste. Die Nabe der Schraube hat per Zeichnung direkt die Aufnahme der Rastachsen von fischertechnik bekommen. Das hat ohne Nachar-

beiten direkt gepasst. Somit sind wir sogar in der Geometrie unserer Formteile in der Norm von fischertechnik geblieben.

Leider musste ich ein größeres Loch in die rote Bodenplatte (32859) des Bootes bohren. Da die Platte aber jederzeit ausgetauscht werden kann, war das kein wirklicher Hinderungsgrund. Auch das Styropor um die Durchgangsstelle musste weichen: Der Propeller braucht Platz (Abb. 3).



Abb. 3: POD im Rumpf, von unten

Wenden wir uns nun der Oberseite zu. Hier sind zwei Bewegungen auf mechanischem Wege nach unten zu übertragen. Das eine ist der eigentliche Antrieb, also die Drehung der Schiffsschraube. Sie wird immer nur in eine Richtung drehen. Eine Schubumkehr wird nicht mit der Drehrichtung der Schraube durchgeführt sondern durch ein Drehen der Gondel umgesetzt. Daher muss die nach unten führende Achse vom Zahnrad getrennt sein; sie darf also keine kraftschlüssige Verbindung haben.

Gleichzeitig soll aber das Z30 (Abb. 4) fest sitzen. Somit habe ich bei angezogener Nabenmutter (31058) an der Flachnabenzange (35031) einen 3,2 mm-Bohrer durch das Wellenloch per Standbohrmaschine geführt. Das funktionierte gut: Die Achse kann trotz geschlossener Klemmung weiterhin gedreht werden.

Die im 3D-Druck erzeugte Gondel unter dem Rumpf muss aber nun noch mit dem Z30 auf der Oberseite verbunden werden. Das habe ich mit einem sehr dünnen Stahlstab erreicht.

In Abb. 2 ist dieser zu erkennen. Er schaut aus dem weißen Kunststoff genau neben der schwarzen Achse heraus. Ein kleines Loch auf der Unterseite in der Z30-Klemmchse gibt den formschlüssigen Zusammenhalt der Anordnung.

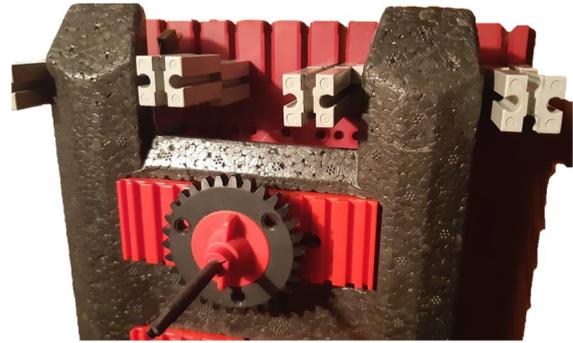


Abb. 4: Achse und Drehteller

Die wichtigen mechanischen Details sind nun erklärt. Wenden wir uns jetzt der Elektronik und der Ansteuerung zu. Es soll ein Schiff werden, wie angekündigt mit Fernbedienung. Dazu habe ich den kleinen Funkempfänger von fischertechnik genutzt.

Eine Energieversorgung muss natürlich auch noch her. Eine Spannung von mindestens 6V ist erforderlich: Die Motoren sollen vernünftig laufen und der Empfänger möglichst tadellos funktionieren. Das kleine Batteriegehäuse für einen 9V-Block empfand ich als geeignet (Abb. 5).

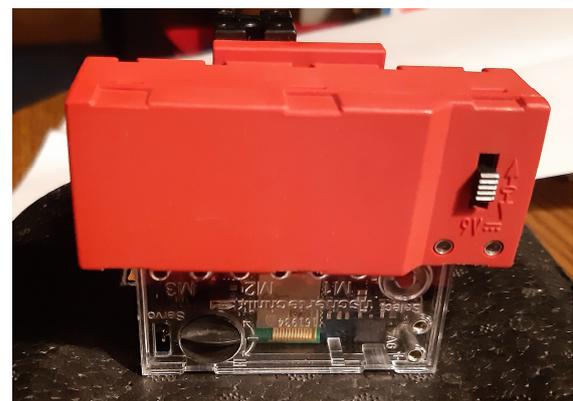


Abb. 5: Fernsteuerung und Akku

Hier ist nicht viel elektrische Energie gespeichert. Mit einem 9V-Akku kann ich gut 15 Minuten fahren. Für die Versuche und den Spaß war das ausreichend.

Einen Schaltplan erspare ich mir hier. Es ist sehr einfach: Die Spannungsversorgung an den Empfänger anschließen, ein Kanal für den großen Motor als Vorschub und ein zweiter Kanal für den kleinen Motor zur Drehung der Gondel. Der Kanal mit der Drehung wird in beiden Richtungen verwendet. Der Vortrieb wird nur in einer Richtung genutzt. Mehr an Verschaltung gibt es nicht.

Den großen Motor habe ich für den Vortrieb gewählt: Ich wollte gerne etwas mehr Power haben. Um die kleine Gondel zu drehen ist ein S-Motor ([32293](#)) mehr als ausreichend. In Abb. 6 ist die Konstellation der beiden Motoren zu erkennen.

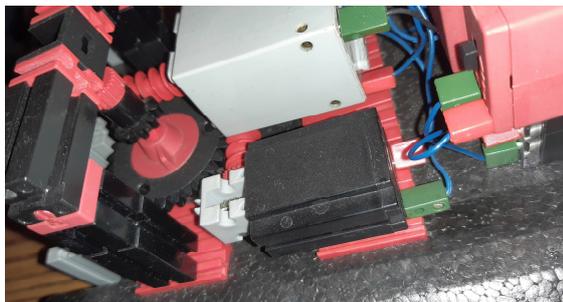


Abb. 6: Anordnungen der Motoren, Steuerbord

Um gut voran zu kommen habe ich ein Z10 für die Schraube verwendet. Die Schnecke vom großen Motor bringt die Schraube auf die genau richtige Drehzahl.

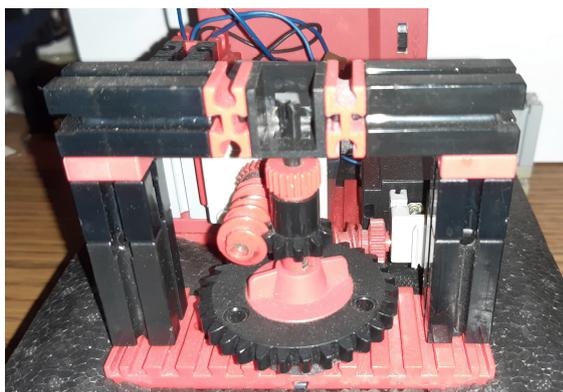


Abb. 7: Eingriff der Motorschnecken

Die Untersetzung über das Z30 zur Verstellung der Gondel könnte etwas schneller reagieren. Dagegen spricht jedoch die benötigte Präzision beim Einstellen der Geradeausfahrt – man ist auch so ständig

am Nachstellen. Genau hier ist aber gleichzeitig das präzise und kraftvolle Manövrieren des Antriebs zu spüren. In alle Richtungen wird der maximal mögliche Schub geleitet. Zum Bugsieren und Schleppen ist das genau der richtige Vortrieb.

In Abb. 7 und 8 ist gut zu erkennen, wie die beiden Zahnräder angeordnet sind. Die darüber gebaute „Brücke“ stellt noch eine weitere Lagerung der Antriebsachsen zur Verfügung. Erst damit ist der Antrieb stabil genug um sicher zu arbeiten.

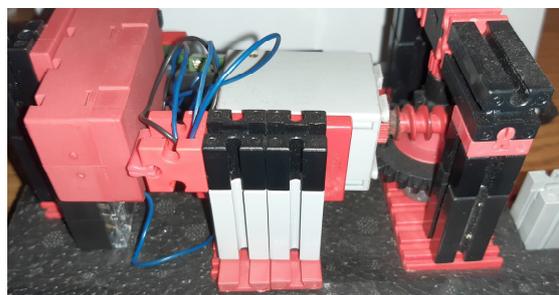


Abb. 8: Backbord-Seite

Die Ansicht von Steuerbord zeigt die Befestigung des großen Motors. Mit dieser Anordnung ist auch eine gute Gewichtsverteilung erreicht: möglichst tief und mittig. Das ist insbesondere bei kraftvollem Kurvenfahren wichtig.

Fahrverhalten

Der Katamaran reagiert sehr empfindlich. Bei einem echten, großen Schiff sind die Bordwände um vieles länger; das begünstigt den Geradeauslauf: Das seitliche Wasser wirkt wie eine Führung. Das fehlt hier bei dem kleinen Modell: Beim Fahren will das Heck immer wieder den Bug überholen. Die Ausrichtung des POD ist daher von entscheidender Bedeutung. Das Austarieren einer präzisen und zugleich kraftvollen Lenkung des Schiffes ist für ein kleines Modell richtig knifflig, zumal der Antrieb richtig „Wums“ hat. Nach einigen Versuchen habe ich mir einen Zeiger auf das Zahnrad Z30 für die Geradeaus-Ausrichtung des POD gemacht. Damit ist es viel leichter, die Richtung und Stabilität des Bootes zu halten.

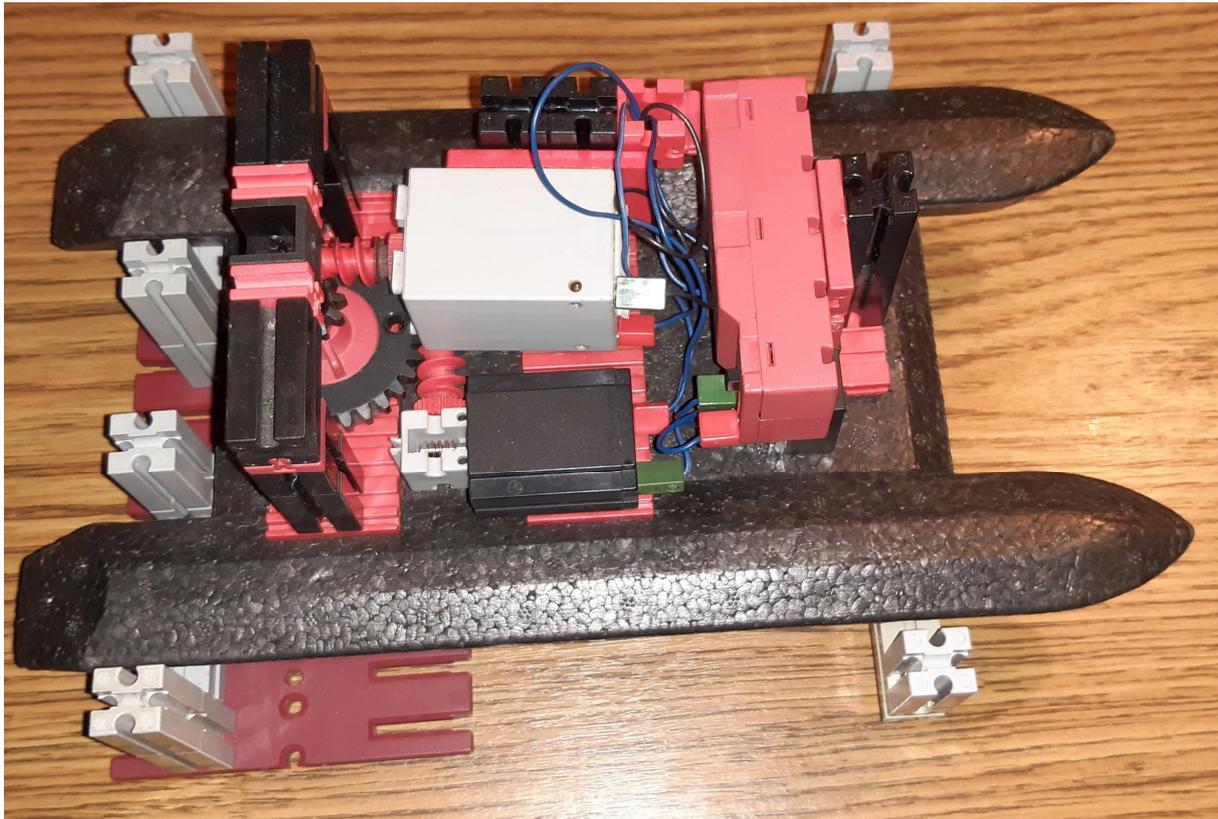


Abb. 9: Gesamtansicht

Anderer Aufbau

Als ich mich später noch einmal mit dem Modell beschäftigte, habe ich eine weitere Lösung gefunden: Der POD kann auch mit zwei senkrecht nach unten stehenden Achsen betrieben werden.

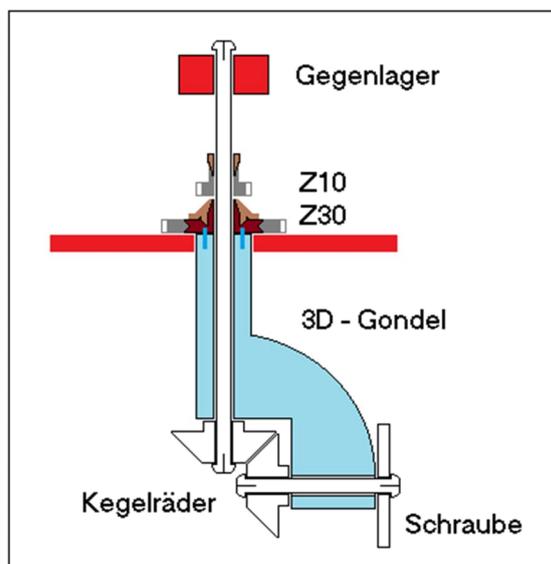


Abb. 10: Schema mit 3D-Plot

Die beiden zugehörigen schematischen Zeichnungen zeigen Abb. 10 und 11.

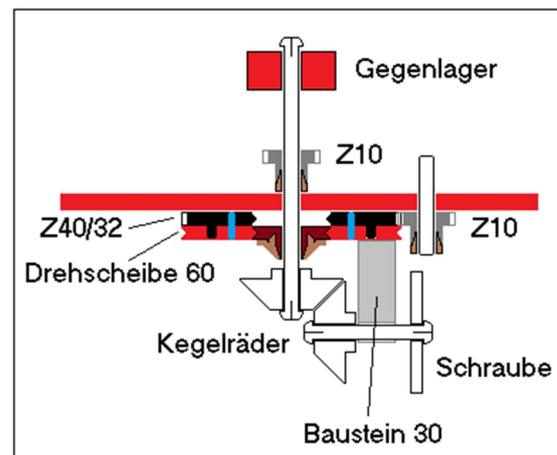


Abb. 11: Schema nur mit fischertechnik

Bleibt man bei der reinen Verwendung von fischertechnik-Bauteilen muss man die Drehbewegung der Gondel außerhalb der Drehachse nach unten führen; hier beispielhaft mit einem großen Z40/32 und einem Z10. Auf das Z40/32 klemmt man eine Drehscheibe 60 – die Grundlage, um

mit weiteren fischertechnik-Steinen weiter zu machen.

Im Grunde haben wir hier das gleiche Problem wie mit den Zeiger-Durchführungen bei Uhren. Wie bekommt man zwei unterschiedliche Drehbewegungen auf gleichem Achsmittelpunkt durch eine Lagerstelle? Die reine fischertechnik-Version hat leider den Nachteil, dass der oben benötigte Durchmesser sehr groß wird. Unterhalb der fischertechnik-Platte muss noch viel mehr

an Styropor entfernt werden. Die 3D-gedruckte Version ist an dieser Stelle eleganter. Wer sie nachbauen möchte, findet auf der [Webseite zu dieser ft:pedia-Ausgabe](#) die STL-Daten zum Download.

Referenzen

- [1] Friedrich W. Pleuger, Friedrich F. Busmann: [Device for Propelling a Ship](#). United States Patent Office, Pat. No. 2,714,866, 09.08.1955.

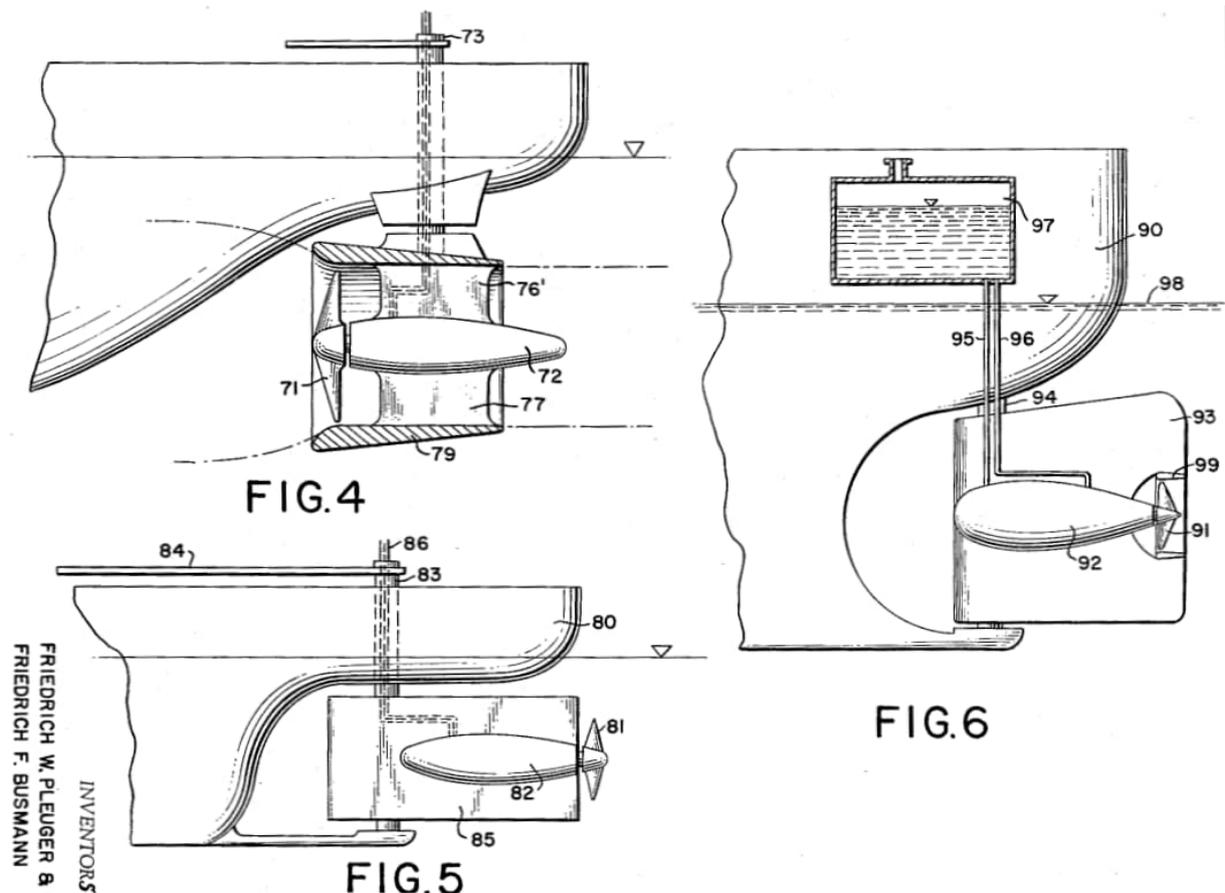


Abb. 12: Auszug aus der Patentschrift von Pleuger/Busmann (1955) [1]

Modell

Ein Bild von einem Modell (Teil 3): Betonmischer 1967

Stefan Falk

In den ersten DIN-A4-Bauanleitungen zu den fischertechnik-Grundkästen der 1960er Jahre finden sich neben Modellen für die Kästen fischertechnik 100, 200, 300 und 400 am Ende auch ein paar „Großmodelle“. Wohl kaum jemand hatte damals genug Bausteine, um sie nachzubauen, und da nur ein einziges Foto gezeigt wurde, kann man auch viele Details nicht sehen. Das rief nach Nachbauversuchen. In Teil 3 versuche ich mich am Betonmischer von 1967.

Worum geht's?

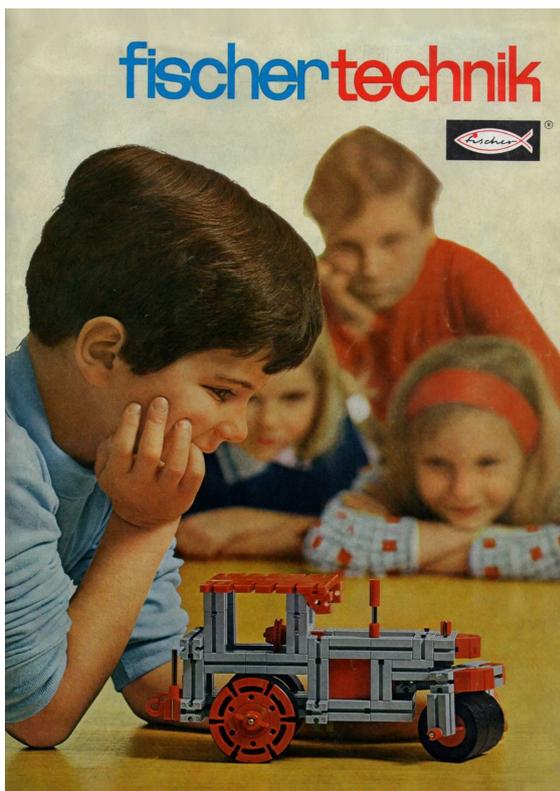


Abb. 1: Titelseite der Anleitung von 1967

Die Anleitung aus dem Jahr 1967 [1, 2, 3] enthält auf Seite 27 ein einziges Foto eines Betonmischers (Abb. 3). Auch in diesem Modell werden abgesehen von einem Haufen Grundbausteine (die fischertechnik-Statik gab es ja noch gar nicht) zwei Moto-

ren mot.1 verbaut. In meinem Nachbau sind es 60 Bausteine 30, 56 Bausteine 15, 30 Bausteine 15 mit zwei Zapfen und 52 Winkelsteine 30° gleichschenkelig – alles weit jenseits meines damaligen Teilebestands.

Der Nachbau

Wieder liegt der Reiz des Nachbaus in dem, was das Foto nicht zeigt: Das angetriebene Fahrwerk kann man ganz gut errahnen, aber wie ist z. B. die Mischtrommel gelagert? Auf welche Probleme stößt man noch beim Nachbau-Versuch?

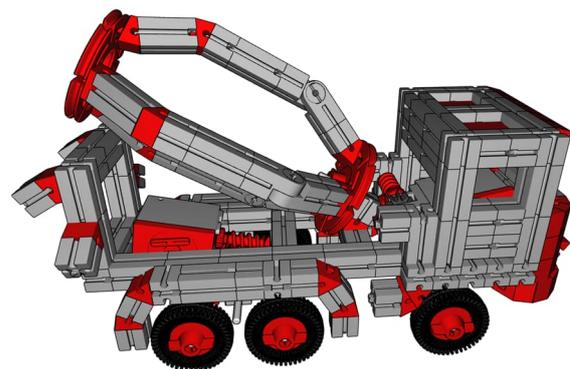


Abb. 2: Das fertige Modell im fischertechnik-Designer

Der Fahrtrieb

Die Konstruktion des Antriebs (Abb. 4) ist recht simpel und folgt etwas, was man schon fast als ein ungeschriebenes fischer-

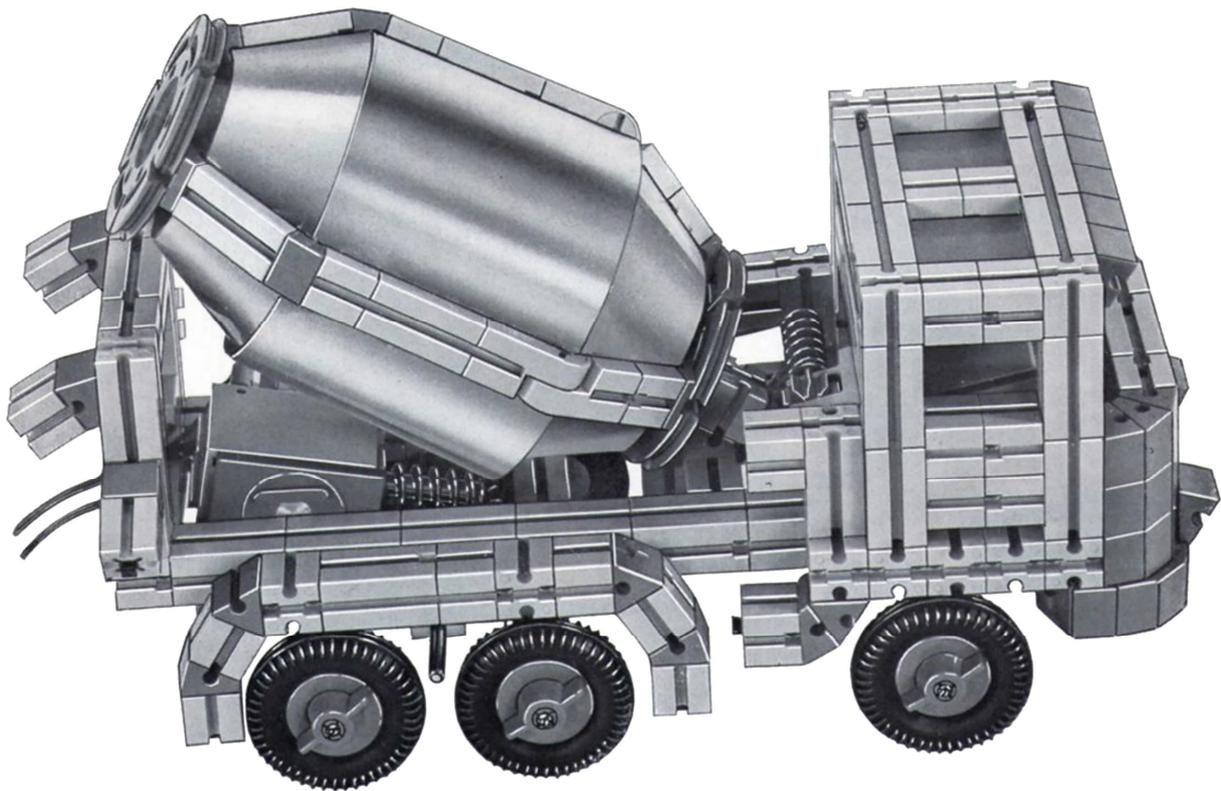


Abb. 3: Dieses eine einzige Foto aus der Anleitung steht als Vorlage zur Verfügung

technik-Gesetz der damaligen Zeit ansehen kann: Achsen von Fahrzeugen sitzen in einem senkrecht stehenden Baustein 30, der oben am Rahmen oder an einer Grundplatte 180×90 ([31001](#)) angebracht ist.

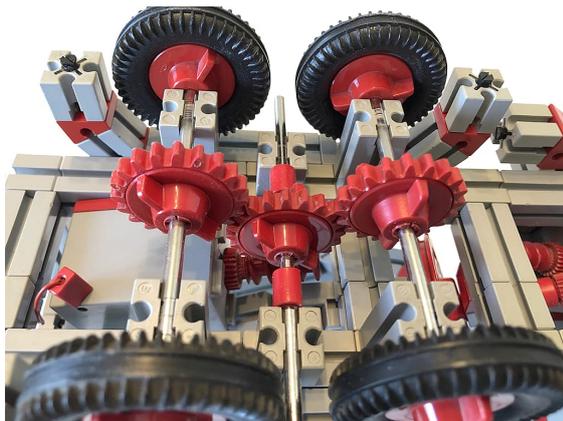


Abb. 4: Der Fahrtrieb von unten gesehen

Der Trommelantrieb

Zuerst nahm ich anhand des Fotos an, dass die Aufsteck-Schnecke des Motors ein Z15 antreibt – schließlich gilt es, die recht schwere Trommel herumzuwuchten.

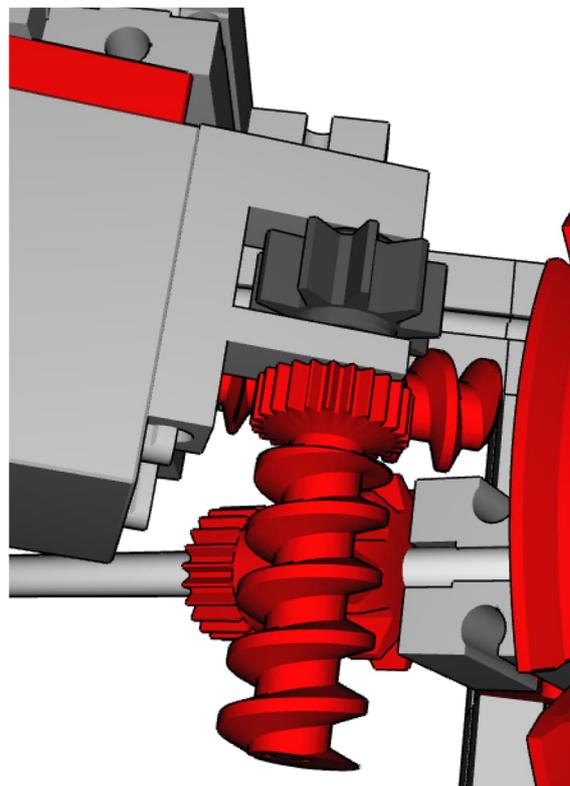


Abb. 5: Es kann nur ein Z10 gewesen sein

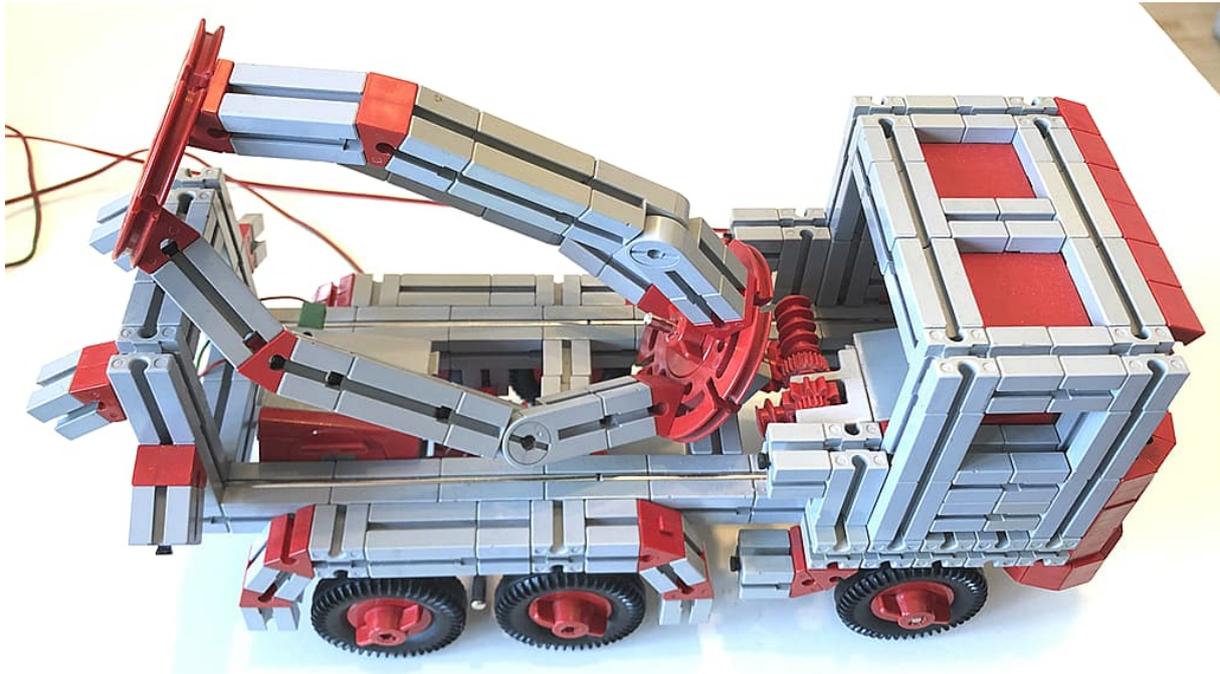


Abb. 6: Das nachgebaute Modell

Zwei Dinge sprechen aber dagegen: Erstens gab es das Z15 laut ft-Datenbank erst ab 1971, und zweitens passt die Aufsteck-Schnecke auch so schon kaum noch ins Modell (Abb. 6).

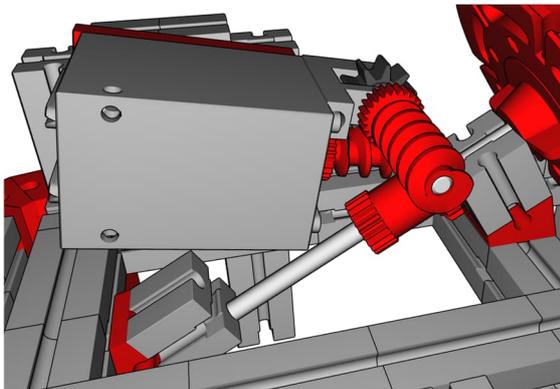


Abb. 7: Alles passt gerade so am Motor vorbei

Die Ur-Variante des Getriebehalters mit Schnecke ([31045](#)), die noch kein Ritzel mit feiner Verzahnung, sondern stattdessen eine längere Schnecke besaß, könnte besser passen – leider besitze ich von der keine und konnte es also nicht selbst nachprüfen.

Die Lagerung der Mischtrommel

Die recht schwere Trommel wird mit ihrer Drehachse wohl kaum nur von dem einzi-

gen von oben sichtbaren Lager-Baustein 15 gehalten werden. Die muss irgendwo da unten, wo's keiner sieht, nochmal gelagert sein. Ansonsten würde sich die Achse sofort heftig verkanten und der Motor hätte keine Chance, die Trommel sauber zu drehen.

Nach einiger Tüftelei fand ich eine Lösung, wie die Lagerung der Trommelachse zweifach ausgeführt gewesen sein könnte. Die muss außerdem zur „Stoßstange“ passen – dazu später mehr.

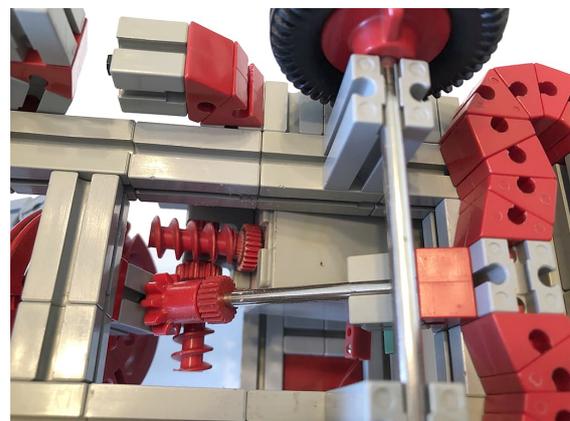


Abb. 8: Zweifach gelagerte Trommelachse

Wenn recht weit vorne im Rahmen eine Querverbindung aus Baustein 30 und Bau-

stein 15 mit zwei Zapfen existiert, kann unter dieser mittig ein Baustein 15 angebracht werden. Daran ein Winkelstein 30° und ein weiterer Baustein 15 ergeben ein gut justierbares Lager für die Achstrommel. Gleichzeitig wird damit klar, dass die Trommel auf einer Achse 110 gelagert sein muss. So gelagert, kann sich die Trommel leichtgängig drehen (vor allem, wenn man die Achse mit etwas Vaseline schmirt), und alles passt noch, ohne anzustoßen, am Motor vorbei.

Die Stoßstange

Ebenso ein Rätsel ist, wie die Konstrukteure damals die „Stoßstange“ gebaut und befestigt haben. Das wird erschwert durch die Tatsache, dass es 1967 zwar noch den Winkelstein 30° rechtwinklig gab ([31012](#), siehe z. B. links vom Vorderrad in Abb. 3), aber ansonsten nur die Ur-Version des Winkelsteins 30° – die etwas kleiner ist als die heutige Fassung [4]. Falls es jemand selbst versuchen möchte: Es gab damals auch noch keine Federnocken, keine Bausteine 15 und keine Bausteine 7,5 – nicht einmal einen Verbinder 15.

Das Folgende ist also pure Theorie, denn man sieht es auf dem Bild ja nicht – aber es funktioniert hervorragend und passt.

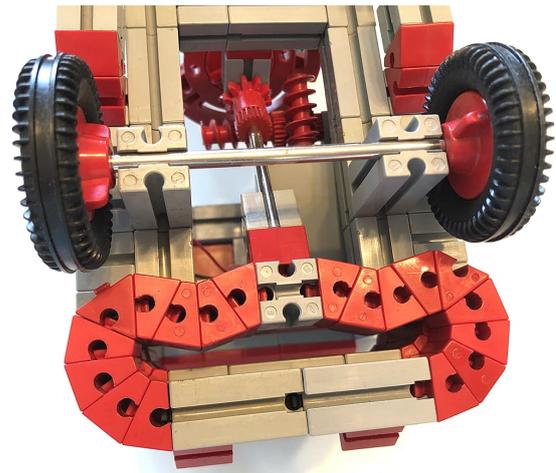


Abb. 9: Lagerung der Trommelachse und Stoßstange

Die Ur-Version der Winkelsteine 30° , von denen je drei links und rechts der Stoßstange sitzen, ergibt einen leicht „krummen“ Abstand der weiteren Bauteile. Der lässt sich ausgleichen, in dem auf beiden Seiten mit einer Mimik aus je vier weiteren Winkelsteinen 30° eine „Welle“ konstruiert wird, die insgesamt einen passenden Abstand ergibt.

Wie schon gesagt: So könnte es gewesen sein, aber vielleicht war es auch anders konstruiert. Wer Lust hat, möge gerne tüfteln und eine andere, vielleicht sogar einfachere Variante demonstrieren.

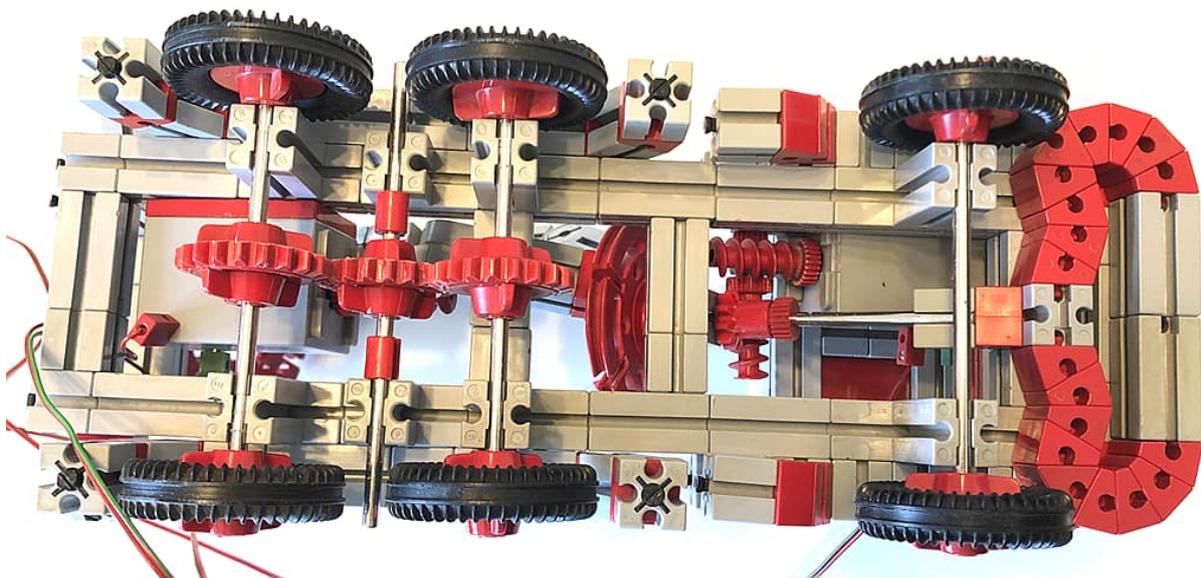


Abb. 10: Der Betonmischer von unten

Quellen

- [1] fischertechnik: Bauanleitungen der 1960er Jahre. In der [fischertechnik-Datenbank](#).
- [2] fischertechnik-Club NL: [Basis Dozen](#). Bauanleitungen der Standard-Baukästen der frühen fischertechnik-Jahre.
- [3] fischertechnik: Bauanleitung zu den Kästen fischertechnik 100, 200, 400 von 1967. Direkt-Link zum [PDF-Download](#) in der fischertechnik-Datenbank.
- [4] Stefan Falk: *Perlentauchen (Teil 1)*. [ft:pedia 3/2012](#), S. 11–18.
- [5] Stefan Falk: *Betonmischer 1967*. Im [Bilderpool der ftCommunity](#), 2023.
- [6] Stefan Falk: *Betonmischer 1967*. fischertechnik-Designer-Datei im [Download-Bereich der ft:pedia](#).
- [7] Stefan Falk: *Ein Bild von einem Modell (Teil 1): Großer Portalkran 1967*. [ft:pedia 1/2023](#), S. 13–19.
- [8] Stefan Falk: *Ein Bild von einem Modell (Teil 2): Großer Raupenkran 1969*. [ft:pedia 1/2023](#), S. 20–25.

Modell

Die erste Lokomotive

Rüdiger Riedel

Beim Stöbern im Internet traf ich auf ein Video, das den Nachbau der ersten Lokomotive im Fahrbetrieb zeigte.

Tatsächlich war es nicht George Stephenson, der die erste brauchbare Lokomotive gebaut hat, sondern 1804 Richard Trevithick. Seine Dampflokomotive war absolut funktionsfähig, nur waren die für Pferdebahnen ausgelegten gusseisernen Schienen für das Gewicht seiner Lokomotive ungeeignet und brachen (Abb. 1). Für den modellmäßigen Nachbau stellte sich die Frage nach dem extrem langen Kolbenhub. Mit einem einzelnen Pneumatik-Zylinder war das nicht machbar. Also baute ich drei Zylinder

hintereinander. Dieses Trio lässt sich erstaunlich gut führen, nur reicht der Luftstrom eines Kompressors dafür nicht aus. Also habe ich zwei Kompressoren parallelgeschaltet. Die Luftmenge pro Zeiteinheit wird verdoppelt, der maximale Druck bleibt natürlich unverändert. Die alten Festventile arbeiten wunderbar und zuverlässig. Man kann sie immer noch kaufen, zum Beispiel bei [2]. Allerdings sind die zugehörigen Rollenhebel knapp.

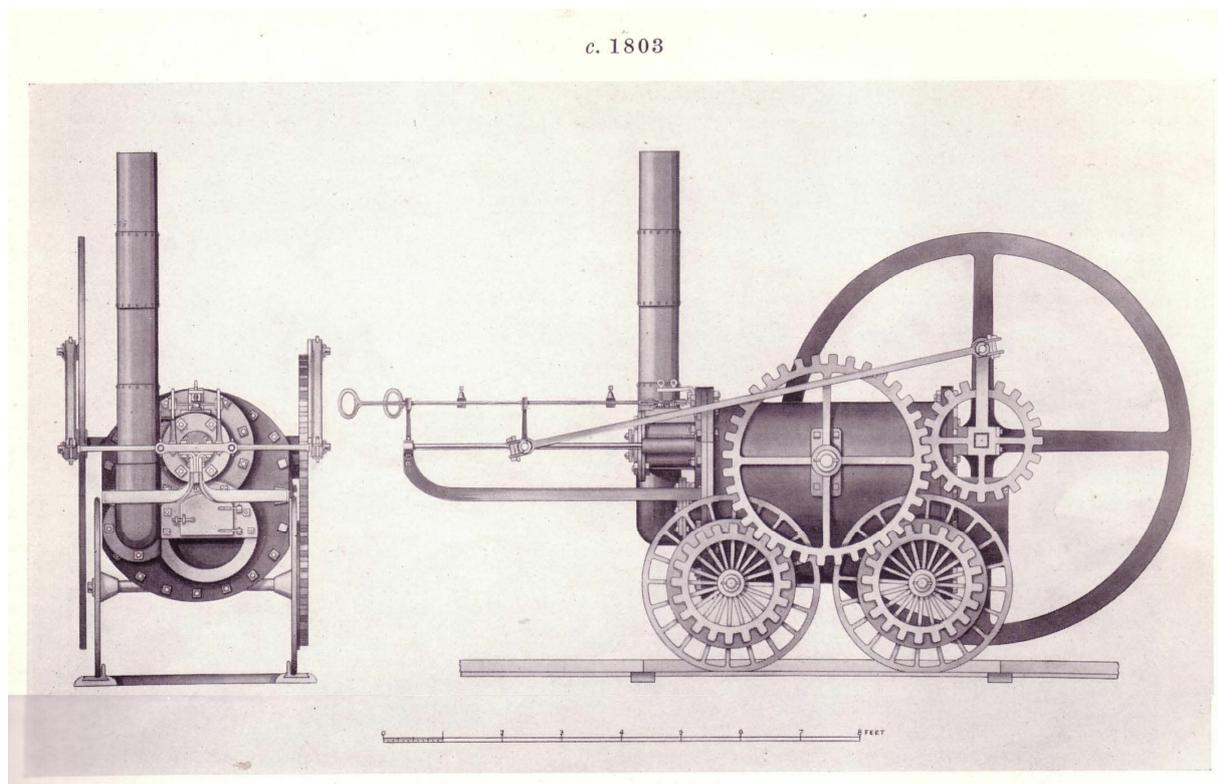


Abb. 1: Die erste Lokomotive, gebaut von Richard Trevithick

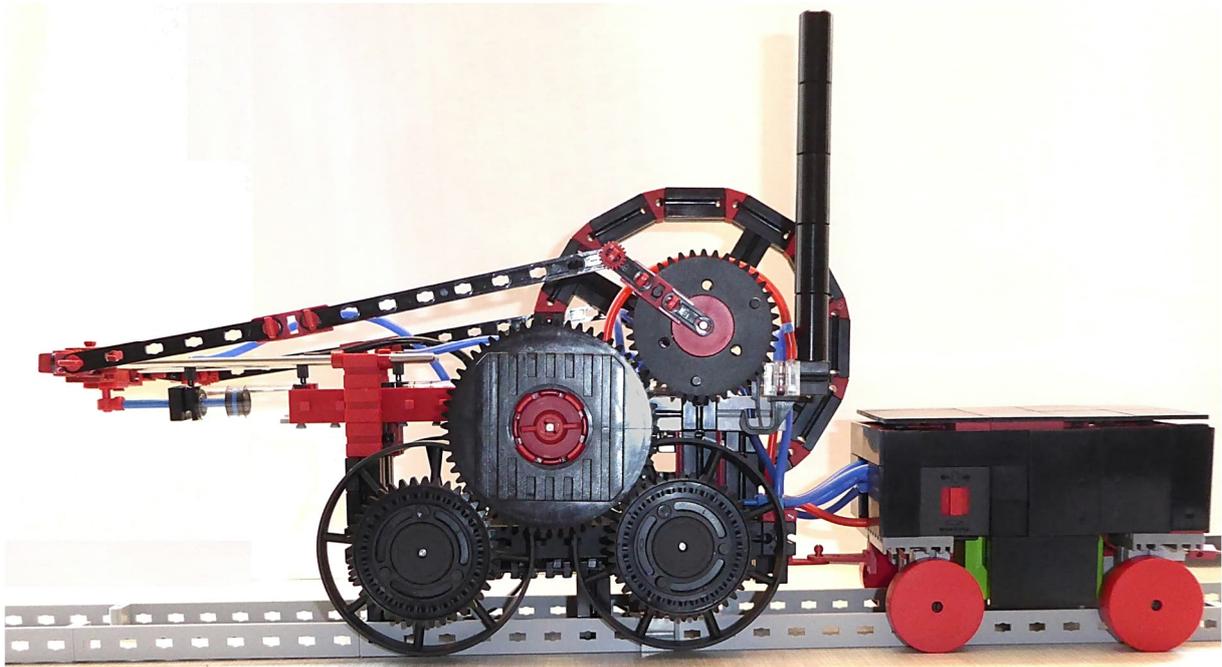


Abb. 2: Das Modell der ersten Lokomotive, hier mit Tender

Die Maschine fängt beim Einschalten sofort an zu arbeiten, wenn sie nicht in einem der beiden Totpunkte stehegeblieben ist.

Der Betrieb erfolgt aufgebockt; ein Fahrbetrieb ist nicht möglich, wie das Video zeigt [1].

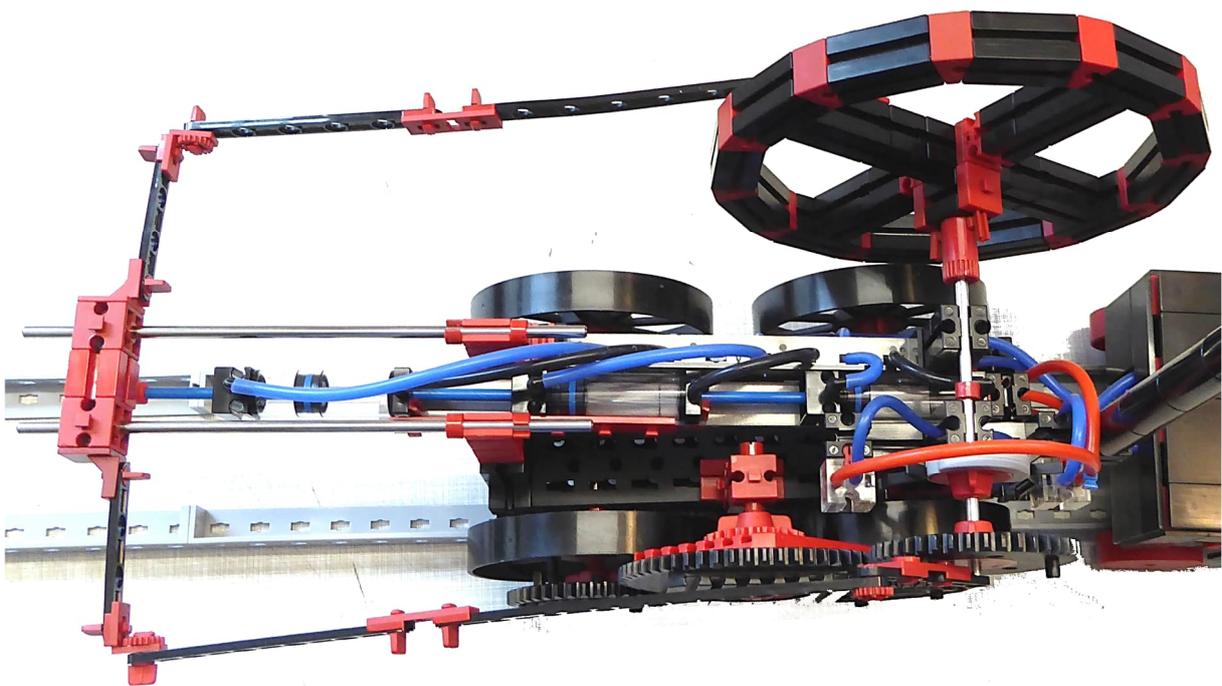


Abb. 3: Oberer Totpunkt



Abb. 4: Unterer Totpunkt

Zu Beginn wollte ich das noch nicht abschließen, deshalb ist im Tender nicht nur der Kompressorenzwilling untergebracht, sondern auch eine Akkubox (von Bello, wurde im Forum beschrieben). Die Akkubox ist grün und befindet sich unter den Kompressoren. Für den aufgeständerten Dauerbetrieb habe ich hinten einen Anschluss zur externen Stromversorgung vorgesehen.

Links im Tender sind zwei Handventile. Damit kann die „Fahrtrichtung“ umgesteu-

ert werden. Deswegen die vielen Silikon-schläuche zwischen Tender und Lokomotive. Die roten Schläuche dienen der Zuluft, alle anderen sind Blau.

Hinweis

Per Zufall habe ich zwei besonders stabile Festo-Ventile erwischt. Sie halten den maximalen Luftdruck von etwa 0,7 bar aus. Versuche mit weiteren Festo-Ventilen verliefen viel schlechter, sie leckten schon bei Drücken von 0,3 bis 0,5 bar. Der Hinweis von Stefan Roth [3] ist also sehr berechtigt.

Quellen

- [1] Rüdiger Riedel: *Rudis ft Projekt Lokomotive Trevithick*. Auf [YouTube](#), 2023.
- [2] Martin Bössner: Pneumatik-Teile auf [Fischertechnik-Teile.de](#).
- [3] Stefan Roth: Pneumatik-Teile auf [fischerfriendsman.de](#).

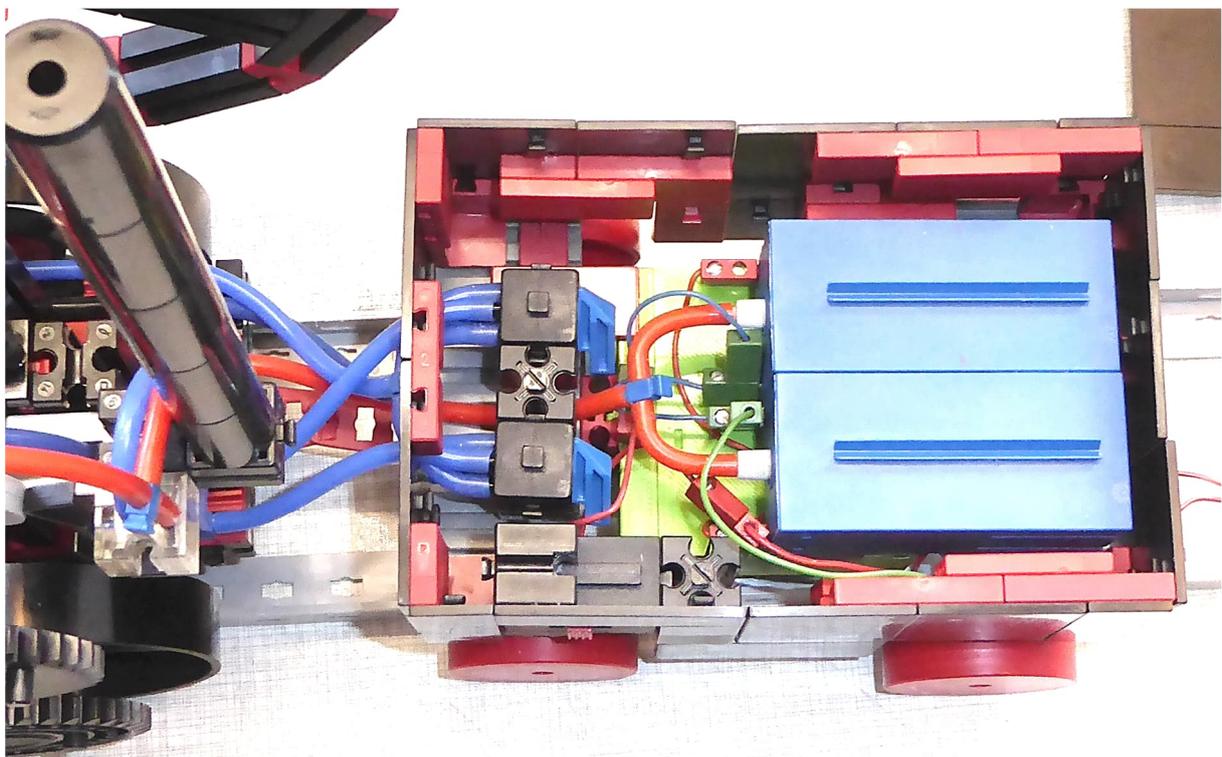


Abb. 5: Blick in den Tender

Tipps & Tricks

Die Odyssee mit den Steckern

Hans-Christian Funke

fischertechnik hat mit seiner neuesten Generation von Steckern die Fan-Welt massiv erschüttert. Obwohl diese Steckerform eine schnellere Montage von Kabeln gestattet, bringt sie auf der anderen Seite leider erhebliche Nachteile mit sich.

Einen – aus meiner Sicht erheblichen – Nachteil der neuen Stecker-Generation stellt die Tatsache dar, dass dem fischertechnik-Fan mit diesen Steckern ein Stück Flexibilität genommen wird. Aber gerade diese Flexibilität ist doch das, was das System von fischertechnik ausmacht. Mit den Steckern der vorherigen Generationen konnten auch Bauteile wie Widerstände, Kondensatoren, mehrere Kabel zusammen und andere Bauelemente montiert werden. Dies war durch eine Schraube im Stecker möglich, über die der Stecker der neuesten

Generation (Abb. 3 und 6) nicht mehr verfügt.

In ft:pedia 1/2020 hat Jan Hanson bereits recht ausführlich allerhand Alternativen für Stecker und Ersatzstecker vorgestellt [1]. Es gibt jedoch einen nicht unerheblichen Anteil an fischertechnik-Fans, die gerne Originalteile verwenden oder wenigstens Teile, die identisch von Form und Aufbau sind. Im Klartext: Ein Eisenbahnstecker oder ähnliches sind für sie in einem fischertechnik-Modell tabu. Einige fischertechnik-Freunde wollen auch gerne bei dem



Abb. 1: Stecker Typ 1
(Original [31337](#))



Abb. 2: Stecker Typ 2
(Original [31337](#))



Abb. 3: Stecker Typ 3
([181584](#))

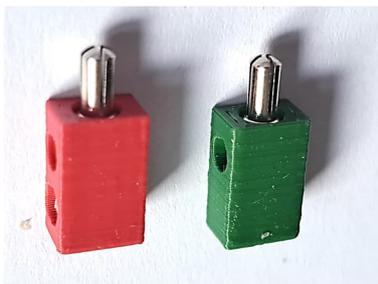


Abb. 4: Stecker Typ 1 (gedruckt,
[31336](#) und [31337](#))

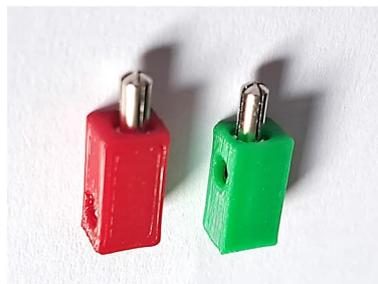


Abb. 5: Stecker Typ 2 (gedruckt,
[31336](#) neu und [31337](#) neu)



Abb. 6: Stecker Typ 3
([181583](#) und [181584](#))

Händler ihrer Wahl einfach nur die benötigten Teile bestellen, wozu natürlich auch die Stecker gehören. Und damit bin ich beim Thema dieses Artikels angekommen:

Welche fischertechnik-Stecker sind noch wo erhältlich?

Die Steckerarten lassen sich grob in drei Typen unterteilen, wobei es noch verschiedene Abarten oder Sonderausführungen gab. Im Wesentlichen gibt es die Stecker vom Typ 1 (Abb. 1 und 4), die eine der ersten von fischertechnik veröffentlichten Versionen darstellen und von der schmaleren Version Typ 2 (Abb. 2 und 5) abgelöst wurden. Die Stecker der Typen 1 und 2 verfügen beide über eine Schraube und können somit auch bedrahtete Bauelemente oder mehrere Kabel aufnehmen.

Bei meiner Recherche bin ich bei zwei Händlern für fischertechnik-Stecker fündig geworden, die diese auch nachhaltig liefern können. Die Stecker der Typen 1 und 2 gibt es allerdings nur noch als gedruckte Version, dafür ist aber die Nachhaltigkeit garantiert.

Bei Stefan Roth [2] kann man die Stecker aller Typen und auch Buchsen erhalten, sogar in mehreren Farben. Die gedruckten Stecker stammen aus eigener Fertigung und sind somit nachhaltig verfügbar. Sie sind dem Original in Material, Farbe und Aussehen sehr ähnlich. Ansonsten gibt es dort gebrauchte fischertechnik-Teile und somit auch immer wieder originale Stecker aller Generationen, aber eben nur in begrenzter Zahl. Das Muster in Abb. 4 entstammt aus seiner Produktion.

Den aktuellen von fischertechnik angebotenen Stecker vom Typ 3 erhält man bei Franz Santjohanser [3], dem von fischertechnik autorisierten Fachhändler. Inzwischen gibt es aber auch bei ihm Stecker vom Typ 2 als gedruckte Version. Diese werden von ihm in eigener Regie angefertigt, sind somit gleichfalls nachhaltig verfügbar und sogar günstiger als die Stecker des Typs 3. Den Stecker vom Typ 2 gibt es bei ihm aktuell in den Farben rot und grün, aber demnächst sollen auch andere Farben erhältlich sein. Das Muster in Abb. 5 stammt aus seiner eigenen Herstellung. Die Stecker von Franz Santjohanser (Abb. 5) werden demnächst in seinem Onlineshop erhältlich sein. Einen genauen Termin konnte ich noch nicht in Erfahrung bringen.

Beide Quellen sind empfehlenswert und je nach Bedarf kann man sich hier zuverlässig und nachhaltig mit Steckern in fischertechnik-Bauform eindecken. Für mich persönlich ist es wichtig, derartige Bezugsquellen zu kennen, da ich für die Ausstattung einiger Elektronikmodule und Baukästen Stecker mit Schraube benötige.

Quellen

- [1] Jan Hanson: *Alternative Verbindungslösungen für Stecker und Buchsen*. [ft:pedia 1/2020](#), S. 21–29.
- [2] Stefan Roth: [fischerfriendsman](#) (die alte Domain [fischerfriendswoman](#) wird manchmal noch in Suchmaschinen angezeigt und funktioniert ebenfalls).
- [3] [santjohanser](#): Spielen. Lernen. Technik. Autorisierter fischertechnik-Fachhändler

Tipps & Tricks

Rutschen und Rappeln

Rüdiger Riedel

Rutschen und Rappeln können ganz schön nerven. Hier kommt Abhilfe.

Der fischertechnik-Power-Controller ist sehr leicht ([139778](#)). Schließe ich das Schaltnetzteil ([505287](#)) an, wird er durch das Gewicht des Dämpfungszyinders des Netzteils von meinem Tisch heruntergezogen. Anti-Rutsch-Füße müssen her.



Abb. 1: Der fischertechnik-Power-Controller von unten

Vier kurze Stücke Silikonschlauch in die Nuten gedrückt und der Power-Controller rutscht nicht mehr.



Abb. 2: Seitenansicht

Die Schlauchstücke sitzen gut in den Nuten und schauen gerade weit genug heraus.



Abb. 3: Kompressor mit Dämpferfüßen

Einer meiner Kompressoren ([121470](#)) macht erhebliche Geräusche, wenn ich ihn auf den Tisch stelle. Die Silikonfüße dämpfen den Lärm auf ein erträgliches Maß.

Tipps & Tricks

Rotationsencoder (Rotary Encoder) KY-040

Axel Chobe

Mit einem Drehgeber (Rotationsencoder) kann das Drehen einer Welle (Richtung, Weite) bestimmt werden. Er kann z. B. als „digitales Potentiometer“ eingesetzt werden. Den KY-040 gibt es günstig als kleine Platine, die auch in ROBO Pro am TXT genutzt werden kann.

Hintergrund

Der Drehgeber (oder Rotationsencoder) KY-040 ist ein Dreheingabegerät (wie ein Drehknopf), das anzeigt, wie weit *und* in welche Richtung der Drehknopf gedreht wurde. Es eignet sich für die Steuerung von Schritt- und Servomotoren oder auch von Geräten wie digitalen Potentiometern.

Außerdem ist ein Drucktastenschalter integriert: Durch Druck auf die Welle wird der Schalter geschlossen.

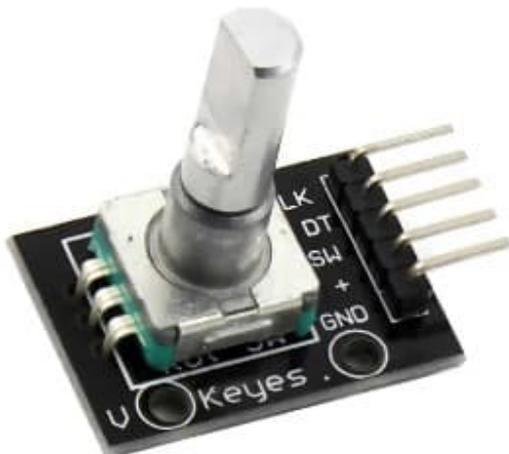


Abb. 1: Platine mit KY-040

Nachdem ich den Drehgeber erfolgreich bei der Programmierung vom micro:bit eingesetzt habe [1] und es ebenfalls Anleitungen für den Arduino [2] gibt, habe ich versucht auch für ROBO Pro eine Lösung zu finden.

Grundlagen des Drehgebers

Ein Drehgeber hat eine bestimmte Anzahl von Positionen pro Umdrehung. Diese Positionen sind leicht als kleine „Klicks“ zu spüren, wenn man am Drehgeber dreht. Im Inneren des Drehgebers befinden sich zwei Schalter. Ein Schalter verbindet Kontakt A mit Kontakt C und der andere Schalter verbindet Kontakt B mit C (Abb. 2).

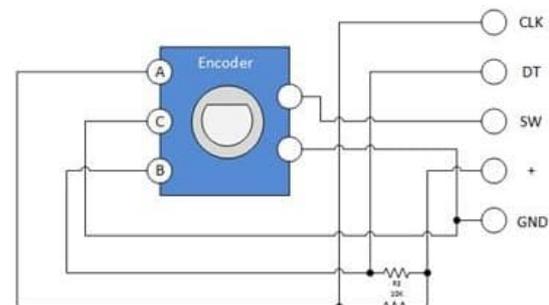


Abb. 2: Anschlüsse [2]

Die Winkelstellung der Kontakte ist so angeordnet, dass sich bei Drehung im Uhrzeigersinn der Zustand des Schalters zuerst in der Verbindung A und C ändert. Bei Drehung gegen den Uhrzeigersinn wird zuerst der Zustand der Verbindung B und C geändert (Abb. 3).

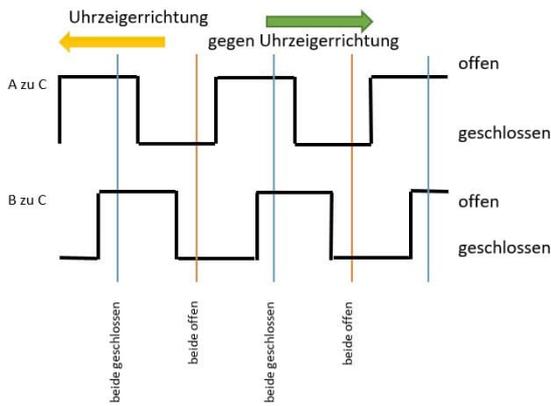


Abb. 3: Zustandsänderungen bei Drehung des Drehgebers mit bzw. gegen den Uhrzeigersinn

Beschaltung

Der eigentliche Drehgeber ist meist bereits auf eine Platine gelötet (Abb. 4).

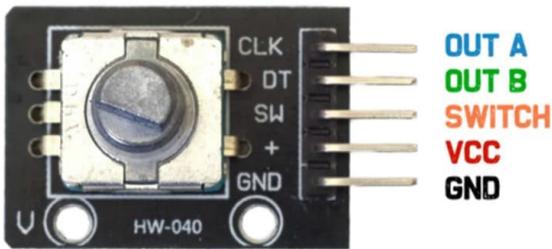


Abb. 4: Anschlüsse der Platine

Hier gibt es dann noch einen VCC-Anschluss, um über zwei Pullup-Widerstände definierte High- und Low-Werte zu bekommen. Für meine Arbeit habe ich die drei Anschlüsse A, B und C aber direkt eingesetzt.

Versuchsaufbau



Abb. 5: Anschluss des KY-040 an den TXT

Im Aufbau sind zwei Taster (I1 und I2), zwei Lampen (O1 und O2) und der Drehgeber (C1 und C2) sowie der Anschluss des Drucktasters (an C4) zu sehen. Den GND-

Anschluss verbinden wir mit einer GND-Buchse des TXT.

Umsetzung

Da der Abstand der Schalteränderungen für die Auswertung sehr kurz ist, habe ich die schnellen Zählgänge des TXT benutzt. Die Ausgangssituation ist unbekannt, d. h. die Kontakte können offen oder geschlossen sein. Deshalb muss bei „Warten auf“ die in Abb. 6 gezeigte Einstellung (*change*) vorgenommen werden.

0 -> 1 oder 1 -> 0

Abb. 6: Reaktion auf jeden Eingangswechsel

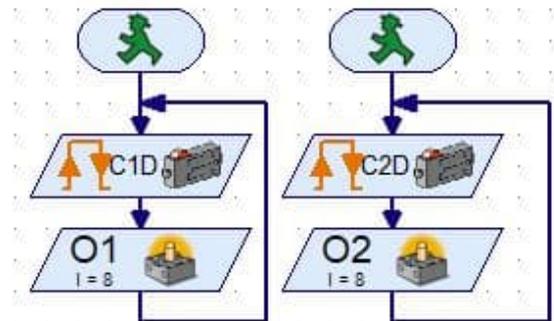


Abb. 7: Auswertung der beiden Schalter des Rotationsencoders

Um beide Tasten gleichzeitig auszuwerten werden hier zwei Prozesse gestartet. Dreht man den Regler nun ganz langsam, dann sollte eine der Lampen kurz vor der anderen aufleuchten.

Auswertung

Um nun die Drehrichtung bestimmen zu können, ändern wir das Programm leicht ab und zählen die ansteigenden Flanken (Wechsel von 0 auf 1).

Je nach Drehrichtung ändern sich die Variablen kurz nacheinander um 1 (Abb. 8).

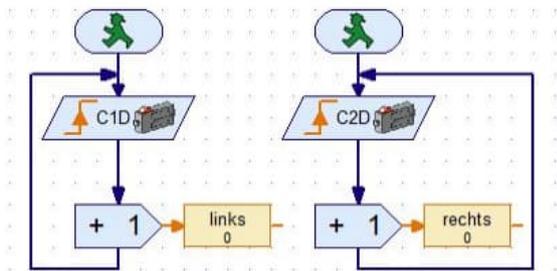


Abb. 8: Zählen der Impulse

Mit dem Programmelement „Vergleich“ lässt sich nun die Drehrichtung auswerten (Abb. 9). Jede Lampe signalisiert eine Drehrichtung; die entsprechende Lampe leuchtet eine Sekunde auf und anschließend werden die Variablen zurückgesetzt.

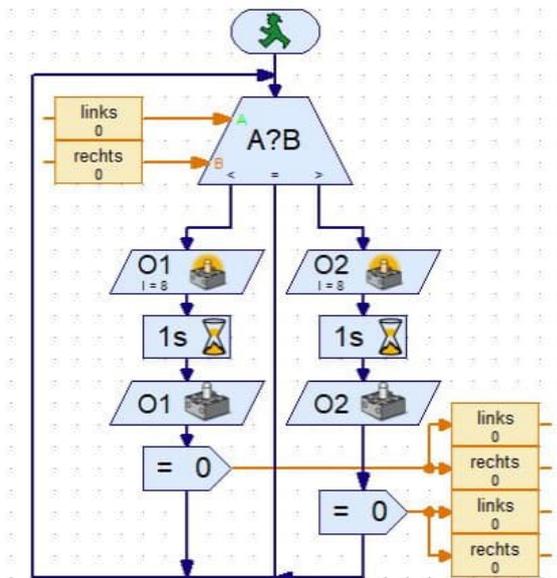


Abb. 9: Auswertung der Drehrichtung

Einschränkung: Bedingt durch die Pause wird nicht jeder Kontaktwechsel ausgewertet!

Anwendungsbeispiel



Abb. 10: Lautstärkeregelung im Fahrzeug

Als Beispiel habe ich die Lautstärkeregelung in modernen Autos ausgewählt. Sie kann entweder über die Taster am Lenkrad (Abb. 10, links) oder über das „Poti“ an der Konsole (Abb. 10, rechts).

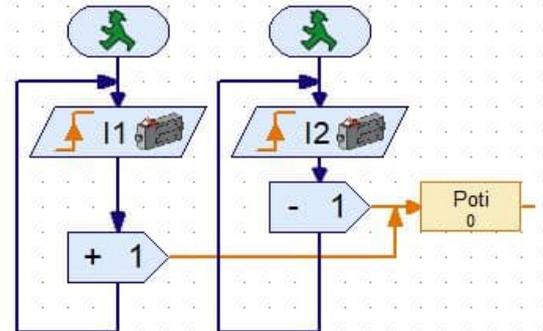


Abb. 11: Lautstärkeregelung über die beiden Taster an I1 und I2

Abb. 11 zeigt die Auswertung über die beiden Taster.

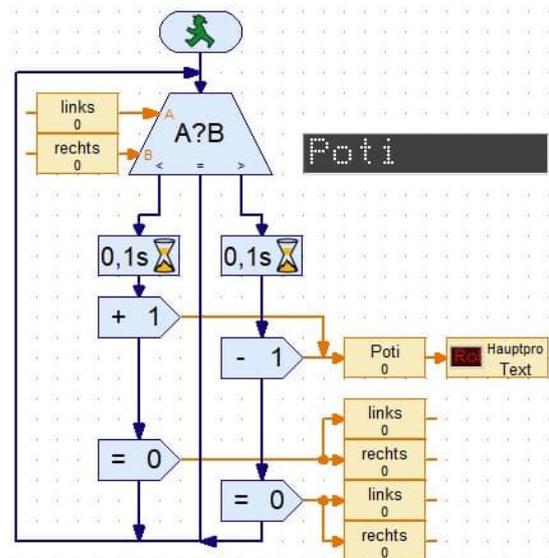


Abb. 12: Auswertung über den Drehgeber

In dem Programmteil in Abb. 12 wird die Richtung ausgewertet und ebenfalls die Variable „Poti“ modifiziert und angezeigt.

Quellen

- [1] Axel Chobe: [Rotary Encoder KY-040 am micro:bit](#), 2020.
- [2] [KY-040 Arduino Rotary Encoder](#). User Manual, 2018.

Tipps & Tricks

Quadratur-Encoder für fischertechnik-Encoder-Motoren

Florian Bauer

fischertechnik-Encoder-Motoren sind bewährte und sehr praktische Aktoren für computerisierte Modelle. Standardmäßig haben sie nur einen Impulsausgang. In diesem Beitrag wird eine Modifikation vorgestellt, mit der man die Motoren mit einem Quadratur-Encoder-Ausgang ausstatten kann, sodass sie auch für einen Servo-Betrieb tauglich werden.

fischertechnik-Encoder-Motoren

fischertechnik-Encoder-Motoren (186175, [153422](#) und [135484](#)) sind gut geeignet, um Modelle mittels Computer anzusteuern. Serielle Roboter, Fabrik-Modelle oder Fahrzeuge können mit diesen Motoren leicht aufgebaut werden. Im Steuerprogramm ist die Drehrichtung bekannt. Man zählt die Impulse und stoppt den Motor, sobald die Zielposition erreicht ist, oder etwas vorher, um dem Nachlauf Rechnung zu tragen.

Jeder Encoder-Motor hat auf der Motor-Rückseite ein magnetisches Encoder-Rad mit mehreren Magneten, welches sich über einen digitalen Magnet-Sensor (Hall-Sensor) auf einer kleinen Platine auf der Rückseite des Motors dreht. Sobald sich ein Magnet über dem Hall-Sensor befindet, gibt der Sensor ein H-Signal aus. Die Impulse dieses Signals können gezählt und damit für die Positionsbestimmung verwendet werden. Die Auflösung der Motoren beträgt laut Datenblatt 63,33 Impulse pro Umdrehung der Getriebe-Antriebsachse.

Will man einen Encoder-Motor als Servo betreiben, muss man ihn in einen Regelkreis einbauen und die Winkelposition der Motorachse kontinuierlich messen.



Abb. 1: Geöffneter fischertechnik-Encodermotor „Competition“ (186175)

Während der Regelung kann es passieren, dass der Regler ein Steuersignal in die Gegenrichtung ausgibt, obwohl der Läufer des Motors wegen seiner Trägheit noch in die ursprüngliche Richtung läuft. Die Zählung der Impulse kommt in diesem Fall zu einem falschen Ergebnis und die Fehlzählungen akkumulieren sich, sodass sich die Ist-Position permanent verschiebt.

Um diesen Effekt zu vermeiden braucht man eine saubere Impulzzählung, die die tatsächliche Drehrichtung des Motors

erkennen kann. Man erreicht dies mit einem Quadratur-Signal, das aus zwei um 90° phasenverschobenen Einzelsignalen besteht. Aus der Reihenfolge der Signalwechsel kann die Drehrichtung des Motors bestimmt werden. Ein weiterer Vorteil eines Quadratur-Signals ist die vierfache Auflösung bei Auswertung aller vier Flanken.

fischertechnik hat auch 24V-Encoder-Motoren mit einem solchen Quadratur-Ausgang im Angebot ([144643](#)) [6]. Die Platine der 9V-Encoder-Motoren ist mit lediglich einem Hall-Sensor bestückt. Die Platine hat aber einen nicht bestückten Pfostenfeld-Pin sowie unbestückte Landing-Pads für die Aufnahme von weiteren Bauelementen. Leider ist mir nicht bekannt, für welche Bauelemente sie gedacht sind. Auf diese Landing-Pads, die bereits sinnhaft für weitere Encoder positioniert sind, kann man aber auch selbst einen weiteren Hall-Sensor auflöten, um einen Motor mit Quadratur-Ausgang zu erhalten. Als Hall-Sensor habe ich einen unipolaren Hall-Schalter von Infineon vom Typ TLE4905L verwendet [5].

Im Folgenden werden die notwendigen Schritte für die Modifikation der neueren Encoder-Motoren ([186175](#), [153422](#)) beschrieben. Man sollte sie nur dann vornehmen, wenn man Erfahrungen mit feinen Lötarbeiten hat. Ich weise darauf hin, dass mit dieser Modifikation jeglicher Garantieanspruch erlischt. Außerdem übernehme ich keinerlei Verantwortung für etwaige Schäden. Alle Angaben sind ohne Gewähr; wer sich dennoch daran wagt, tut dies auf eigenes Risiko.

Die Modifikation beschränkt in keiner Weise den 1-Kanal-Betrieb, da der zweite Kanal nicht zwingend genutzt werden muss. Die Modifikation erweitert lediglich die möglichen Anwendungsfälle. Darüber hinaus ist der „Mod“ reversibel: Wenn man aus irgendeinem Grund zum ursprünglichen Zustand zurückkehren will, entlötet man einfach alle Bauelemente wieder.

Es sei hier noch einmal darauf hingewiesen, dass die hier beschriebene Modifikation nur für die neueren Encoder-Motoren gilt, die eine kleine Ausbuchtung an der Rückseite haben. Die Platinen der alten Motoren sind etwas anders aufgebaut. Ein entsprechender „Mod“ sollte möglich sein, gestaltet sich aber etwas schwieriger, da diese Motoren nicht für einen weiteren Ausgang vorbereitet sind.

Beschreibung der Modifikation

Die Modifikation beinhaltet folgende Änderungen:

- Einlöten eines zweiten digitalen Hall-Sensors auf Landing-Pads auf der rechten Seite der Motor-Platine
- Einlöten eines Pfostenfeld-Pins für das zweite Sensor-Signal
- Anlöten von Drähten für die Stromversorgung für den Sensor
- Löten einer Brücke auf der rechten Seite, um das Hall-Sensor-Signal auf den Pfostenfeld-Pin zu leiten

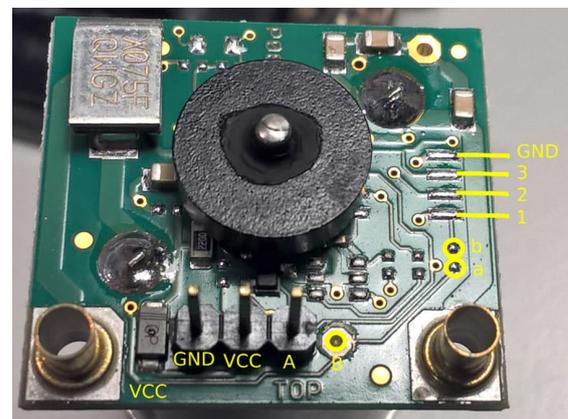


Abb. 2: Platine des Encoder-Motors mit Kennzeichnung der für die Modifikation relevanten Stellen.

Schritt 1

Zuerst muss das Gehäuse vorsichtig geöffnet und der Motor komplett herausgenommen werden. Abb. 2 zeigt die Platine des Motors mit Beschriftungen für wichtige

Stellen, die für die Modifikation relevant sind.

Der Sensor wird auf die Landing Pads 1, 2 und 3 aufgelötet. Die Lötunkte a und b müssen verbunden werden, VCC wird von der Anode der Diode unten abgezweigt und in das Lötauge B muss ein Pfostenfeld-Pin gelötet werden.

Schritt 2

Als nächstes muss der digitale Hall-Sensor vorbereitet werden. Als Sensor wird ein TLE4905L von Infineon verwendet, dessen Pinout in Abb. 3 zu sehen ist. Dieser Sensor hat einen sehr großen Versorgungsspannungsbereich bis 32V.

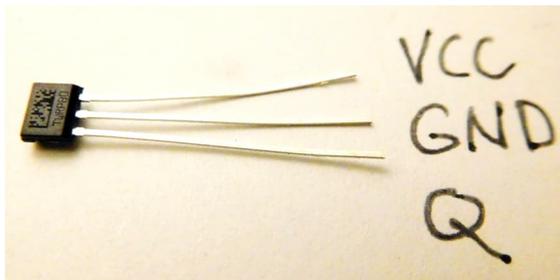


Abb. 3: Pinbelegung des Hall-Schalters TLE4905L

Der Sensor muss auf 8mm gekürzt werden (Abb. 4). Damit soll vermieden werden, dass die Anschlussleitungen in den äußeren Rand der Platine reichen, wo sie mit dem Snap-In-Verschluss der Rückplatte kollidieren würden.

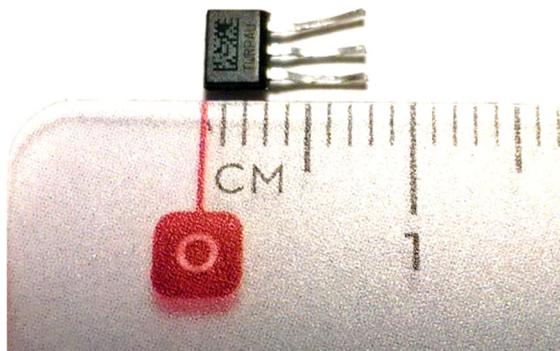


Abb. 4: Gekürzter Hall-Sensor, vorbereitet für die „Einlötlung“ auf die Landing Pads

Schritt 3

Der Pfostenfeld-Stift für den zweiten Ausgang muss in das Lötauge B eingelötet werden. Er muss ggf. auf der Platinen-Unterseite mit einem Seitenschneider gekürzt werden, damit er nicht das Motorgehäuse aus Metall berührt. Des Weiteren wird eine Brücke zwischen die Punkte a und b gelötet, die das Ausgangs-Signal des Sensors Q auf den Pfostenfeld-Stift B führt (Abb. 5)



Abb. 5: Platine mit eingelötetem Pin für den zusätzlichen Ausgang und Brücke, die das Signal des Sensors herausführt

Schritt 4

Der Hall-Sensor wird unter der Encoder-Scheibe platziert und auf die Landing Pads 1,2 und 3 gelötet (Abb. 6). Dabei wird der Ausgang Q auf Landing Pad 1, der GND-Anschluss des Sensors auf Landing Pad 2 und der VCC-Anschluss auf Landing Pad 3 gelötet. VCC (rote Leitung) und GND (schwarze Leitung) müssen an die entsprechenden Anschlüsse des Sensors gelötet werden.

Dabei ist darauf zu achten, dass der Hall-Sensor nicht am Encoder-Rad schleift. Ggf. muss die Motor-Platine etwas „tiefergelegt“ werden. Dazu lötet man die dicken Anschluss-Pads für die Versorgungsspannung des Motors auf und drückt dabei die Platine leicht herunter. Dabei muss man aufpassen, dass die Platine nicht auf dem metallenen Motorgehäuse aufliegt, damit es

nicht zu Kurzschlüssen kommen kann. Hier ist ein dünner Kartonstreifen als Abstandshalter zwischen Platine und Motorgehäuse hilfreich.



Abb. 6: Fertig installierter Hall-Sensor mit Leitungen für die Versorgungsspannung: rot: VCC, schwarz: GND.

Schritt 5

Nach dem Zusammenbau hat man einen Motor mit Quadratur-Ausgang (Abb. 7)



Abb. 7: Fertig zusammengebauter Motor mit Quadraturausgang

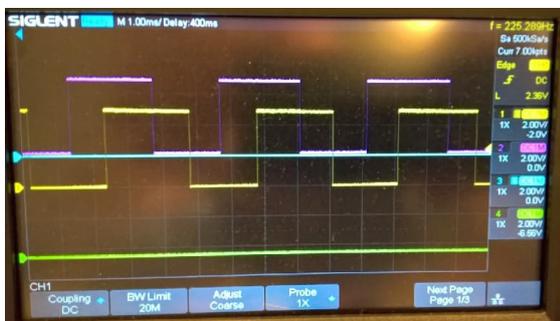


Abb. 8: Oszilloskop-Messung des Quadratur-Signals eines laufenden Motors (gelb: Original-Signal A, violett: Neues Signal B).

Schritt 6

Der Nachweis des Quadratursignals und der Phasenlage kann mit einem Oszilloskop erfolgen (Abb. 8). Um ein Signal zu messen, sind die Ausgänge mittels Pull-Up-Widerständen auf VCC (+5V) zu legen.

Aber die Validierung ist eigentlich nicht notwendig, denn die Position der Landing Pads ist so, dass die Phasenlage des zweiten Sensors fast automatisch auf 90° kommt.

Servo mit modifiziertem Motor

Mit einem derart modifizierten Motor kann man nun einen Servo aufbauen, was ich im Folgenden kurz beschreibe. Der Aufbau ist eine Servo-Steuerung mit einem Arduino-Mega und einem Motor Shield. Der Verdrahtungsplan ist in Abb. 9 dargestellt. Die Position des Motors kann mit einem Potentiometer eingestellt werden. Die Encoder-Ausgänge werden auf die Interrupt-fähigen Pins D18 und D19 des Arduino-Mega geführt. Im Programm müssen für diese Leitungen Pull-Up-Widerstände konfiguriert werden.

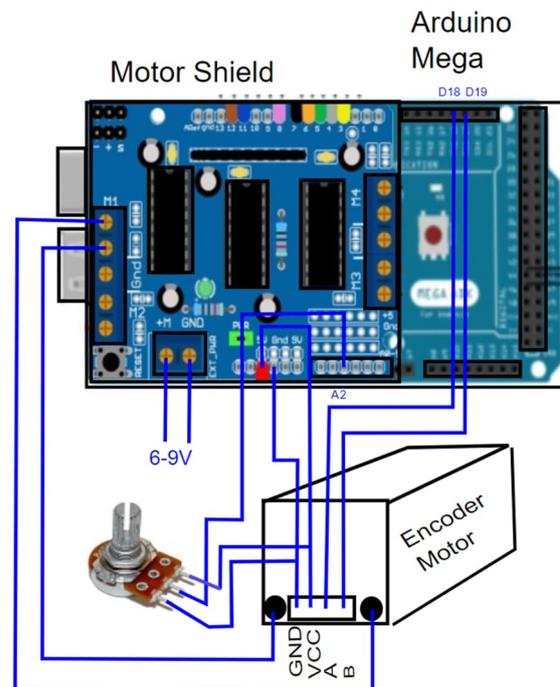


Abb. 9: Verdrahtungsplan für eine Servo-Steuerung aus Arduino-Mega mit Motor-Shield

```
#include <PID_v1.h>
#include <AFMotor.h>

AF_DCMotor mot1(1, MOTOR12_2KHZ);

int encoderPin1 = 18; // Encoder Output 'A' an Interrupt-Pin 18 des Arduino-Mega
int encoderPin2 = 19; // Encoder Output 'B' an Interrupt-Pin 19 des Arduino-Mega

volatile int lastEncoded = 0; // letzter Encoder-Wert
volatile long encoderValue = 0; // Raw-Encoder-Wert

double kp = 250, ki = .01, kd = 0.3; // PID-Parameter
// Ein und Ausgangsgrößen für den PID-Controller:
double input = 0; // Ist-Wert
double output = 0; // Stell-Wert
double setpoint = 0; // Soll-Wert
int limit = 222; // Begrenzung des Stell-Wertes

// PID-Controller-Objekt, das auf den Ein- und Ausgangsgrößen arbeitet:
PID myPID = PID(& input, & output, & setpoint, kp, ki, kd, DIRECT);

// PID-Controller for Servo
void setup() {
  Serial.begin(250000); // initialize serial communication

  pinMode(encoderPin1, INPUT_PULLUP); // Konfig. mit internem Pullup-Widerstand
  digitalWrite(encoderPin1, HIGH); // Pullup-Widerstand einschalten
  // Pin-Change-Interrupt konfigurieren
  attachInterrupt(digitalPinToInterrupt(encoderPin1), updateEncoder, CHANGE);

  pinMode(encoderPin2, INPUT_PULLUP);
  digitalWrite(encoderPin2, HIGH); // turn pullup resistor on
  attachInterrupt(digitalPinToInterrupt(encoderPin2), updateEncoder, CHANGE);

  myPID.SetMode(AUTOMATIC); // set PID in Auto mode
  myPID.SetSampleTime(1); // refresh rate des PID controllers
  myPID.SetOutputLimits(-limit, limit); // maximaler PWM Wert für Motor-Ansteuerung
}

void loop() {
  int User_Input = analogRead(2); // liest das Potentiometer aus (Wert 0-1023)
  setpoint = map(User_Input, 0, 1023, -250, 250);

  input = encoderValue; // data from encoder werden als Prozess-Wert betrachtet

  myPID.Compute(); // Stell-Wert neu berechnen
  pwmOut(0, output); // PWM-Stell-Wert an Motor ausgeben
}

void pwmOut(int i, int out) {
  if (out > 0) {
    mot1.run(FORWARD);
    mot1.setSpeed(out);
  } else {
    mot1.run(BACKWARD);
    mot1.setSpeed(abs(out));
  }
}

// Interrupt-Service-Routine, um den Quadratur-Encoder auszuwerten:
void updateEncoder() {
  int MSB = digitalRead(encoderPin1); // MSB = most significant bit
  int LSB = digitalRead(encoderPin2); // LSB = least significant bit

  int encoded = (MSB << 1) | LSB; // Konvertiere die 2-Pin-Werte in eine Zahl
```

```

int sum = (lastEncoded << 2) | encoded; // zu vorherigem Encoder-Wert addieren

if (sum == 0b1101 || sum == 0b0100 || sum == 0b0010 || sum == 0b1011)
    encoderValue++;
if (sum == 0b1110 || sum == 0b0111 || sum == 0b0001 || sum == 0b1000)
    encoderValue--;

lastEncoded = encoded; // alten Wert merken
}

```

Listing 1: Einfacher Sketch für Arduino Mega mit Motor Shield

Der obenstehende kommentierte Servo-Sketch verwendet die AFMotor-Bibliothek [1] und die PID_v1-Arduino-Bibliothek [2].

Die Encoder-Ausgänge werden an die Interrupt fähigen Pins 18 und 19 des Arduino-Mega angeschlossen.

Performance

Der schwarze Encoder-Motor hat eine Auflösung von ca. 253 Flanken pro Umdrehung der Getriebeachse. Die Stellzeit hängt von der Betriebsspannung und den Regelparametern ab. Bei 5V Betriebsspannung habe ich 1100 ms für eine Umdrehung erreicht, bei 9V 360 ms pro Umdrehung, was einer Stellrate von 1°/ms entspricht, wie die eines Graupner-Servos SJ (No. 7954).

Alle Werte wurden ohne Belastung gemessen. Da der Motor mit PWM angesteuert wird, könnte man die Betriebsspannung sicher auch etwas höher setzen und käme dann auf noch höhere Stellraten.

Der rote Encoder-Motor ([153422](#)) hat auch eine Auflösung von ca. 253 Flanken pro Umdrehung der Getriebeachse, ist aber langsamer und braucht bei 9V für 360° 550 ms.

Zusammenfassung

In diesem Beitrag habe ich einen einfachen „Mod“ vorgestellt, um einen 9V-fischer-technik-Encoder-Motor mit einem Quadratur-Ausgang auszustatten. Die vorgestellte Modifikation erlaubt es, diese Motoren als Servos einzusetzen. Vorteil gegenüber einem Servo ist sicherlich der größere Arbeitsbereich.

Der vorgestellte Sketch ist nur ein Minimal-Programm, um das Prinzip der Ansteuerung zu verdeutlichen. Es sei angemerkt, dass es weitaus ausgereifere und robustere Arduino-Bibliotheken wie [3] oder [4] gibt. Den Sketch findet ihr zum Download auf der [Webseite zu dieser Ausgabe der ft:pedia](#).

Referenzen

- [1] Adafruit: [Adafruit Motor shield V1 firmware with basic Microstepping support](#), Github Repository
- [2] br3ttb: [Arduino-PID-Library v1](#), Github Repository
- [3] mothertel: [Rotary-Encoder-Library](#), Github Repository
- [4] Paul Stoffregen: [Quadrature Encoder Library for Arduino](#), Github Repository
- [5] Infineon: [Digitaler Hall-Sensor TLE4905L](#), Datenblatt
- [6] fischertechnik: [Datenblatt Encodermotor 24V Art.-Nr. 144643](#). 19.05.2017.

Elektronik

Fahrtregler (2): Vom Widerstandsdraht bis zur PWM

Peter Krijnen

Im ersten Teil dieser kleinen Beitragsreihe habe ich bereits angedeutet, dass ich in Teil 2 auf einen von mir mit den Silberlingen gebauten PWM-Regler eingehen werde. Wie soll das gehen?

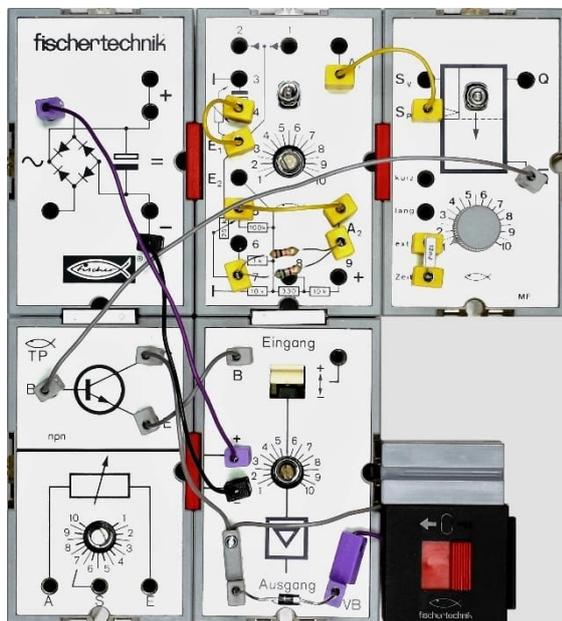


Abb. 20: Fahrtregler mit Silberlingen

Rezept

Man Nehme ein stabilisiertes Netzteil, einen Gleichrichterbaustein (GB1), einen Taktgenerator (G1), ein Mono-Flop (MF1) zum Einstellen der Impulse, eine Endstufe (TP1 + VB1) und einen Schalter (S1) zum Umpolen.

Gleichrichterbaustein

Ich habe vor langer Zeit herausgefunden [1], dass der Gleichrichterbaustein keine stabile Spannung ausgibt. Ich habe daher zwei meiner GBs mit einem Stabilisator-IC ausgestattet: einen mit einem 7805 für die 5 V und einen mit einem 7809 für die 9 V

der Silberlinge. Dazu habe ich den Gleichrichter und den Kondensator aus dem Gehäuse genommen und durch eine Platine ersetzt, auf der ich den Gleichrichter des GB, den 7809 und einen kleineren Kondensator von 1000 μF platziert habe. Für einen besseren Betrieb befindet sich am Ausgang ein Kondensator von 47 μF .

Grundbaustein

Als Impulsgeber wird der Grundbaustein GB1 verwendet. Abb. 20 zeigt, dass ich das normale Setup verwende: Anschluss A2 mit Anschluss 5 (100 k Ω) verbunden. Deutlich zu erkennen ist, dass der Anschluss A2 mit Anschluss 7 (Potentiometer) nicht mit einem Draht verbunden ist, sondern mit zwei parallel geschalteten Widerständen. Ich habe das gemacht, weil die Potentiometer in den Silberlingen nie richtig funktionieren wollten. Nach mehreren Tagen des Experimentierens mit selbstgebaute Widerstands- und Kapazitätsdekaden (Abb. 21) stellte sich heraus, dass der Aufbau dieser Schaltung nur dann richtig funktionierte, wenn ich das Potentiometer nicht benutzte. Es wäre besser gewesen, wenn ich das Potentiometer komplett umgangen hätte, indem ich A2 direkt mit E1 verbunden hätte, das habe ich aber nicht getan.

Am Ende habe ich zwei Widerstände von 10 k Ω und 560 Ω parallelgeschaltet. Dies ergibt inkl. Toleranzen einen Widerstand

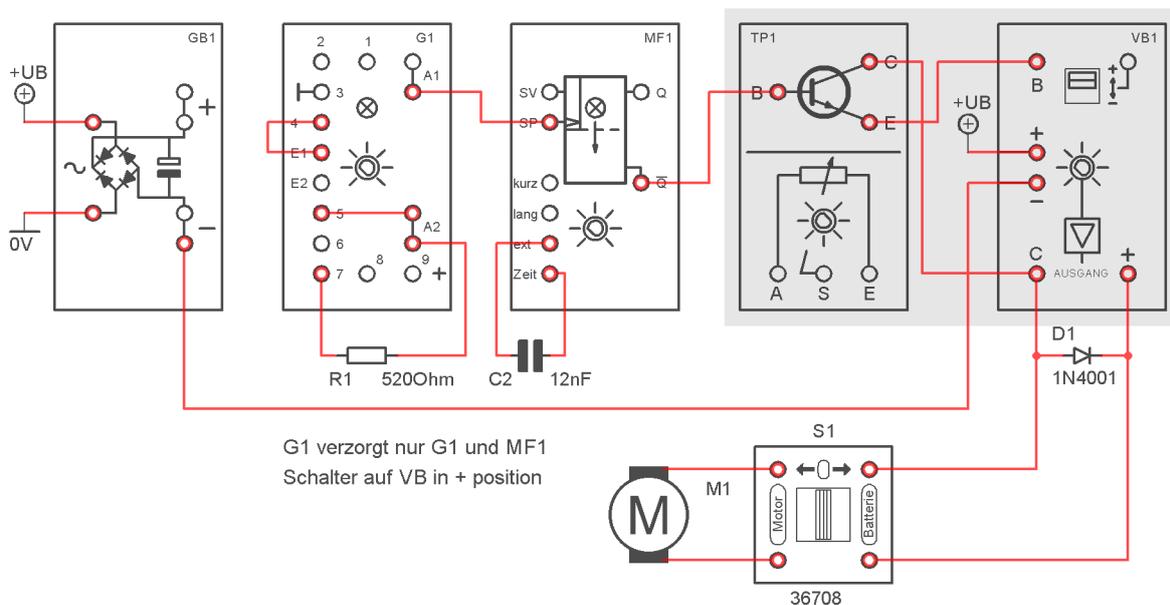


Abb. 21: Das Schaltbild

von etwa 520 Ω . In Reihe mit dem 220 Ω -Widerstand (R4) [2], der in Reihe mit dem Potentiometer liegt, ergibt dies eine Frequenz von 122 Hz. Das Potentiometer

bleibt am Linksanschlag stehen. Deshalb habe ich den Drehknopf weggelassen.

Mono-Flop

Ich verwende das Mono-Flop MF1, um die Breite (oder Länge) der von G1 erzeugten Impulse einzustellen. So, wie ich eine Widerstandsdekade benötigte, um den Widerstand zwischen A2 und 7 von G1 zu bestimmen, brauchte ich eine Kapazitätsdekade, um die Kapazität auf der MF (Zeit – ext) zu bestimmen. Nach den Experimenten stellte sich heraus, dass ein Kondensator von 12 nF in Kombination mit den 520 Ω auf G1 das beste Ergebnis liefert.

Dabei muss ich darauf hinweisen, dass es eigentlich umgekehrt ist. Das heißt, der Wert des Widerstands R1 hängt vom Wert von C2 ab. Ich habe zuerst den Wert des Kondensators ermittelt und erst dann versucht, den Widerstand für den Grundbaustein zu finden. Wichtig ist, dass das Potentiometer des MF auf Rechtsanschlag gedreht ist.

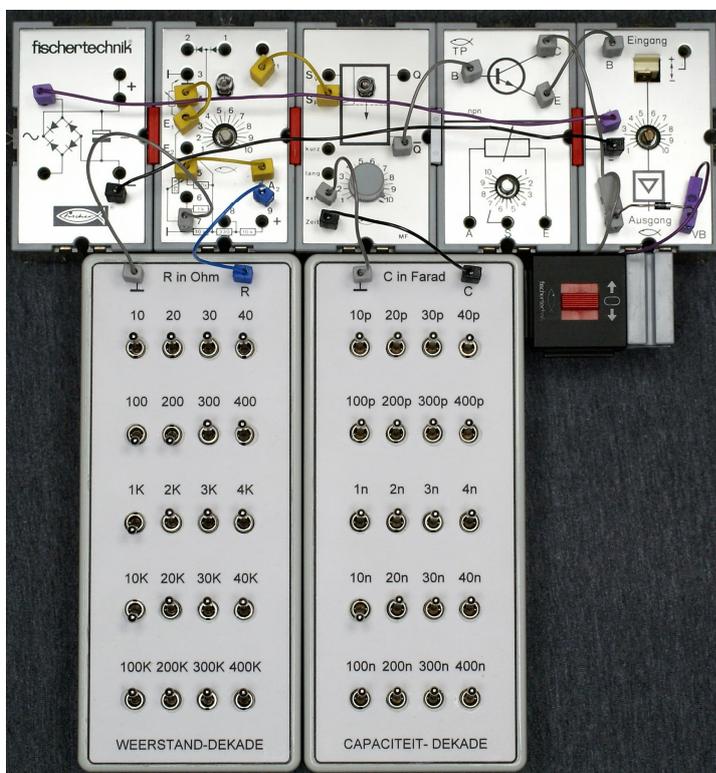


Abb. 22: Der Fahrregler in der Experimentierphase mit selbstgebaute Widerstands- und Kapazitätsdekaden

C1 μF	R1 $\text{k}\Omega$	F Hz		Width „-“ ms	Width „+“ ms
68	0,40	143,0	max	0,1	6,9
68	0,40	143,0	min	6,7	<0,4
47	0,53	143,0	max	0,1	6,9
47	0,53	143,0	min	6,7	<0,4
22	1,52	141,0	max	0,1	7,0
22	1,52	141,0	min	6,7	0,5
10	3,63	145,0	max	0,1	6,8
10	3,63	145,0	min	6,6	<0,4
2,2	15,50	145,0	max	0,1	6,8
2,2	15,50	145,0	min	6,5	<0,4

Tab. 1: Pulsweiten bei $C2 = 10 \text{ nF}$

Dann habe ich mit der Kapazitätsdekade einen Wert aus der E12-Reihe eingestellt, und schließlich mit der Widerstandsdekade den Wert von R1 ermittelt. Ich musste immer darauf achten, dass die Impulsbreite möglichst groß ist (Abb. 27), ohne dass das MF kippt [3].

C1 μF	R1 $\text{k}\Omega$	F Hz		Width „-“ ms	Width „+“ ms
68	0,53	120,0	max	0,1	8,3
68	0,53	120,0	min	7,9	<0,4
47	0,69	119,0	max	0,1	8,3
47	0,69	120,0	min	7,9	<0,4
22	1,84	119,0	max	0,1	8,3
22	1,84	119,0	min	7,9	<0,4
10	4,40	120,0	max	0,1	8,3
10	4,40	120,0	min	7,9	<0,4
2,2	19,00	119,0	max	0,1	8,3
2,2	19,00	118,0	min	7,9	<0,4

Tab. 2: Pulsweiten bei $C2 = 12 \text{ nF}$

C1 μF	R1 $\text{k}\Omega$	F Hz		Width „-“ ms	Width „+“ ms
68	0,69	97,1	max	0,1	10,2
68	0,69	97,1	min	9,9	<0,4
47	0,90	97,1	max	0,1	10,2
47	0,90	97,1	min	9,9	<0,4
22	2,30	97,1	max	0,1	10,2
22	2,30	97,1	min	9,9	<0,4
10	5,50	96,1	max	0,1	10,3
10	5,50	97,1	min	9,9	<0,4
2,2	23,00	98,0	max	0,1	10,2
2,2	23,00	98,0	min	9,9	<0,4

Tab. 3: Pulsweiten bei $C2 = 15 \text{ nF}$

Da der Kondensator im Grundbaustein eine viel zu große Abweichung hat (68 μF statt 47 μF), habe ich mehrere Werte ausprobiert. In Tab. 1, 2 und 3 habe ich alle Messdaten für 10 nF, 12 nF und 15 nF aufgelistet: „max“ = Poti auf MF am rechten Anschlag = Regler offen; „min“ = Poti am linken Anschlag = Regler geschlossen.

Aus den Tabellen kann man ersehen, dass $C2$ die Frequenz bestimmt. Wird für $C1$ ein anderer Wert gewählt, muss R1 jedes Mal neu bestimmt werden. Es ist sehr umständlich, wenn man das mit einzelnen Widerständen tun muss. Natürlich kann man das mit dem internen Potentiometer machen, aber diese sind nicht sehr gut (wie bereits weiter oben erwähnt). Am bequemsten gelingt das z. B. mit einem 25-Gang-50-k Ω -Spindeltrimmer [5].

Warum habe ich für $C2$ einen Wert von 12 nF gewählt? Ganz einfach: Ich habe mehrere Kondensatoren gemessen und bin auf eine Kapazität von 12,1 nF gekommen. Die anderen hatten alle eine viel größere Abweichung.

Oszilloskop

Zunächst wollte ich alle Daten, die ich während der Experimente gefunden hatte, in einem 3D-Diagramm visualisieren. Das Ergebnis war allerdings wenig übersichtlich. Ich denke, dass mehrere Bilder meines Oszilloskops ein besseres Bild vermitteln.

Auf den Bildern 23 bis 28 sehen wir Folgendes:

- Kanal 1 (gelb): Frequenz und Pegel auf A1 des Grundbausteins.
- Kanal 3 (rot): Impulsbreite auf \bar{Q} des MF (oder B vom TP).
- Kanal 4 (blau): Kollektor C des VB.

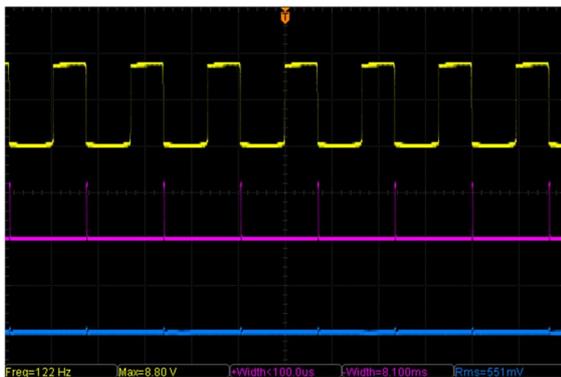


Abb. 23: Gelb = Taktfrequenz auf A1 von G1.
Rot = Basis B von TP.
Blau = Kollektor C von VB



Abb. 24: Die Last (zwei Lämpchen) ist mit einem Schalter ([36708](#)) an die Leistungsstufe angeschlossen

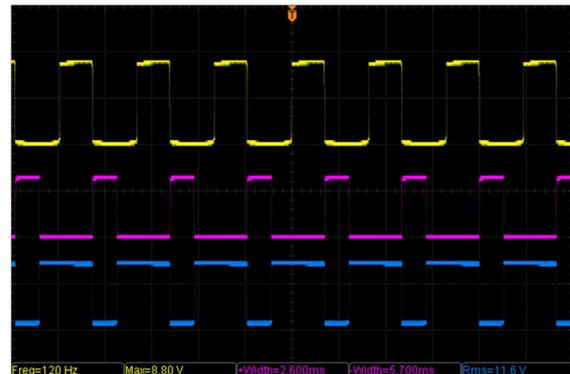


Abb. 25: Potentiometer von MF1 auf 4

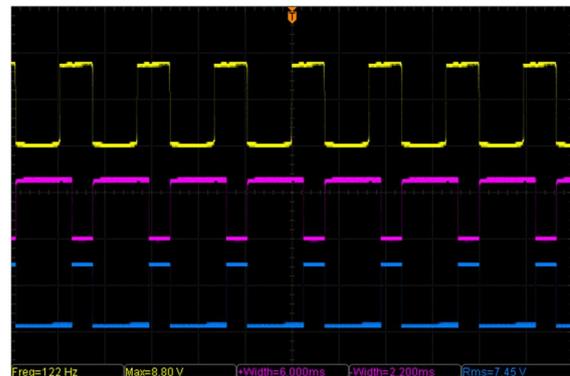


Abb. 26: Potentiometer von MF1 auf 7

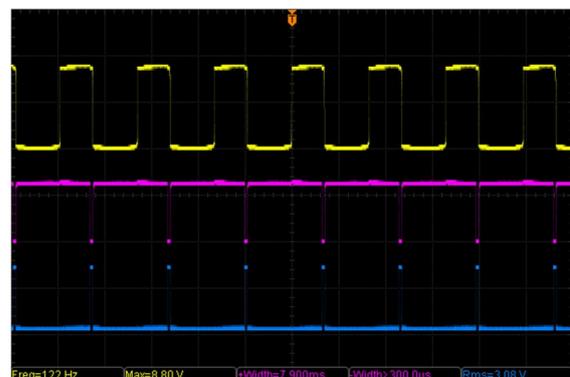


Abb. 27: Potentiometer von MF1 auf 10

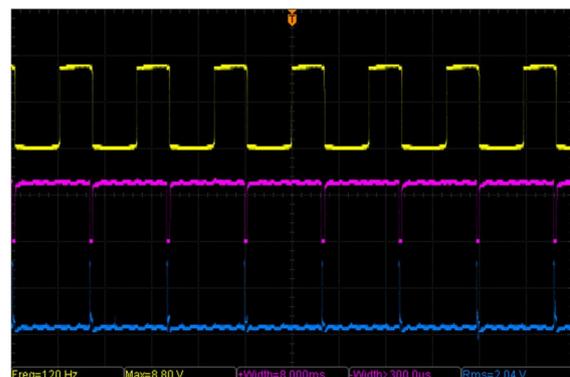


Abb. 28: Ein XM-Motor statt der beiden Lämpchen als Last

Leistungsstufe

Um einen Motor zu steuern habe ich eine Leistungsstufe gebaut, indem ich die Transistoren im TP und im VB als Darlington-Transistor geschaltet habe. Die Idee dazu fand ich auf Seite 15 des Begleitheftes von u-t4 [4].

Allerdings muss berücksichtigt werden, dass die in beiden Modulen verwendeten Transistoren vom Typ BC108 nur 200 mA vertragen. Werden beide als Darlington zusammengeschaltet sind das insgesamt 400 mA. Da ich es regelmäßig schaffe, zu hohe Ströme durch die Module fließen zu lassen, musste ich beide BC108 durch einen BC517 ersetzen. Diese sind Pin-kompatibel, können aber 400 mA verarbeiten.

Polwenden

Wenn wir zusätzlich den Polwende-Schalter (36708) erwenden, um die Polarität des Ausgangs umzukehren, erreichen wir den gleichen Effekt wie beim Power Controller (139778), allerdings mit einigen Einschränkungen.

Stabilität

Erstens hängt die Stabilität von der Versorgungsspannung ab. Ist diese nicht stabil, schwankt die vom Grundbaustein erzeugte Frequenz stark. Wenn man einen Motor anschließt, wird das noch schlimmer. Deshalb habe ich den GB modifiziert und die Endstufe TP + VB an ein separates Netzteil angeschlossen. Dazu habe ich die roten Zwischenstecker durch selbstgebaute weiße Stecker ersetzt (die waagerechten Zwischenstecker in Abb. 20), die komplett aus Kunststoff bestehen und somit die Leistungsstufe von der Steuerung isolieren. GB1 speist jetzt nur noch G1 und MF1.

Zweitens ist auch die Langzeitstabilität nicht gut. Wenn man den Controller einschaltet, funktioniert er entweder richtig oder nicht richtig. Umgekehrt ist es möglich, dass der Regler einige Zeit, nachdem er eingeschaltet wurde, plötzlich anfängt,

abzuweichen. Als ich die Abb. 23 bis 27 mit zwei in Reihe geschalteten Lämpchen als Last gemacht habe, funktionierte der Controller zum Glück noch gut. In Abb. 28 wurden die Lämpchen durch einen XM-Motor ersetzt.

Drittens kann der Regler nur mit dem Polwendeswitcher komplett abgeschaltet werden.

Reproduzierbarkeit

Oder besser gesagt, das Fehlen derselben: Aufgrund der Qualität der in den 1970er Jahren verwendeten Komponenten gibt es große Unterschiede zwischen den Silberlingen.

In Tab. 4 sehen wir die Abweichungen von C2, R4 und P1 von vier Grundbausteinen [2]. C2 sollte 47 μ F betragen; das ist zumindest auf den Kondensatoren aufgedruckt. Aber tatsächlich ist die Kapazität deutlich höher. Dasselbe gilt für R4 und P1. Diese sollten 220 Ω und 25 k Ω groß sein. Die Werte für den 100-k Ω -Widerstand an Pin 5 und den 1-k Ω -Widerstand an Pin 6 stimmen allerdings.

	C2 μ F	R4 Ω	P1 k Ω
Soll	47,00	220	25,00
G1	67,30	230	22,76
G2	61,80	244	50,09
G3	66,30	222	23,90
G4	66,10	256	28,43

Tab. 4: Abweichungen vom Sollwert von vier Grundbausteinen

Es dürfte klar sein, dass die Mono-Flops auch nicht alle gleich sind. Dank Jeroen Regtien, der mir einige seiner Silberlinge zur Verfügung stellte, konnte ich einige Messungen an vier MFs durchführen. Auch bei den MFs weichen die Werte der Kondensatoren und der Potentiometer erheblich voneinander ab.

Die hier angegebenen Werte sind natürlich nur beispielhaft. Bei Verwendung eines anderen Grundbausteins oder Mono-Flops musste ich den Wert für R immer neu definieren. Tab. 5 zeigt die Ergebnisse meiner Messungen.

	C1 μF	C2 μF	R1 kΩ	R1 + P1 MΩ
Soll	47,0	2,2	10,000	1,010
MF1	60,7	2,7	10,149	1,264
MF2	64,5	2,8	10,011	1,310
MF3	59,7	2,8	10,470	1,482
MF4	69,4	2,9	9,951	1,243

Tab. 5: Abweichungen vom Sollwert von vier Mono-Flops

Fazit

Ist es ein Wunder, dass der Regler überhaupt funktioniert?

So weit möchte ich nicht gehen. Aber es erfordert einige Anstrengungen, um ihn zum Laufen zu bringen. Wer ein Oszilloskop hat, ist im Vorteil, denn ohne Oszil-

loskop lässt sich der Regler kaum einstellen.

Durch die Verwendung von Sägezahn- anstatt von Rechteckspannungen konnte ich einen Regler erstellen, der besser funktioniert. Wie das aussieht, zeige ich euch im dritten Teil dieser Beitragsreihe.

Quellen

- [1] Peter Krijnen: *Silberlinge: Original oder Nachbau (Teil 2)*. [ft:pedia 2/2021](#), S. 80–89.
- [2] Peter Krijnen: *Silberlinge: Original oder Nachbau (Teil 4)*. [ft:pedia 3/2021](#), S. 38–48.
- [3] Peter Krijnen: *Silberlinge: Original oder Nachbau (Teil 3)*. [ft:pedia 2/2021](#), S. 90–100.
- [4] Fischerwerke: *fischertechnik u-t4*. Begleitbuch auf docs.fischertechnikclub.nl.
- [5] Conrad Electronic SE: *Bourns 3296X-1-503LF 3296X-1-503LF Spindeltrimmer 25-Gang linear 0.5 W 50 kΩ 9000 ° 1 St.* Auf conrad.de.

Elektronik

Digitale Uhr mit Echtzeitgenerator

Hans-Christian Funke

Im Heft 4/2020 [1] habe ich dieses Modell schon einmal vorgestellt, allerdings wurde dort der Takt durch den Taktgenerator generiert bzw. durch den Controller. Dies führt unweigerlich früher oder später zu einer Ungenauigkeit der Uhr. Mit dem Echtzeitgenerator, dem ein quarz-gesteuerter Takt zu Grunde liegt, kann eine sehr genaue Taktung ohne Abweichung erzielt werden; er ist auch an einem Controller einsetzbar.

Das Schaltbild in Abb. 3 zeigt, wie die Zählermodule (70001 oder 70002) für Stunden, Minuten und Sekunden, die Taster für die Zeiteinstellung sowie der Echtzeitgenerator an den Controller angeschlossen werden können. Man könnte auch noch einen Taster für die Sekunden vorsehen, aber darauf habe ich verzichtet, weil man die Uhr einfach zu den vollen Minuten starten kann und somit die Sekunden nicht genau einstellen muss.



Abb. 1: Digitale Uhr (Stunden-Minuten-Sekunden) mit grüner LED

Programme

Zur Steuerung der Uhr habe ich drei Programmstränge definiert. Der Programmstrang „Taktabtastung“ (Abb. 2) wertet den Takt des Echtzeitgenerators aus. Ein zweiter Programmstrang „Steuerung“ (Abb. 4) wird immer dann aktiv, wenn die Variable aus dem ersten Programmstrang gesetzt wurde. Er verwaltet sowohl die Variablen für Sekunden, Minuten und Stunden als auch die Ansteuerung der Zählermodule für die Anzeige der Zeit. Der dritte Programmstrang „Zeiteinstellung“ (Abb. 5) ist für das Setzen der beiden Variablen „Minuten“ und

„Stunden“ über die entsprechenden Taster zuständig.

Programmablauf

Taktabtastung

Hier wird nach jedem zweiten Flankenwechsel eine globale Variable „Takt 1Hz“ gesetzt. Der Echtzeitgenerator (70051) liefert einen Takt von 2 Hz, was einer Periodendauer von 0,5 s entspricht. Durch das Abwarten von zwei Takten erhält man den 1-s-Takt.

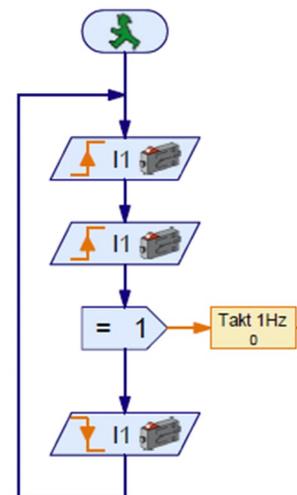


Abb. 2: Programmteil zur Taktumwandlung in einen 1-Hz-Takt (= 1 Sekunde Periodendauer)

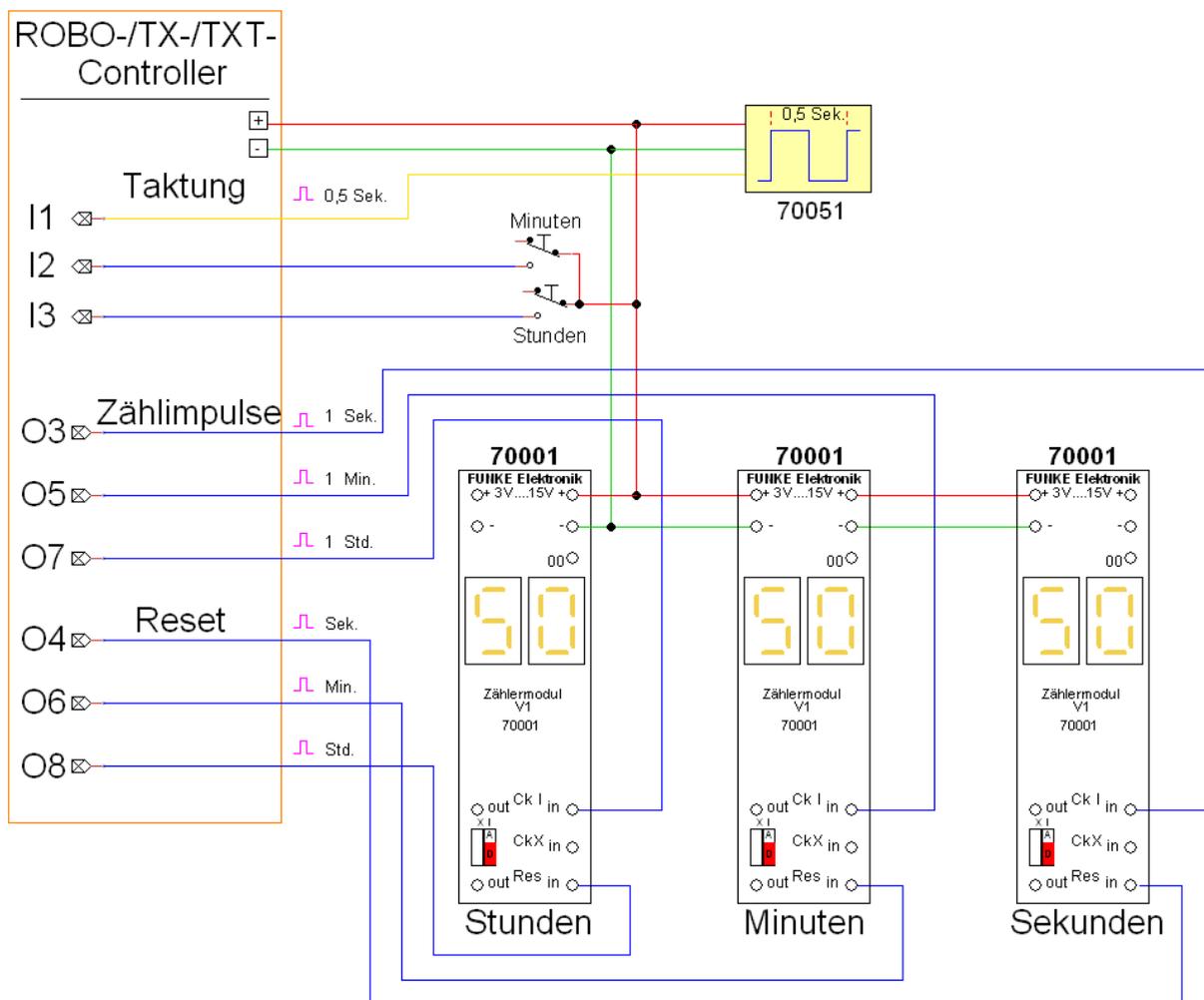


Abb. 3: Anschaltung von Echtzeitgenerator, Schalter und Zählermodule an einem Controller

Steuerung

Dieser Programmstrang wird immer erst aktiv, wenn die globale Variable „Takt 1Hz“ aus dem ersten Programmstrang gesetzt ist. So lange läuft das Programm in einer Warteschleife.

Die Zählvariable für Sekunden wird um eins erhöht und ein Impuls an das Zählermodul für Sekunden (an O5 angeschlossen) gesendet. Anschließend erfolgt die Prüfung, ob die Variable „Sekunden“ den Wert 60 erreicht. Wenn nicht, geht es zurück zur Warteschleife. Anderenfalls werden die Variable „Sekunden“ und das Zählermodul für die Sekundenanzeige zurückgesetzt und es erfolgt die gleiche Prüfung für die Minuten. Entsprechend werden hier die Variable

„Minuten“ und das Zählermodul für die Minutenanzeige zurückgesetzt, wenn der Variablenwert von „Minuten“ den Wert 60 erreicht hat. Analog erfolgt die Überprüfung der Variable für Stunden, die aber schon beim Erreichen eines Wertes von größer als 23 zurückgesetzt wird.

Zu beachten ist, dass der Impuls, der an das Zählermodul gesendet wird, eine Mindestlänge von 10 ms haben muss (Wartezeit 0,01 s), damit das Signal am Ausgang des Controllers sichtbar wird.

Zeiteinstellung

Dieser Teil des Programms wertet die Taster „Minuten“ und „Stunden“ aus. In einer Schleife wird ständig geprüft, ob einer der beiden Taster gedrückt wurde. Sobald

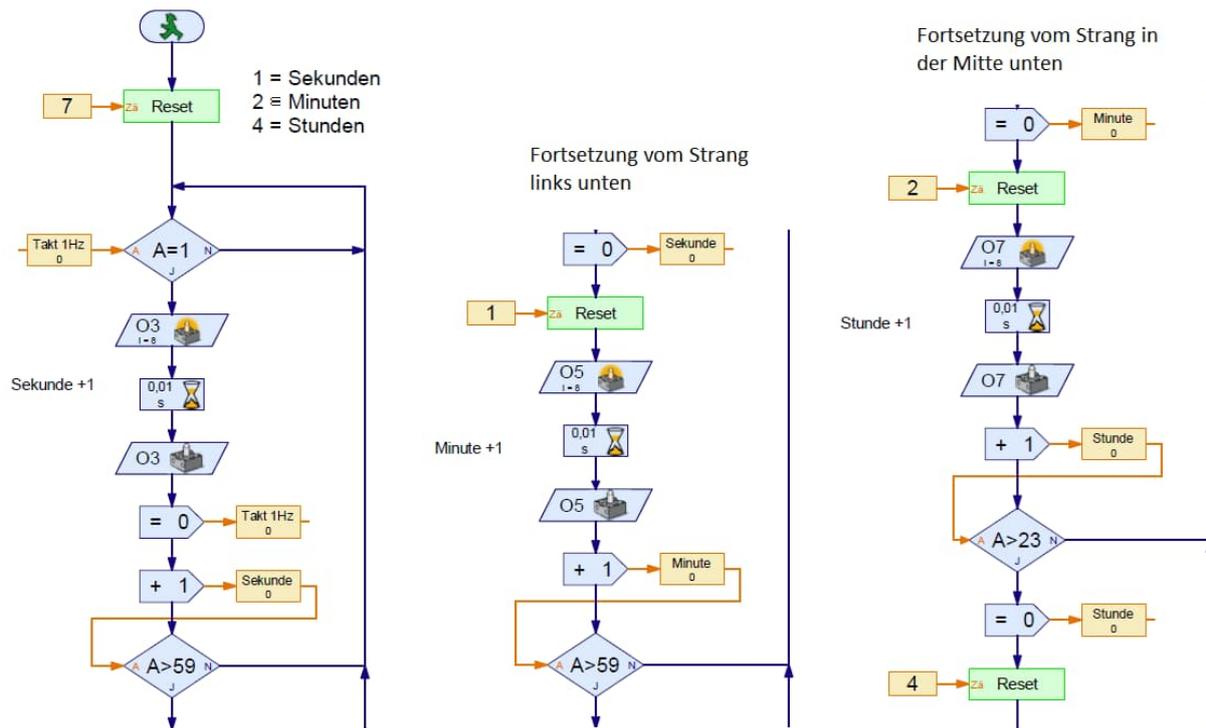


Abb. 4: Programmteil zur Steuerung der Uhr

ein Taster gedrückt wurde, wird das entsprechende Zählermodul um eins weitergezählt (O5 für Minuten und O7 für Stunden) und die dazugehörige Variable um eins erhöht. Für den Fall, dass der höchste zulässige Wert – bei Minuten 59 und bei Stunden 23 – überschritten wird, werden die Anzeige und die zugehörige Variable zurückgesetzt.

Hier wurde im Programmablauf nach dem Zurücksetzen der Anzeige eine Wartezeit von 30 ms eingefügt. Diese Wartezeit hat nur eine optische Funktion. Hält man den Taster gedrückt, dann läuft die Erhöhung zwar entsprechend schnell durch, aber noch so langsam, dass man es nachverfolgen kann und genug Zeit bleibt, rechtzeitig beim gewünschten Wert zu stoppen.

Unterprogramm „Reset“

Dieses Unterprogramm habe ich unverändert übernommen – es wurde bereits im Artikel in Ausgabe 4/2020 beschrieben [1].

Schlussbemerkung

Das RoboPro-Programm kann wieder heruntergeladen werden [2]. Für Fragen und Hinweise stehe ich gerne unter der E-Mail-Adresse elektronik@funke4you.de zur Verfügung.

Quellen

- [1] Hans-Christian Funke: *Elektronik-Module (Teil 5)*. [ft:pedia 4/2020](#), S. 80–94.
- [2] Hans-Christian Funke: *Modell Digitale Uhr_2023_2.rpp*. RoboPro-Programm im [Downloadbereich der ftc](#), 2023.

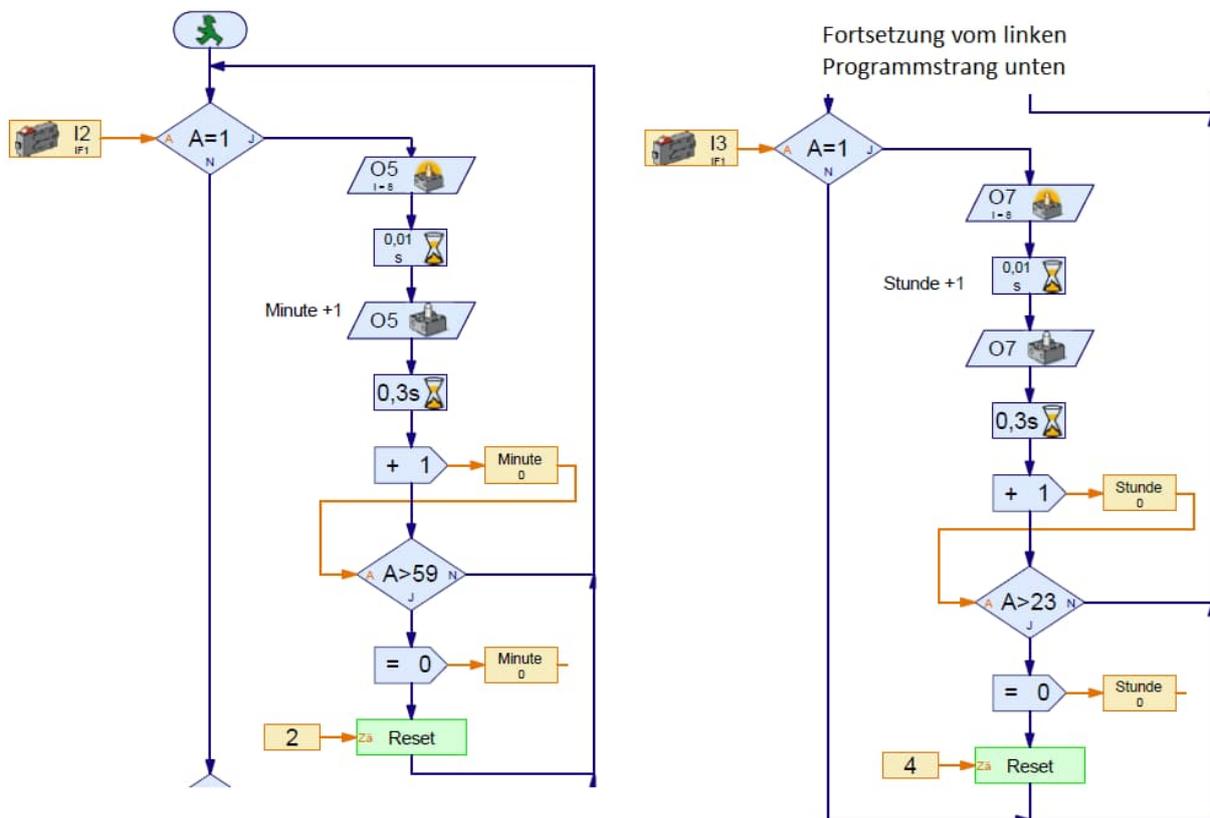


Abb. 5: Programmteil zum Setzen der Variablen „Minuten“ und „Stunden“



Abb. 6: Das Uhrenmodell am Controller (LED-Anzeige grün – gelb – rot für Std. – Min. – Sek.)

Elektronik

Silberlinge: Original oder Nachbau (Teil 11)

Peter Krijnen

Ich schaue regelmäßig im Bilderpool auf ftcommunity.de nach, ob neue Fotos gepostet wurden. Vor einigen Jahren habe ich unter Basteleien – Elektronik – IC-Praktikum [1] eine Reihe mir unbekannter Module gefunden. Zusätzlich zu den fünf Modulen, die bereits in Teil 10 besprochen wurden, scheint es noch viele weitere Module zu geben. Diese wurden von fischer-technik bisher in Industrie-, Ausbildungs- und Schulmodellen eingesetzt. Warum die Bilder dieser Module unter „Basteleien“ platziert sind, ist mir nicht ganz klar. Das hielt mich jedoch nicht davon ab, selbst ans Basteln zu gehen.

Einschließlich der fünf bereits bekannten Module habe ich auf dem Übersichtsbild und den anderen Einzelbildern insgesamt folgende Module gefunden:

- 2 – IC-Bausteine
- 2 – Schwellenwertschalter
- 3 – Leistungsstufe
- 4 – Relais

- 2 – IC-Spannungsversorgung
- 2 – Taktgeber
- 2 – Universal - Zähler

In diesem Teil behandle ich:

- 1 – Schwellenwertschalter
- 1 – Leistungsstufe
- 1 – Taktgeber

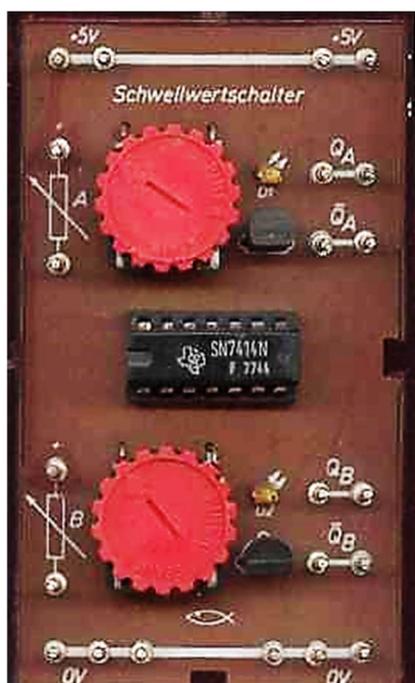


Abb. 324: Industrieversion eines vereinfachten Schwellenwertschalters ([36295](#))

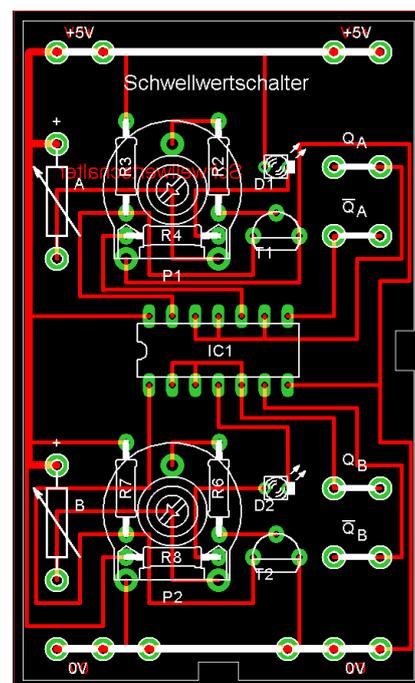


Abb. 325: Schwellenwertschalter 2, Layout

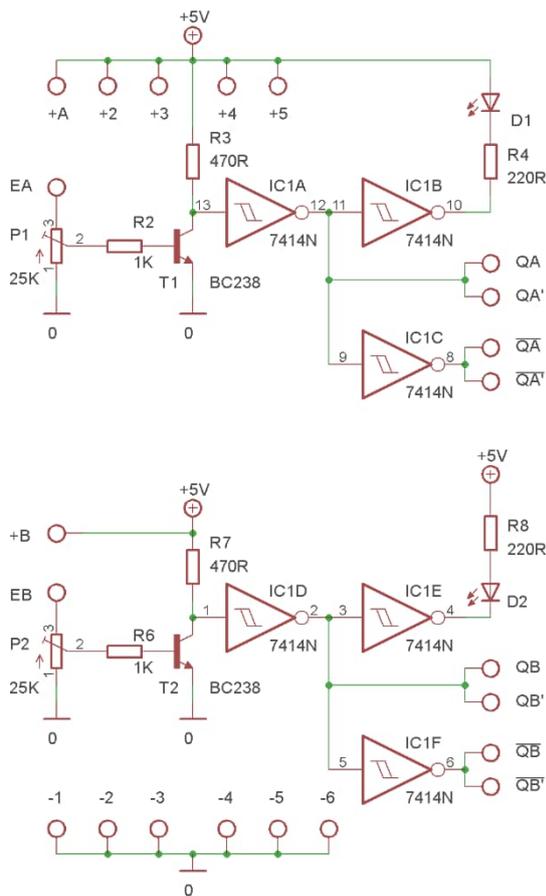


Abb. 326: Schwellenwertschalter 2, Schaltbild

Schwellenwertschalter 2

Ich weiß nicht, ab wann diese Module verwendet wurden.

Auf Abb. 324 sehen wir eine vereinfachte Version des Schwellenwertschalters (36295). Vergleichen wir dieses Bild mit dem von Abb. 303 [2] sehen wir, dass beide „Br“-Pins entfernt wurden. Beide Eingänge haben nur einen Pin, und die +5V- und 0V-Pins links vom IC wurden ebenfalls entfernt. Andererseits verfügen nun beide Eingänge über einen eigenen „+“-Pin. Vergleicht man das Schaltbild von Abb. 326 mit dem von Abb. 305 [2] wird der Unterschied etwas deutlicher.

Leistungsstufe 2

In der zweiten Version der Leistungsstufe wurden die „C“-Pins durch „Last_A“ und „Last_B“ ersetzt. Die Eingangspins heißen jetzt „A“ und „B“ und jeder hat nur noch einen Pin übrig. Beide haben jetzt jeweils einen zusätzlichen „+“-Pin. Andererseits wurden zwei „+“-Pins an der Oberkante des Moduls entfernt. Zum Vergleich schaut man sich Abb. 306 [2] an.

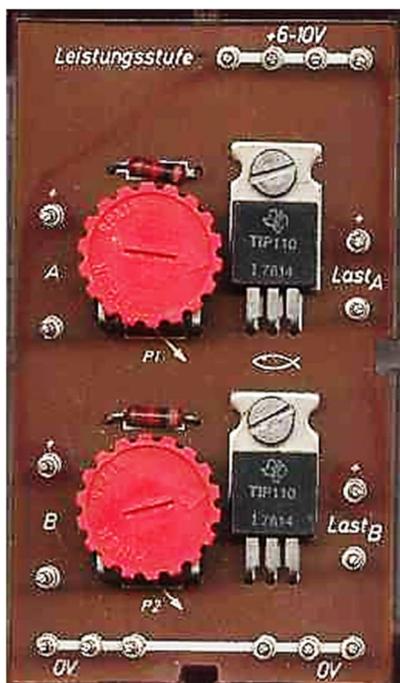


Abb. 327: Leistungsstufe 2

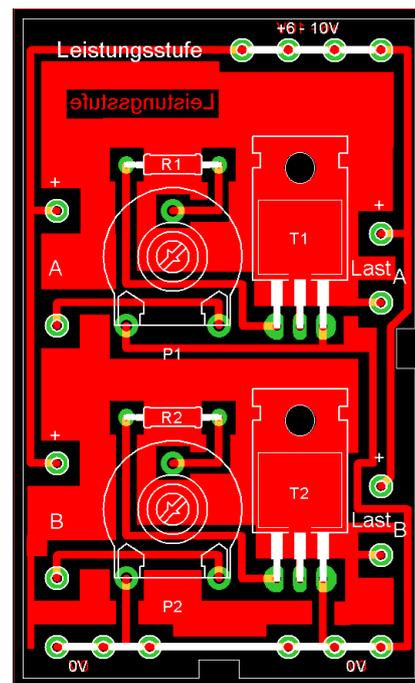


Abb. 328: Leistungsstufe 2, Layout

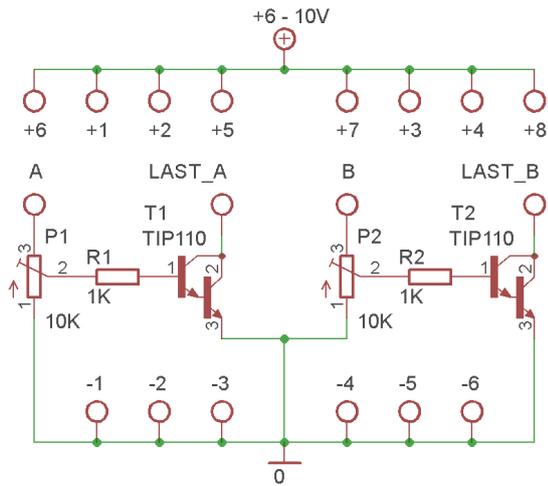


Abb. 329: Leistungsstufe 2, Schaltbild

Taktgeber 1

Der Aufbau des Taktgebers ist recht einfach. Wenn wir uns das Schaltbild in Abb. 332 ansehen, sehen wir, dass ein halber SN7413 die Basis der Schaltung bildet. Der SN7413 ist ein „Dual 4-Input Positive-NAND Schmitt Trigger“ [3]. Warum es konkret eine Schmitt-Trigger-Version sein muss, erschließt sich mir nicht. Eine „normale“ Version des NAND funk-

tioniert jedoch nicht in einer Oszillator-schaltung.

Die frequenzbestimmenden Elemente sind das Potentiometer und die Kondensatoren. Es stehen drei Kondensatoren zur Verfügung, die mittels einer Drahtbrücke auf „0V“ geschaltet werden müssen. Sollten die Werte der Kondensatoren nicht ausreichen, kann ein externer Kondensator an den „ext“-Pin angeschlossen werden.

Da ich selbst keinen Taktgeber und daher nicht alle Werte der Kondensatoren und des Potentiometers habe, fing ich einfach an zu experimentieren. Tab. 1 zeigt, was dabei herausgekommen ist.

Als Kondensatoren habe ich die des Mono-Flop [4] verwendet: 47 μ F, 2,2 μ F, und für den kleinen habe ich 100 nF gewählt. Da es im Mono-Flop ein 1-M Ω -Potentiometer gibt, habe ich damit begonnen. Es stellte sich schnell heraus, dass dies viel zu hoch war. Ich habe mich daher auf 1, 10, 25, 50 und 100 k Ω beschränkt. Tabelle 1 zeigt, welche Frequenzen damit generiert werden können.



Abb. 330: Taktgeber 1

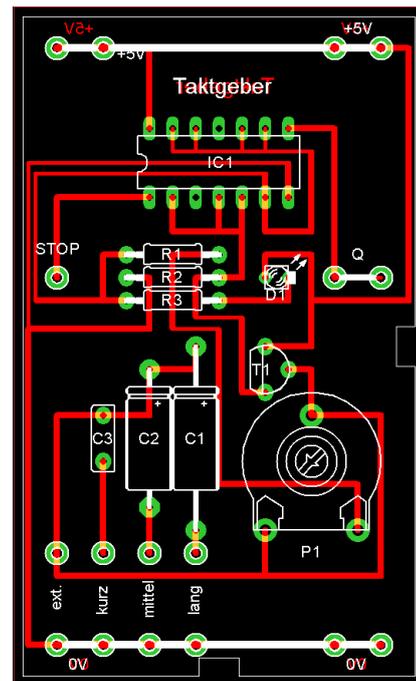


Abb. 331: Taktgeber 1, Layout

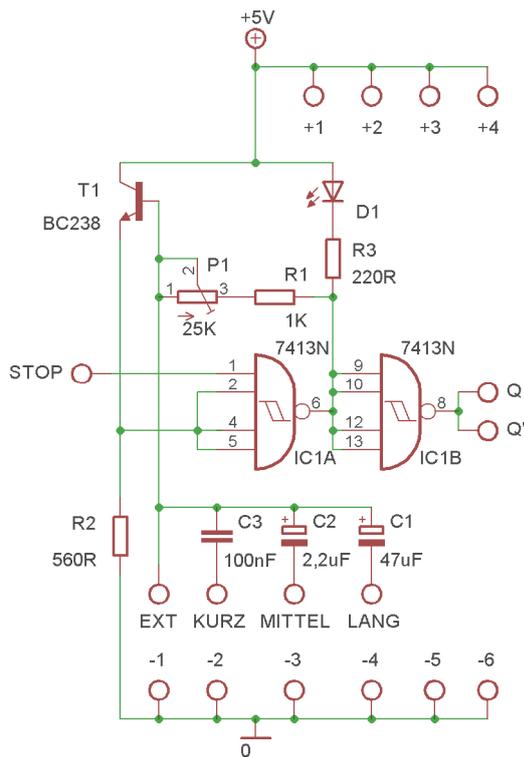


Abb. 332: Taktgeber 1, Schaltbild

Es ist klar, dass die Kondensatoren die Höhe der Frequenz bestimmen und dass das Potentiometer die Breite des Bereichs bestimmt. Mit einem Kondensator von 10 nF erreichen wir eine Frequenz von max. 100 kHz. Wenn wir ein 1-k Ω -Potentiometer verwenden, liegt der Bereich zwischen 55 kHz und 100 kHz. Wenn wir nun ein Potentiometer von 100 k Ω nehmen, liegt der Bereich zwischen 1 kHz bis 100 kHz.

Das zweite NAND dient als invertierender Treiber für die beiden Ausgangspins Q. Die LED sorgt für die nötige Anzeige. Allerdings ist das hochfrequente Blinken der LED natürlich nicht mehr sichtbar.

Der Taktgeber funktioniert, wenn der STOP-Pin offen bleibt. Um den Taktgeber zu blockieren, muss der STOP-Pin mit 0V verbunden werden.

Nachbau

Für den Nachbau kann man natürlich die Layouts der Originalmodule verwenden. Mein Ausgangspunkt war das Silberling-

Gehäuse aus hobby-Labor 1, wie ich es auch bei den Modulen in Teil 10 genutzt hatte.

Pot.	Cap.	F _{min}	F _{max}
1 k Ω	10 nF	55,2 kHz	101 kHz
	100 nF	6,17 kHz	11,5 kHz
	2,2 μ F	284 Hz	520 Hz
	47 μ F	12,8 Hz	23,5 Hz
	100 μ F	6,25 Hz	11,4 Hz
10 k Ω	10 nF	8,85 kHz	102 kHz
	100 nF	934 Hz	11,4 kHz
	2,2 μ F	44 Hz	520 Hz
	47 μ F	2 Hz	23,5 Hz
	100 μ F	0,973 Hz	11,4 Hz
25 k Ω	10 nF	4,38 kHz	102 kHz
	100 nF	450 Hz	11,3 kHz
	2,2 μ F	21,6 Hz	520 Hz
	47 μ F	0,980 Hz	23,5 Hz
	100 μ F	0,478 Hz	11,4 Hz
50 k Ω	10 nF	2,02 kHz	102 kHz
	100 nF	207 Hz	11,4 kHz
	2,2 μ F	10 Hz	520 Hz
	47 μ F	0,454 Hz	23,5 Hz
	100 μ F	0,220 Hz	11,4 Hz
100 k Ω	10 nF	1,05 kHz	102 kHz
	100 nF	107 Hz	11,5 kHz
	2,2 μ F	5,2 Hz	520 Hz
	47 μ F	0,235 Hz	23,5 Hz
	100 μ F	0,114 Hz	11,4 Hz

Tab. 1: Frequenzen, die mit dem SN7413 erzeugt werden können



Abb. 333: Nachbau: Schwellenwertschalter 2

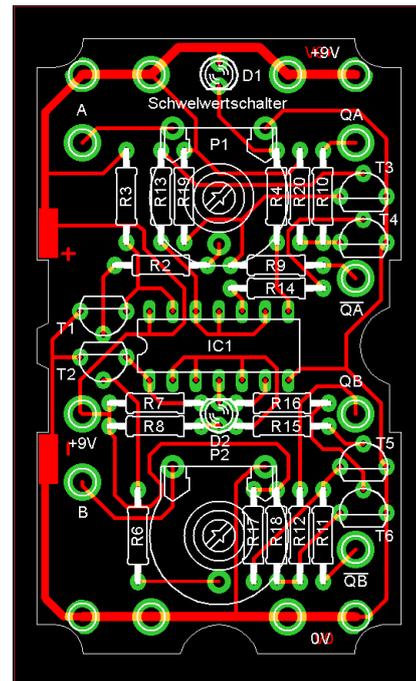


Abb. 334: Nachbau: Schwellenwertschalter 2,

Ich habe diese Versionen noch nicht selbst gebaut. Allerdings gefällt es mir zu sehr, sie zu entwerfen. Mein Ansatz war: Wie hätten sie aussehen können, wenn sie Silberlinge wären?

Schwellenwertschalter 2

Der Schwellenwertschalter 2 ist in den Abb. 333 bis 335 zu sehen. Vergleiche dies mit der Abb. 321 bis 323 in Teil 10.

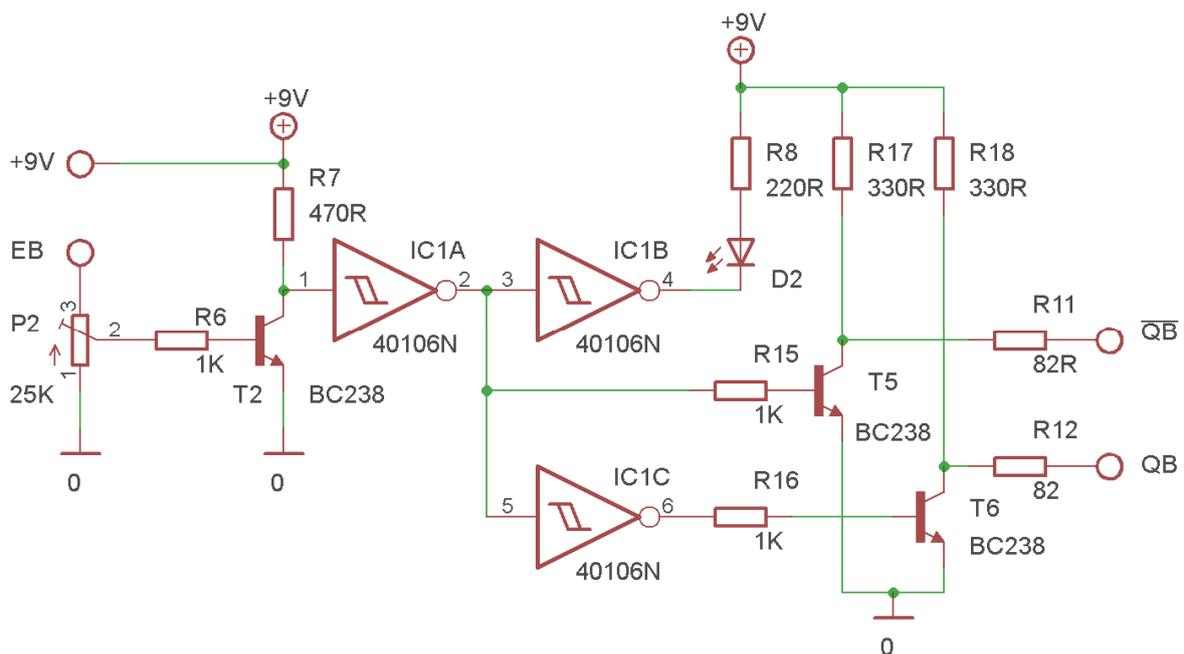


Abb. 335: Nachbau: Schwellenwertschalter 2, Schaltbild



Abb. 336: Nachbau: Leistungstufe 2

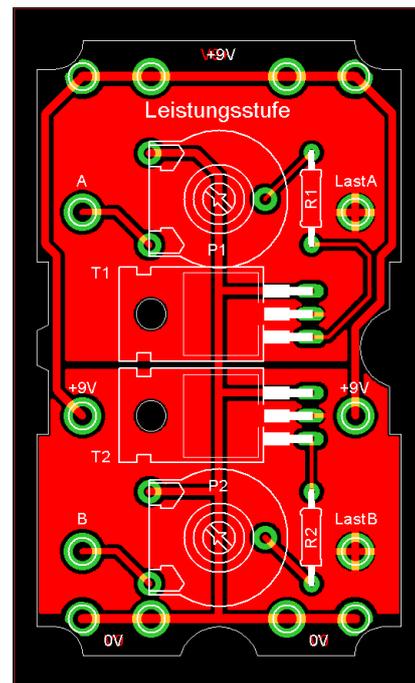


Abb. 337: Nachbau: Leistungstufe 2, Layout

Auch hier habe ich den SN7414 durch den CD40106 [5] ersetzt und ihm die gleichen Ausgangsstufen gegeben wie beim Grundbaustein.

Leistungstufe 2

Eine Silberling-Version der Leistungstufe 2 ist in Abb. 336 bis 338 zu sehen. Dies kann mit Abb. 318 bis 320 in Teil 10 verglichen werden.

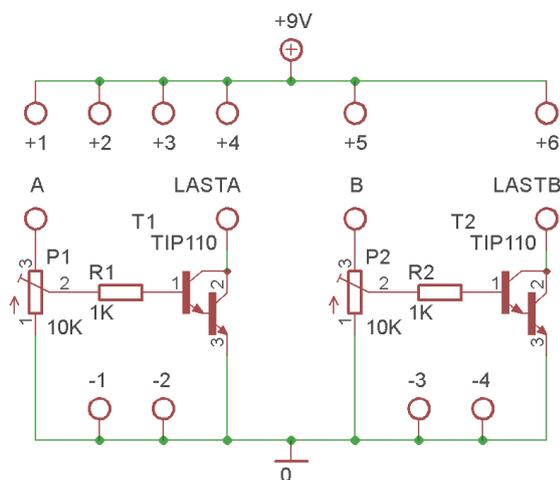


Abb. 338: Nachbau: Leistungstufe 3, Schaltbild

Taktgeber

Für den Taktgeber 1 benötige ich etwas mehr Text. Da es keine vergleichbare CMOS-Version des SN7413 gibt, musste ich mir einen CD4093 [6] zulegen. Dabei handelt es sich nicht um einen „DUAL 4 Input NAND“, sondern um einen „QUAD 2 Input NAND“. Für den Taktgeber werden nur zwei NANDs benötigt. Die anderen beiden habe ich für den Ausgang genutzt.

Es lässt sich nicht leugnen, dass meine Version des Taktgebers dem Mono-Flop [4] sehr ähnlich ist. Das war auch die Absicht. Kein „Sv“ oder „Sp“ und kein „Zeit“. Kein \bar{Q} , aber mit „Mittel“ und „STOP“.

Im Gegensatz zum SN7413 muss beim CD4093 der STOP-Pin mit einem Widerstand ($R6 = 10\text{ k}\Omega$) mit „+“ verbunden werden. Zum Blockieren muss er mit 0V verbunden werden.

Da der CD4093 nicht mit dem SN7413 identisch ist, sind die zu erzeugenden Frequenzen unterschiedlich. Tabelle 2 zeigt, was mit dem CD4093 erzeugt werden kann.



Abb. 339: Nachbau: Taktgeber 1

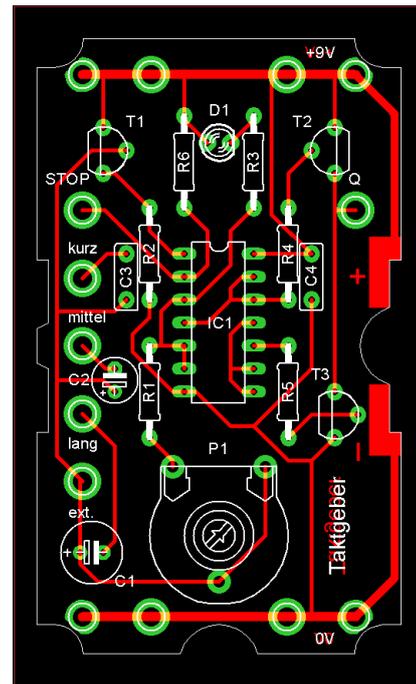


Abb. 340: Nachbau: Taktgeber 1, Layout

Universal-Zähler

Es war auch meine Absicht, den Universal-Zähler in diesem Beitrag zu behandeln, aber dann wäre der Beitrag doppelt so lang geworden. Deshalb findet ihr den Universal-Zähler im Teil 12 der Beitragsreihe.

Da ich selbst nicht über Industriemodule verfüge, habe ich einige Fotos aus dem Bilderpool übernommen. Das sind die Abb. 324, 327 und 330.

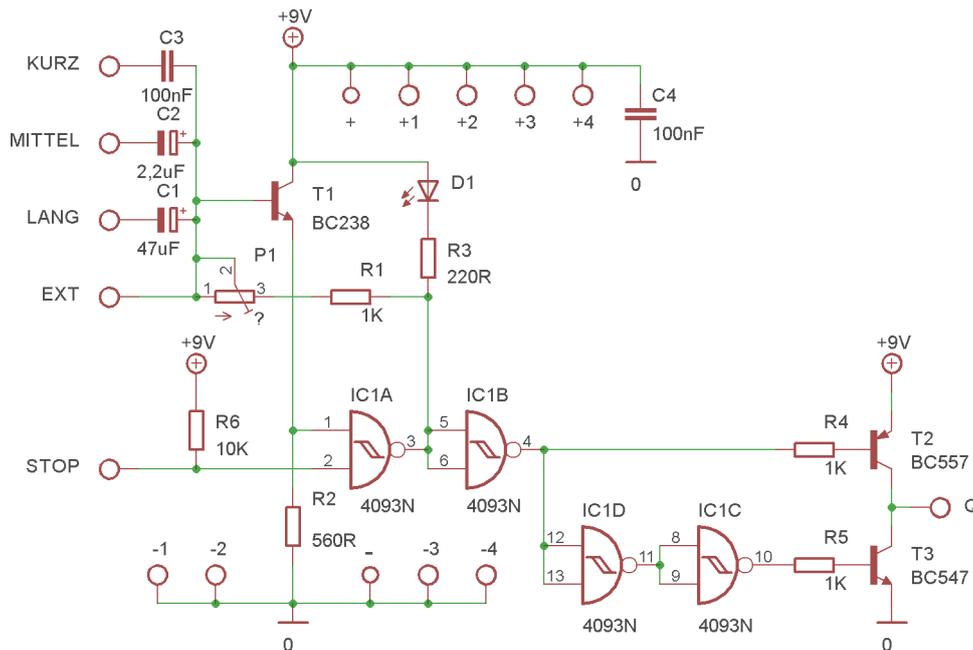


Abb. 341: Nachbau: Taktgeber 1, Schaltbild

Pot.	Cap.	F _{min}	F _{max}
1 kΩ	10 nF	73,5 kHz	125 kHz
	100 nF	11,2 kHz	20,3 kHz
	2,2 μF	398 Hz	709 Hz
	47 μF	17,8 Hz	31,6 Hz
	100 μF	8,86 Hz	15,2 Hz
10 kΩ	10 nF	10,1 kHz	124 kHz
	100 nF	19,1 kHz	20,7 kHz
	2,2 μF	52,3 Hz	704 Hz
	47 μF	2,37 Hz	31,6 Hz
	100 μF	1,15 Hz	15,2 Hz
25 kΩ	10 nF	2 kHz	125 kHz
	100 nF	286 Hz	20,4 kHz
	2,2 μF	12,2 Hz	709 Hz
	47 μF	0,575 Hz	31,8 Hz
	100 μF	0,131 Hz	15,3 Hz

Tab. 2: Frequenzen, die mit dem CD4093 erzeugt werden können

Quellen

- [1] IC-Praktikum [im Bilderpool](#) der ft Community.
- [2] Peter Krijnen: *Silberlinge: Original oder Nachbau (Teil 10)*. [ft:pedia 1/2023](#), S. 101–110.
- [3] Texas Instruments: *SN7413N Datasheet*. Auf [alldatasheet.de](#).
- [4] Peter Krijnen: *Silberlinge: Original oder Nachbau (Teil 6)*. [ft:pedia 1/2022](#), S. 60–70.
- [5] Texas Instruments: *CD40106B CMOS Hex Schmitt-Trigger Inverters*. Auf [ti.com](#), 1998 – 2017.
- [6] Texas Instruments: *CD4093B Types CMOS Quad 2-Input NAND Schmitt Triggers*. Auf [ti.com](#), 2003.

Elektronik

Sensoren

Hans-Christian Funke

Sensoren oder Messfühler dienen dazu, physikalische Größen wie Temperatur, Helligkeit oder Leitfähigkeit zu messen oder Veränderungen dieser Größen zu registrieren. Die althergebrachten Messfühler von fischertechnik waren klein und flexibel einsetzbar, aber diese gibt es nur noch auf dem Gebrauchtmarkt.

Viele werden sich noch an die Messfühler aus den Elektrobaukästen ec1, ec2 bzw. hobby 4 erinnern, mit denen großartige und interessante Experimente durchgeführt werden konnten. Diese Messfühler waren klein, einfach gehalten und oft in einem Leuchtstein untergebracht. Dafür gibt es jetzt den Umweltsensor ([167358](#)), der Luftdruck, -qualität, -feuchtigkeit und -temperatur erfassen kann – die Auswertung der Daten kann allerdings nur noch über einen Controller erfolgen.

Dieser Umstand hat mich dazu inspiriert, die alten Sensoren wieder aufleben zu lassen. Unter dem riesigen Angebot an elek-

tronischen Komponenten habe ich ein paar Bauelemente gefunden, die sich sehr gut mit meinen Elektronikmodulen sowie den fischertechnik-Elektronikbausteinen wie den Silberlingen, E-Tec ([108227](#)), Electronics ([152063](#)), Mini-Bots ([156499](#)) und E-Tronics ([182323](#)) verwenden lassen.

In Abb. 1 bis 4 sind die alten Messfühler zu sehen und im Vergleich dazu die neu erstellten Messfühler bzw. Sensoren in Abb. 5 bis 8. Es gibt einen Feuchtigkeitssensor (Abb. 5), zwei Temperatursensoren (Abb. 6 und 7) sowie einen Lichtsensor (Fotowiderstand, Fotozelle; Abb. 8).



Abb. 1: ft-Feuchtigkeitssensor



Abb. 2: ft-NTC-Temperatursensor



Abb. 3: ft-NTC-Temperatursensor



Abb. 4: ft-Fotowiderstand (Lichtsensor)

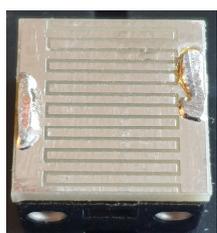


Abb. 5: Feuchtigkeitssensor



Abb. 6: PTC-Temperatursensor



Abb. 7: NTC-Temperatursensor

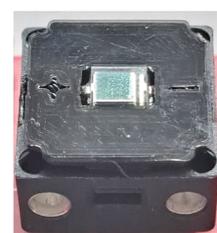


Abb. 8: Fotodiode (Lichtsensor)

Feuchtigkeitssensor

Der Feuchtigkeitssensor (70070) kann zur Feststellung von Feuchtigkeit (Luft, Flüssigkeit) eingesetzt werden. Er ist genauso aufgebaut und in einem Leuchtstein montiert wie das Pendant von fischertechnik seinerzeit.

In einer trockenen Umgebung ist der Widerstandswert unendlich groß. Je nach Feuchtigkeitsgehalt der Luft oder der Leitfähigkeit einer Flüssigkeit nimmt der Widerstand des Sensors ab. Bei Kontakt mit Wasser beträgt der Widerstandswert $\sim 100\text{ k}\Omega$ und bei einer gesättigten Salzlösung $\sim 10\text{ k}\Omega$.



Abb. 9: Feuchtigkeitssensor (70070) als Kontaktgeber

Bei Überbrückung wird der Widerstandswert sogar auf null Ω sinken. Damit ergibt sich eine zweite Funktion für den Feuchtigkeitssfühler: Er kann auch als Kontaktgeber fungieren. Überbrückt man mit einem Gegenstand aus Metall die Leiterbahnen auf dem Sensor, so wird der Kontakt (Schalter) geschlossen. Ideal eignen sich dafür die Metallkugeln von den Kugelbahnen – so könnte man den Sensor in den Bahnverlauf einbauen (Abb. 9), um Zähler oder andere Funktionen auszulösen.

Schaltungsbeispiel

Der Feuchtigkeitssensor sollte über ein Multifunktionsmodul angeschlossen werden (Abb. 10). Damit wird das Signal verstärkt. Es steht auch das invertierte Signal am Ausgang \bar{A} zu Verfügung. Über den Drehregler

kann eine Abstimmung auf den gewünschten Feuchtigkeitsgrad erfolgen. Über den Ausgang A kann ein anderes Elektronikmodul wie z. B. das Relaismodul (70018) angesteuert werden. Wird der Schwellenwertregler im Multifunktionsmodul nicht benötigt, könnte der Feuchtigkeitssensor auch direkt an ein Relaismodul angeschlossen werden.

Soll der Sensor als Kontaktgeber verwendet werden, z. B. direkt an einem Eingang eines Elektronikmoduls, dann ist zu beachten, dass es zu Prellimpulsen kommen kann. Zur Entprellung eignet sich ein Kondensator ($10\text{ }\mu\text{F}$), wie dieser in der Abb. 11 bei einem Zählermodul (70001) eingesetzt wurde. Sechs verschiedene Kondensatorwerte sind im Multifunktionsmodul bzw. im Kondensatorset (71046) enthalten.

Temperatursensoren

Temperatursensoren finden sich in vielen unserer Haushaltsgeräte, zum Beispiel zur Temperaturregulierung in unserer Kaffeemaschine, damit die Heizplatte den Kaffee warmhält ($80\text{-}90^\circ\text{C}$), beim Wasserkocher (100°C), der beim Kochen des Wassers abschalten oder im Bügeleisen, das die ein-

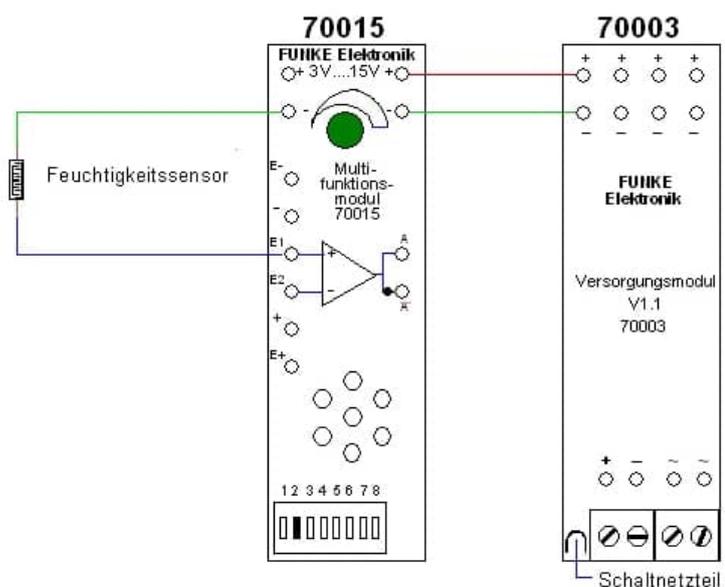


Abb. 10: Feuchtigkeitssensor am Multifunktionsmodul angeschlossen

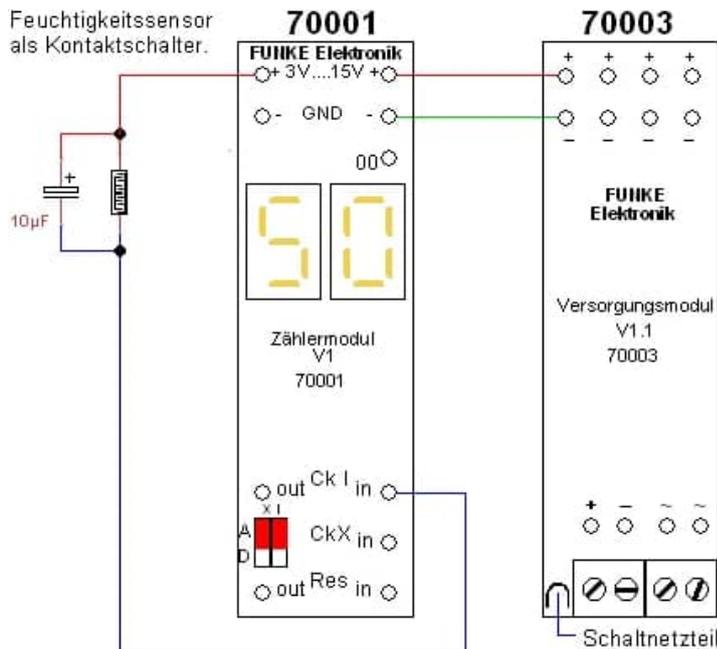


Abb. 11: Feuchtigkeitssensor als Kontaktgeber an einem Zählermodul

gestellte Temperatur halten soll (150-200°C). In diesen Geräten ist ein Temperatursensor (NTC oder PTC) verbaut.

Solche Temperatursensoren gab es auch von fischertechnik, und zwar verschiedene Heißeleiter (NTC, Abb. 2, 3). Bei den neuen Temperatursensoren habe ich einen PTC (Abb. 6) und einen NTC (Abb. 7) in einen Leuchtstein eingebaut.

Die Temperatursensoren sind Widerstände, die in Abhängigkeit von der Temperatur ihren Widerstandswert ändern. Dabei wird ein Kaltleiter bei niedrigeren Temperaturen leitender (kleiner Widerstandswert) und ein Heißeleiter bei höheren (der Widerstandswert sinkt mit steigender Temperatur).

Eine Beschreibung der Funktion und Modelle zum Nachbauen finden sich im Heft vom Baukasten ec2 auf den Seiten 25-28 [1].

Schaltungsbeispiel

Die Temperatursensoren können an das Multifunktionsmodul (70015) angeschlossen werden und so mit dem Drehregler auf die gewünschte Schalttemperatur eingestellt werden – siehe Abb. 12 und 13.

Über den Ausgang A oder \bar{A} des Multifunktionsmoduls kann ein Relaismodul angeschlossen werden, um damit Motoren oder andere Verbraucher anzusteuern. Es können aber auch andere Elektronikmodule an die Ausgänge des Multifunktionsmoduls angeschlossen werden (siehe „Grundlagen der Elektronikmodule“ 70099).

Photosensor

Der Photosensor (70073) ist ein Ersatz für die Fozelle (Fotowiderstand [31361](#) u. ä.) oder den Fototransistor ([36134](#)). Der Photosensor ist eine Fotodiode, die auf Licht mit einer Wellenlänge von 420-1120 nm reagiert und in einem Leuchtstein verbaut ist.

Damit eine optimale Funktion gewährleistet ist, muss die Lichtquelle eine ausreichende Intensität besitzen. LED-Lampen liefern im

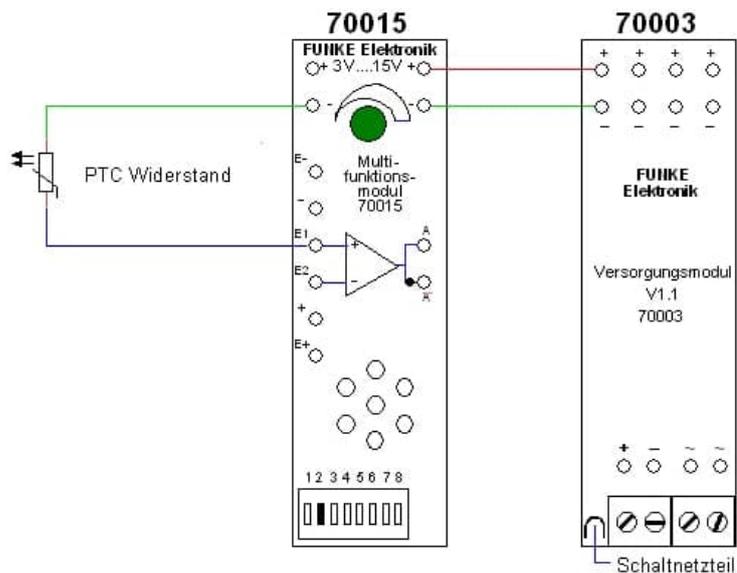


Abb. 12: PTC-Temperatursensor am Multifunktionsmodul

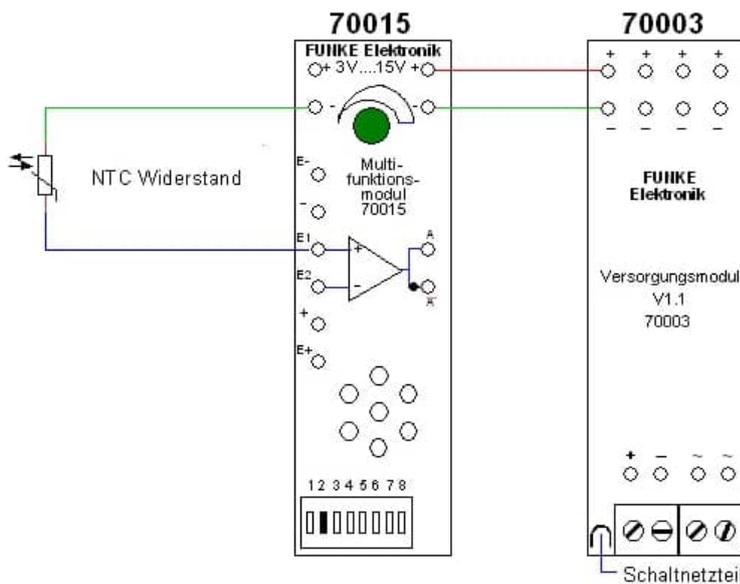


Abb.13: NTC-Temperatursensor am Multifunktionsmodul

allgemeinen zwar Licht im entsprechenden Wellenlängenbereich, haben aber selbst bei maximalem Stromdurchfluss (20-30 mA) nicht genug Lichtintensität, um den Sensor so auszuleuchten, dass dieser ausreichend leitend wird.

Die Fotodiode wird bei Lichteinfall leitend, aber nur in eine Richtung. Aus diesem Grund gibt es auf der Abdeckung des Sensors eine Kennzeichnung für Plus und Minus (Abb. 8).

Schaltungsbeispiel

Wie die Fotodiode an einem Multifunktionsmodul angeschlossen wird, ist in einer Beispielschaltung in Abb. 14 zu sehen. Die Anschaltung kann auch direkt an einem Eingang eines Elektronikmodules betrieben werden – siehe Abb. 15.

Der Photosensor eignet sich z. B. als Lichtschranke. Die Abdeckung auf dem Photosensor (70073) ist so konstruiert, dass auf den Leuchtstein von fischertechnik eine Störlichtkappe (36532, 6 mm) aufgesetzt werden kann, um so die einfallende Lichtmenge zu reduzieren bzw. zu fokussieren. Wichtig ist

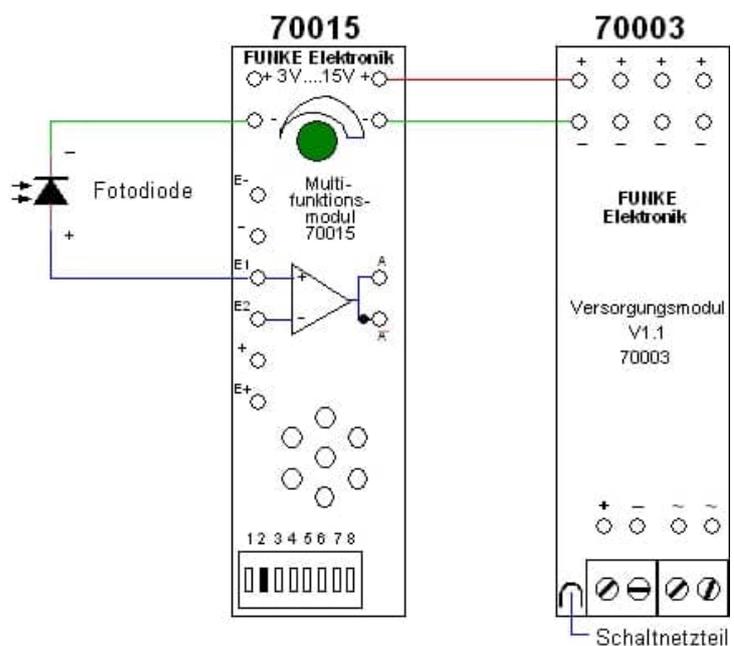


Abb. 14: Photosensor am Multifunktionsmodul

hierbei, dass eine Lichtquelle mit genügend Lichtintensität eingesetzt wird, z. B. eine Glühfadenslampe für 6 V mit 130 mA.

Abb. 17 bis 20 zeigen einen Beispielaufbau für eine Lichtschranke. Im abgebildeten Modell wird eine Lichtschranke durch eine schnell vorbeilaufende Kugel unterbrochen. Dieser Unterbrechungsimpuls wird von einem Zählermodul registriert. Bis zum Stillstand hat die Kugel elfmal die Lichtschranke passiert.

Die Leuchtintensität mancher Lampen ist zu schwach. In diesem Fall kann die Empfindlichkeit am Multifunktionsmodul erhöht werden.

Dafür muss der DIP-Schalter 2 auf OFF (Schalter oben) gestellt und zusätzlich ein Widerstand zwischen „+“ und E1 montiert werden. Der Widerstandswert hängt von der benötigten Sensibilität ab, aber ein guter Mittelwert ist 47 kΩ (Widerstandsset 71047). Eine Beispielschaltung stelle ich anhand eines Modells in der nächsten Ausgabe vor.

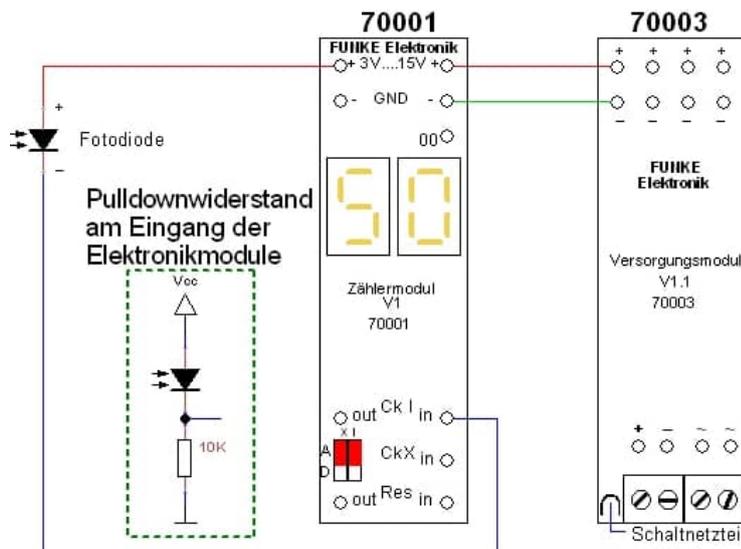


Abb. 15: Der Photosensor direkt angeschaltet an ein Elektronikmodul



Abb. 16: Der Photosensor als Lichtschranke.

Quellen

- [1] Fischerwerke: *ec2*. Anleitung [39171](#) auf [ft-datenbank.de](#).
- [2] [santjohanser](#): Spielen. Lernen. Technik. Autorisierter fischertechnik-Fachhändler.

Die Sensoren werden demnächst bei Franz Santjohanser [2] im Online-Shop erhältlich sein. Bei Fragen und Anregungen zu Schaltungsentwürfen wendet euch gerne an elektronik@funke4you.de.

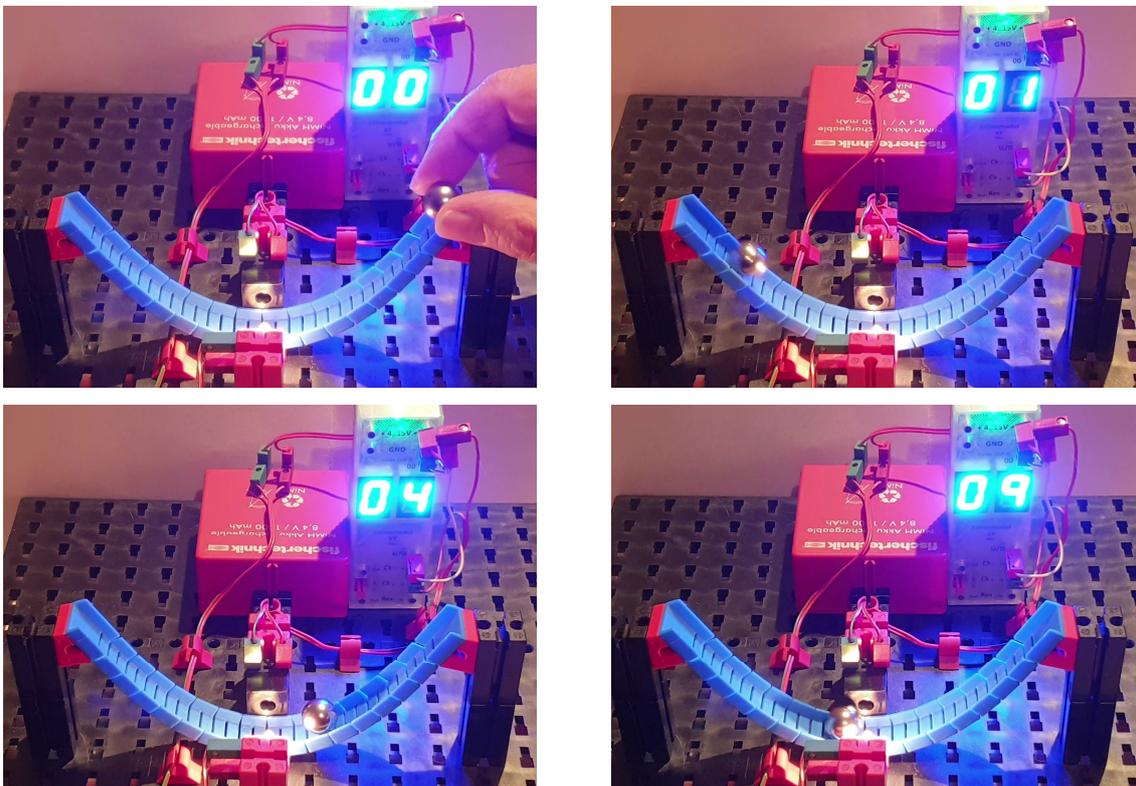


Abb. 17-20: Kugelbahn Simulationsmodell mit dem Photosensor als Lichtschranke

Computing

V. I. P. – Ein I²C-nach-Computing-Interface-Umsetzer (Teil 5)

René Trapp

Nach dem Abspann.

Und hier kommt sie noch, exklusiv für die geduldigen Leser, die auch im Kino immer sitzen bleiben, bis der Nachspann komplett durchgelaufen ist, der Projektor ausgeht und der Reinigungsstrupp die Sessel hochklappt – nur damit man auch wirklich nichts verpasst: die Beschreibung, wie beim V. I. P. die automatische Erkennung des angeschlossenen Interfaces funktioniert. Dabei beziehen wir uns auf den V. I. P. Schaltplan aus Teil 4, Abb. 45.

Die Analogeingänge als Schlüssel

Wie in Teil 3 bereits beschrieben werden die analogen Eingänge EX und EY durch das Ausmessen der Zeitdauer ausgewertet [63]. Im verwendeten Mikrocontroller ist für genau solche Zwecke eine sogenannte Timer-Capture-Einheit enthalten [69]. Diese besteht aus einem Zähler, der mit genau bekannter Taktfrequenz arbeitet, und einem Speicherregister. Wenn eine steigende oder eine fallende Flanke auf einem bestimmten Eingang registriert wird, wird der aktuelle Zählerstand in das Speicherregister kopiert. Die Software kann dann später diesen Zählerstand abholen obwohl der Zähler schon weitergelaufen ist.

Im Ruhezustand liegen beide Leitungen, TRIGGER_X und TRIGGER_Y, auf ‚1‘.

Um die Stellung vom Poti an EX abzufragen, wird die Leitung TRIGGER_X auf ‚0‘ gelegt. Der Zähler startet in diesem Augenblick bei 0. Nach kurzer Zeit wird die

Leitung TRIGGER_X wieder auf ‚1‘ gelegt. Die Software kann sich nun anderen Aufgaben widmen. Läuft die im Interface enthaltene monostabile Kippstufe, die jeweilige Hälfte des 556, ab, so wird dies von der Timer-Capture-Einheit erkannt und der erreichte Zählerstand wird gespeichert. Die Software kann diesen Wert ablesen und weiterverarbeiten.

Allerdings wird die Timer-Capture-Einheit von der aktuellen Software gar nicht verwendet. Weil in der Zwischenzeit sowieso nichts anderes zu tun ist, zählt die Software die Zeitdauern in allen Fällen ganz archaisch selbst per Zählschleife aus. Auf diese Art lässt sich die Zeiteinheit der Zählung besser auf die Verhältnisse am Interface anpassen und es kommen die gewohnten Werte heraus – völlig ohne aufwendige Umrechnungen. Am grundsätzlichen Ablauf ändert sich dadurch aber nichts.

Einige Interfaces bieten anstelle der Impulsdauer Zählimpulse an. Diese werden über einen eigenen Zähler erfasst. Hierzu wird der im Mikrocontroller zweite vorhandene Timer T0 genutzt [69]. In diesem Fall kümmert sich die Software um einen angemessenen Timeout, nachdem die Messung generell beendet wird. In diesem Fall liefert die Abfrage des Analogeingangs dann die Bereichsüberschreitung ‚255‘ zurück.

Speziell für Apple-Interfaces

Eine Extrawurst gibt es für Besitzer eines [30563](#). Hier übernimmt der Apple-Computer selbst die Auswertung der angeschlossenen Zeitglieder. Daher muss auch V. I. P. für dieses Interface die entsprechende Funktion übernehmen.

Man könnte nun den Schaltungsteil mit dem 556 des Interface kopieren. Aber das ist gar nicht nötig. Erfreulicherweise ist im Mikrocontroller bereits ein Komparator enthalten [69]. Das ist eine Schaltung, die zwei analoge Spannungen miteinander vergleicht und am Ausgang eine digitale Auskunft darüber gibt, welche der beiden Spannungen größer ist. Darüber hinaus kann der Ausgang des Komparators im Bedarfsfall noch mit der Zeiterfassung per Timer-Capture gekoppelt werden.

Allerdings gibt es nur einen Komparator, so dass dieser auf die beiden Poti-Anschlüsse aufgeteilt werden muss. Dazu dienen die beiden Dioden D1 und D2 sowie der Widerstand R7. Sie bilden eine Sammelschiene, die zum Anschluss AIN1 des Mikrocontrollers führt.

Im Ruhezustand liegen die Leitungen TRIGGER_X und TRIGGER_Y auf ‚1‘. Die MOSFETs T3 und T4 sind voll durchgesteuert und schließen jeweils die Leitung POTI_X bzw. POTI_Y nach GND kurz. Die Kondensatoren der Zeitglieder im Interface sind entladen. R7 zieht die Sammelschiene zum Komparator auf 0 V, wenn keine der beiden Dioden in Durchlassrichtung gepolt ist.

Um die Stellung des Potis an EX zu erfassen, wird TRIGGER_X auf ‚0‘ gelegt – und bleibt vorerst auf diesem Pegel. Der Kondensator im Interface lädt sich über die 470 Ω und das Poti auf. Da POTI_Y weiterhin auf 0 V liegt, sperrt D2 und über D1 wird die ansteigende Spannung von POTI_X auf die Sammelschiene geleitet. Die Spannung an der Sammelschiene ist dabei etwa 0,5 V niedriger als die Spannung

an POTI_X. Überschreitet die Spannung an der Sammelschiene die durch R8 und R9 vorgegebene Referenzspannung, schaltet der Komparatorausgang im Mikrocontroller um. Die Timer-Capture-Einheit registriert dies und speichert den erreichten Zählerstand. Die Zeitdauer ist damit erfasst und die Leitung TRIGGER_X wird umgehend wieder auf ‚1‘ gelegt. Das Zeitglied im Interface wird entladen. Dabei fließen kurzzeitige Spitzenströme um 1 A. Für die Bedienung von POTI_Y gilt der Ablauf entsprechend.

Die Referenzspannung wird von C2 gepuffert. R10 und R11 sorgen in jedem Fall für ein Erreichen der Komparatorschwelle, auch ohne eine Verbindung zwischen EX bzw. EY und COM. Beim [30563](#) könnte der Ladevorgang allerdings auch jederzeit per Software abgebrochen werden.

Interface-Automatik

Je nach Interface ist es nun also sehr unterschiedlich, wie genau das Abflauen einer monostabilen Kippstufe erkannt wird. Und je nach Interface ist der Anschluss an V. I. P. unterschiedlich gelöst.

Exakt diese Unterschiede ermöglichen die automatische Erkennung des angeschlossenen Interfaces. Die Idee ist relativ simpel: Jedes Interface hat seine spezielle Antwort für V. I. P. auf die fallende Flanke an TRIGGER_X oder TRIGGER_Y.

Abb. 50 zeigt die Situation an den Leitungen zum Interface [30563](#). Nach der fallenden Flanke an TRIGGER_X oder TRIGGER_Y beginnt die Spannung an AIN1 gemächlich zu steigen. Selbst nach einer bestimmten Zeitdauer t hat die Spannung an AIN1 die Komparatorschwelle noch nicht erreicht. Alle anderen Leitungen halten ihren Zustand unverändert.

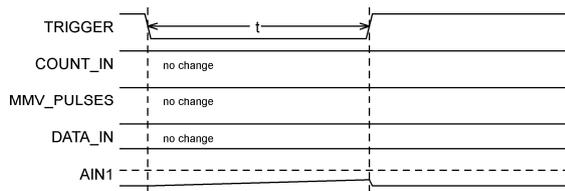


Abb. 50: Triggerantwort [30563](#)

Legt man die Leitung TRIGGER_X oder TRIGGER_Y wieder auf ‚1‘, so wird der Kondensator sofort entladen. Eine Eigenschaft, die dem Original leider fehlt.

Abb. 51 gilt für ein [30561](#). Hier reagiert die Leitung COUNT_IN umgehend mit einer steigenden Flanke auf den TRIGGER. Die Spannung an AIN1 folgt der Spannung an TRIGGER invertiert und mit einer kurzen Verzögerung. Nach einer bestimmten Zeitdauer t ist AIN sicher bereits oberhalb der Komparatorschwelle. Die anderen Leitungen halten ihre Pegel unverändert.

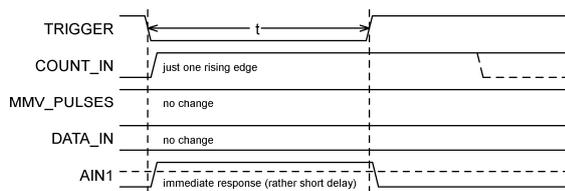


Abb. 51: Triggerantwort [30561](#)

Im Falle eines [30562](#) oder [30564](#) löst die fallende Flanke an TRIGGER den Impuls-generator im Interface aus (Abb. 52). Während einer bestimmten Zeitdauer t treffen mehrere Impulse ein und werden von T0 gezählt. Die Spannung an AIN1 folgt der Spannung an TRIGGER invertiert und mit einer kurzen Verzögerung. Nach einer bestimmten Zeitdauer t ist AIN sicher bereits oberhalb der Komparatorschwelle. Die anderen Leitungen halten ihre Pegel unverändert.

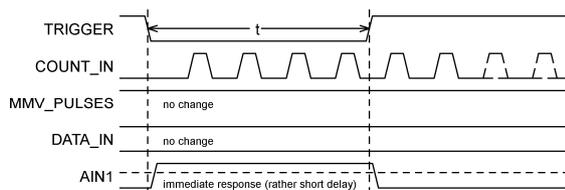


Abb. 52: Triggerantwort [30562](#), [30564](#)

Abb. 53 gilt für ein [30565](#) oder [30567](#). Hier reagiert die Leitung DATA_IN umgehend mit einer fallenden Flanke auf den TRIGGER. Die Spannung an AIN1 folgt der Spannung an TRIGGER invertiert und mit einer kurzen Verzögerung. Nach einer bestimmten Zeitdauer t ist AIN sicher bereits oberhalb der Komparatorschwelle. Die anderen Leitungen halten ihre Pegel unverändert.

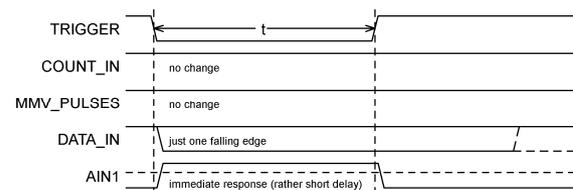


Abb. 53: Triggerantwort [30565](#), [30567](#)

Bei den universellen Interfaces [30566](#), [39319](#), [30510](#) oder [66843](#) ist die Antwort nicht anders als bisher auch, können doch diese Interfaces alle genannten Typen ersetzen. Die Verschaltung an V.I.P. jedoch sorgt für eine einzigartige Rückmeldung von diesen Interfaces, Abb. 54. Die fallende Flanke an TRIGGER löst den Impuls-generator im Interface aus, wie bei [30562](#) oder [30564](#) auch. Während einer bestimmten Zeitdauer t treffen so mehrere Impulse ein und werden von T0 gezählt.

Zusätzlich reagiert die Leitung MMV_PULSES umgehend mit einer steigenden Flanke auf den TRIGGER. Die Spannung an AIN1 folgt der Spannung an TRIGGER invertiert und mit einer kurzen Verzögerung. Nach einer bestimmten Zeitdauer t ist AIN sicher bereits oberhalb der Komparatorschwelle. Die Leitung DATA_IN hält ihren Pegel unverändert.

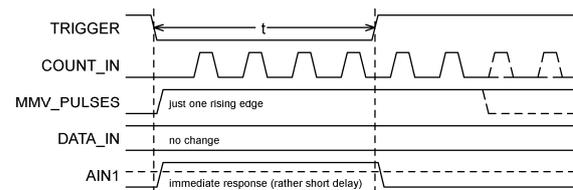


Abb. 54: Triggerantwort [30566](#), [39319](#), [30520](#), [66843](#)

Aus diesen Rückantworten lässt sich unmittelbar ablesen, welches Interfaces angeschlossen ist.

Liegt die Leitung MMV_PULSES im Ruhezustand auf ‚0‘, ist es ein [30566](#), [39319](#), [30520](#) oder [66843](#). Diese Leitung ist nur bei diesen Interfaces angeschlossen und liegt sonst durch Maßnahmen im Mikrocontroller immer auf ‚1‘. Zusätzlich gibt es nur hier innerhalb einer Wartezeit von ca. 50 µs nach TRIGGER die umgehende steigende Flanke auf MMV_PULSES sowie eine bestimmte Anzahl Taktimpulse auf COUNT_IN. Timer T0 zählt die Taktimpulse auf COUNT_IN.

Löst TRIGGER innerhalb ca. 50 µs mehrere Zählimpulse auf COUNT_IN aus und lag die Leitung MMV_PULSES bereits im Ruhezustand auf ‚1‘, so ist es ein [30562](#) oder [30564](#). Timer T0 zählt die Taktimpulse auf COUNT_IN.

Liegt nach einer Wartezeit von ca. 50 µs AIN1 noch unterhalb der Komparatorschwelle, dann ist es ein [30563](#). Die Software ermittelt per Zählschleife die Zeitdauer vom Analog-Komparator.

Reagiert mit dem TRIGGER nur die Leitung COUNT_IN innerhalb ca. 50 µs mit einer einzigen steigenden Flanke, so ist es ein [30561](#). In diesem Fall muss die Impulsdauer per Software erfasst werden.

Reagiert die Leitung DATA_IN mit einer fallenden Flanke auf den Impuls an TRIGGER, so ist es ein [30565](#) oder [30567](#). Die Software ermittelt per Zählschleife die Zeitdauer, diesmal am Anschluss PD6.

Auf diese Art ist das Interface identifiziert und alle weiteren Eigenschaften bezüglich Motorendstufen und eventuelle Invertierung von DATA_IN sind ebenfalls bekannt. Das Programm im Mikrocontroller kann sich nun selbst auf die richtige Ansteuerung einstellen.

Interface	Poti Direktanschluss	Poti Impulsdauer an DATA_IN	Poti Impulsdauer an MMV_PULSES	Poti Impulsdauer an COUNT_IN	Poti Impulsanzahl an COUNT_IN	Motorendstufen Diskret	Motorendstufen Integriert	DATA-IN Nichtinvertiert	DATA-IN Invertiert
30561				✓		✓		✓	
30563	✓					✓		✓	
30562					✓	✓		✓	
30564					✓	✓		✓	
30565		✓				✓			✓
30567		✓				✓			✓
30566			✓		✓		✓	✓	
39319			✓		✓		✓	✓	
30520			✓		✓		✓	✓	
66843			✓		✓		✓	✓	

Tab. 1: Wichtige Eigenschaften der Interfaces aus V. I. P.-Sicht

Wird keine dieser erwarteten Reaktionen erkannt, ist die Auswertung der Analogeingänge nicht möglich. Die Leitung DATA_IN wird als nichtinvertiert angenommen. Sicherheitshalber wird von diskreten Motorendstufen ausgegangen.

Tab. 1 fasst die Eigenschaften der Interfaces noch einmal übersichtlich zusammen.

Nicht Stören

In Teil 4 [72] habe ich erklärt, wie das V. I. P. auch an das computerseitige Kabelende angeschlossen werden kann. Dabei ist die Störsicherheit der Potentiometeranschlüsse beeinträchtigt. Insbesondere beim [66843](#) (seinen engen Verwandten) und auch beim [30563](#) ist zu beachten, dass die Signaldern im Flachkabel neben schnellen digitalen Signalen angeordnet sind. Dazu kommt die Anschlusslänge von rund 1,5m; beim [66843](#) laufen die Leitungen je Poti zweimal durch das Kabel und weisen so jede für sich insgesamt 3m Strecke auf.

Beides zusammen begünstigt erheblich die Einkopplung von Störsignalen. Diese Störsignale können die Zeitmessung deutlich verfälschen und zu fehlerhaften analogen Werten führen. Die effektivste Maßnahme ist das Kürzen des Kabels auf ein Minimum oder der Einbau des V.I.P. direkt ins Interface – sofern nicht komplett auf die analoge Auswertung verzichtet wird.

Nachklapp

Ein Punkt ist in der Betrachtung noch untergegangen. In Teil 2 [54] gibt es die Betrachtung zum maximalen Ausgangsstrom der Motortreiber zusammen mit der Bemerkung, dass für die damals erhältlichen fischertechnik-Motoren ausreichend Leistung bereitsteht. Nun schreiben wir aber nicht das Jahr 1985 sondern leben im Jahr 2017. Die besagten alten Motoren mit 6V Nennspannung sind nur noch antiquarisch erhältlich. Und so steht die Frage im Raum, welche aktuellen fischertechnik-Motore mit 9V Nennspannung denn ersatzweise angeschlossen werden können.

Zuerst einmal: Die modernen Motore arbeiten auch mit 6V Betriebsspannung sehr gut, haben allerdings eine geringere Drehzahl und ein geringeres Drehmoment gegenüber dem Betrieb an 9V. Geeignet erscheinen:

- XS-Motor ([137096](#))
- S-Motor ([32293](#), 32293L, 32293R)
- ‚Traktor‘-Motor ([151178](#))
- Encoder-Motor alt ([135484](#))
- Quadratur-Encoder-Motor ([75150](#))

Ferner noch Magnetartikel:

- Pneumatik-Kompressor ([121470](#))
- Magnetventil ([35327](#))
- Relais ([35793](#))
- Elektromagnet ([31324](#))
- Elektromagnet ([142504](#))

... und natürlich die diversen Lampen.

Vom XM-Motor und den Power-Motoren sollte man ausdrücklich die Finger lassen!

Hinter dem Horizont

Im Zuge der Recherchen über die alten Interfaces sind dem Autor noch einige Internet-Seiten ohne direkten Bezug zum Inhalt der Artikelreihe aufgefallen, die Euch aber aufgrund ihres Bezuges zu Computing mit den alten Interfaces nicht vorenthalten werden sollen. Deswegen sind sie am Schluss der Quellenliste zusätzlich aufgeführt – ausnahmsweise.

Quellen

- [1] Dirk Fox: *I²C mit TX und Robo Pro – Teil 1: Grundlagen*. [ft:pedia 3/2012](#), S. 32
- [2] Dirk Fox: *I²C mit dem TX(T) – Teil 12: Temperatursensor*. [ft:pedia 4/2015](#), S. 44
- [3] Jens Lemkamp: *Parallel-Interface durch Arduino gesteuert (1)*. [ft:pedia 1/2014](#), S. 24

- [4] Dirk Uffmann: *Nutzung des Universal-Interfaces 30520 als Port-Erweiterung an einem Mikrocontroller*. [ft:pedia 2/2014](#), S. 30
- [5] Dirk Uffmann: [Altes FT Universal-Interface \(Parallelschnittstelle\) gesteuert durch AVR Mikrocontroller](#).
- [6] Thomas Kaiser: [C-64 Interface an RoboPro](#).
- [7] Forumsdiskussion: [I2C clock stretching am TXT](#).
- [8] Forumsanfrage an fischertechnik: [Unterstützung von i2c clock stretching beim TXT controller?](#).
- [9] Erneute Forumsanfrage an fischertechnik: [Unterstützung von i2c clock stretching beim TXT controller?](#)
- [10] Forumsdiskussion: [TXT und clock-stretching](#).
- [11] J. P. M. Steeman: *Robotik mit dem Homecomputer*. Elektor, 1987
- [12] Klaus Merkert: [Zwei Schaltpläne](#).
- [13] Holger Howey: [Kombi-Schaltplan](#).
- [14] Klaus Merkert: [Umbau Apple II-Interface auf PC](#).
- [15] Ulrich Müller: [Übersicht der Interfaces](#).
- [16] fischertechnik-Datenbank: [30561 \(Interface CBM\)](#).
- [17] fischertechnik-Datenbank: [30562 \(Interface Commodore\)](#).
- [18] fischertechnik-Datenbank: [30563 \(Interface Apple\)](#).
- [19] fischertechnik-Datenbank: [30564 \(Interface Acorn\)](#).
- [20] fischertechnik-Datenbank: [30565 \(Interface Schneider\)](#).
- [21] fischertechnik-Datenbank: [30567 \(Interface IBM\)](#).
- [22] fischertechnik-Datenbank: [30566 \(Interface Centronics\)](#).
- [23] fischertechnik-Datenbank: [39319 \(Centronics Schul-Interface CVK\)](#).
- [24] fischertechnik-Datenbank: [66843 \(Universal Schul-Interface CVK\)](#).
- [25] fischertechnik-Datenbank: [30520 \(Interface Universal\)](#).
- [26] Fischertechnikclub Nederland: *Twee robotarmen om je fantasie te prikkelen*. [Clubblad 2/1992](#), S. 6
- [27] Gerhard Bader: *Fischer-Technik und Computer: Programme für Atari ST, Commodore 64/128, Schneider CPC, IBM PC u. Kompatible*. [CHIP-SPECIAL 1987](#), Vogel Verlag, 1987
- [28] CPC Wiki: [Foto eines 30562](#).
- [29] CPC Wiki: [Foto eines 30567](#).
- [30] C. Hehr: [Platine eines 30566](#). fischertechnik community.
- [31] Fischertechnikclub Nederland: [Computing Dozen](#).
- [32] fischertechnik: [CVK-fischertechnik Schul-Interface \(67319\)](#).
- [33] C64 Wiki: [C64 Userport](#).
- [34] A2wiki: [Apple II Game-IO](#).
- [35] Chris Whytehead: [Acorn Pinouts](#).
- [36] Larry Davis: [PC Parallel Port Pin-Out](#).
- [37] CPC Wiki: [CPC Printerport](#).
- [38] fischertechnik-Datenbank: [32680 \(Stecker-Netzgerät 220V:6,8V 1,5A Computing Experimental\)](#).
- [39] alldatasheet: [Datenblatt 1N4001 \(NXP\)](#).
- [40] Texas Instruments Inc.: [Datenblatt 7805](#).
- [41] René Trapp: *V. I. P. – Teil I*. [ft:pedia 2/2017](#), S. 63

- [42] alldatasheet: [Datenblatt BD135 \(NXP\)](#).
- [43] alldatasheet: [Datenblatt BD136 \(NXP\)](#).
- [44] alldatasheet: [Datenblatt BC547 \(NXP\)](#).
- [45] alldatasheet: [Datenblatt TLE4201 \(Siemens\)](#).
- [46] Stefan Falk: *Motorsteuerungen (Teil 1)*. [ft:pedia 1/2011](#), S. 4
- [47] Dirk Uffmann: *PWM-Motorsteuerung am fischertechnik-Universal-Interface*. [ft:pedia 4/2015](#), S. 30
- [48] wikipedia: [Induktion](#).
- [49] wikipedia: [Freilaufdiode](#).
- [50] alldatasheet: [Datenblatt 1N4148 \(Vishay\)](#).
- [51] alldatasheet: [Datenblatt 4011 \(NXP\)](#).
- [52] alldatasheet: [Datenblatt 4081 \(NXP\)](#).
- [53] alldatasheet: [Datenblatt 4094 \(NXP\)](#).
- [54] René Trapp: *V. I. P. – Teil 2*. [ft:pedia 3/2017](#), S. 57
- [55] alldatasheet: [Datenblatt 4014 \(NXP\)](#).
- [56] fischertechnik-Datenbank: [39485 \(Interface Commodore Computer\)](#).
- [57] U. Tietze, C. Schenk: *Halbleiter-Schaltungstechnik*. Springer, 1993, 10. Auflage, S. 215
- [58] alldatasheet: [Datenblatt 4050 \(NXP\)](#).
- [59] alldatasheet: [Datenblatt 4071 \(NXP\)](#).
- [60] alldatasheet: [Datenblatt 556 \(ON\)](#).
- [61] H. Camenzind: [Designing Analog Chips](#).
- [62] Wikipedia: [IEEE_1284](#).
- [63] René Trapp: *V. I. P. – Teil 3*. [ft:pedia 4/2017](#), S. 36
- [64] Forumdiskussion: [I²C-Expander](#).
- [65] Dirk Fox: *I²C mit dem TX(T) – Teil 13: Farbsensor*. [ft:pedia 1/2016](#), S. 79
- [66] NXP Semiconductors: [UM10204 I²C-bus specification and user manual, Rev. 6](#).
- [67] wikipedia: [Bipolartransistor, Inversbetrieb](#).
- [68] Analog Innovation: [Inverted Transistors \(Hunter\)](#).
- [69] Atmel Corporation: [Datasheet ATtiny2313](#).
- [70] NXP Semiconductors: [Datenblatt PCF8574\(A\)](#).
- [71] [V.I.P.](#) bei github.com
- [72] René Trapp: *V. I. P. – Ein I²C-nach-Computing-Interface-Umsetzer (Teil 4)*. [ft:pedia 1/2023](#), S. 111–122.
- Blockschaltbild und Anschlüsse für IBM:**
- [73] Ulrich Müller: [Internal construction and programming of the parallel interfaces](#).
- Bild vom 30562:**
- [74] C64 Wiki: [Fischertechnik Computing Modell 30562](#).
- Bild vom 30566 (eher 30567):**
- [75] Atari Wiki: [Fischer Technik Computing](#).
- Interface 30567:**
- [76] CPC Wiki: [Fischertechnik Interface](#).
- Umbauanleitung 30520 am C64:**
- [77] Sascha Bader: [Fischertechnik Computing PC Interface am C64](#).
- [78] Ulrich Keller: [fischertechnik-computing-Ecke](#).

Computing

Die Beispielprogramme von ROBO Pro Coding für den TXT 4.0 im Überblick

Axel Chobe

Wer sich erstmalig mit dem neuen Controller beschäftigt, vermisst eine zusammenhängende Bedienungsanleitung. Was es aber gibt, sind zahlreiche Beispielprogramme, einige PDF- und DOCX-Dateien zur Erklärung. Der Beitrag gibt einen Überblick über deren Zusammenhang.

Die Beispielprogramme werden über das Menü (1), Neu (2), Beispiele (3), NÄCHSTE (4) aufgerufen. Nach der Auswahl des Baukastens (5) oder der Grundlagen des TXT 4.0 können die Beispielprogramme geladen werden.



Abb. 1: Hauptmenü

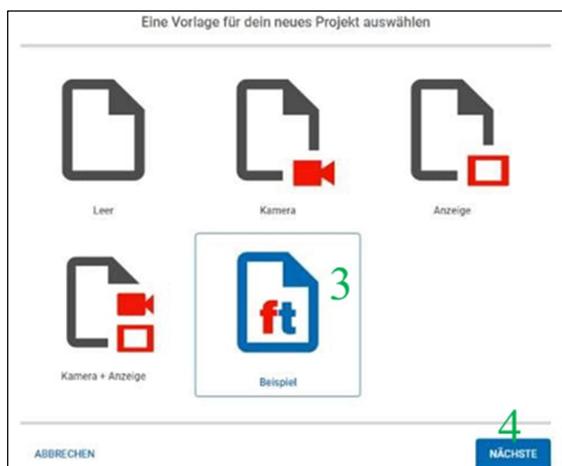


Abb. 2: Auswahl der Beispiele



Abb. 3: Auswahl des Baukastens

Inzwischen gibt es außer dem Grundbaukasten *Robotics Hightech* auch den Baukasten *Robotics TXT 4.0 Base Set* und mehrere darauf aufbauende Erweiterungskästen (Add-Ons, siehe Abb. 4-7) aus der fischertechnik-Education-Reihe, zu denen ebenfalls Beispielprogramme bereitgestellt werden.

Außerdem gibt es PDF-Dateien zum Download [1], in denen die Möglichkeiten des *Robotics Hightech*-Kastens näher erläutert werden.



Abb. 4: Robotics Add On: Autonomous Driving. Zusatzset für autonomes Fahren zum Robotics TXT 4.0 Base Set



Abb. 7: Robotics Add On: Competition. Zusatzset für Wettbewerbe zum Robotics TXT 4.0 Base Set



Abb. 5: Robotics Add On: Omniwheels. Zusatzset für Omniwheels zum Robotics TXT 4.0 Base Set

Des Weiteren können Beispielprogramme für den Grundkasten ROBOTICS Base Set geladen werden. Dazu gibt es passend jeweils eine Beschreibung als Word-Datei. Diese können unter [2] geladen werden.



Abb. 8: STEM Robotics TXT 4.0 Base Set



Abb. 6: Robotics Add On: IoT. Zusatzset für IoT zum Robotics TXT 4.0 Base Set

Zu guter Letzt sind auch die Programme der Trainingsmodelle ladbar. Dazu gibt es derzeit aber keine weiteren Erläuterungen.

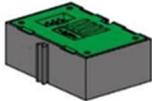
1. TXT 4.0 Controller



Abb. 9: TXT 4.0

In den mit **Phyton** markierten Dateien sind Phyton-Code-Elemente eingesetzt, die noch nicht von fishertechnik dokumentiert wurden.

- **test_environmental_sensor**
Test für den Umweltsensor



- **test_combi_sensor**
Test für den Kombisensor



- **test_gesture_sensor**
Test für den Gestensensor



- **test_image_processing**
Test für die Bildverarbeitung **Phyton**



- **voltage_extended_range_I1**
Test für erweiterten Spannungsbereich
Phyton



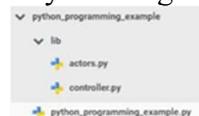
- **test_voice**
Test für Sprachsteuerung



- **python_programming_template**
Phyton-Programmervorlage



- **python_programming_example**
Phyton-Programmierbeispiel



- **test_battery**
Test für den Akku **Phyton**



- **test_speedtest**
Schnelligkeitstest **Phyton**



2. Baukasten Robotics Hightech



Abb. 10: Robotics Hightech

Funktionsmodell.pdf

Abb. 11: Testaufbau

- **function_motor_on**
Funktion Motor ein
- **function_lightbarrier_motor_on**
Funktion Lichtschranke Motor ein
- **function_ultrasonic_motor_on**
Funktion Ultraschall Motor ein
- **function_ultrasonic_motor_speed**
Funktion Ultraschall Motordrehzahl
- **function_encodermotor_distance**
Funktion Encodermotor Entfernung

Kameramann.pdf

Abb. 12: Testaufbau Kameramann

- **cameraman_detect_color**
Es sollen drei verschiedene Farben (rot, gelb, grün) erkannt werden (Farbtafeln). Die Farbe, die erkannt wurde, soll im Display der TXT 4.0 Controllers angezeigt werden.
- **cameraman_alarm**
Bewegung im Kamerabild soll erkannt werden. Ist dies der Fall, wird ein akustisches und ein optisches Signal ausgegeben.

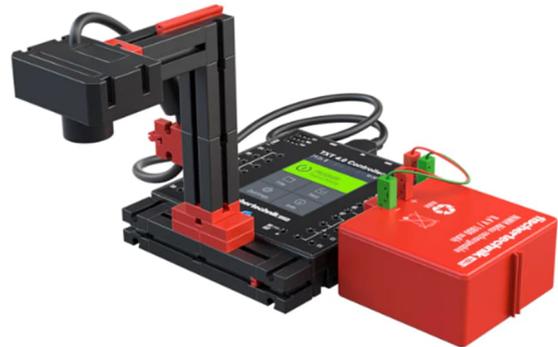
Barcodeleser.pdf

Abb. 13: Testaufbau Barcode-Leser

- **barcodereader_code39**
Über die Kamera werden 5 Linien nach dem Prinzip „Barcode 39“ erkannt und in Buchstaben oder Zahlen auf dem Display des TXT 4.0 Controllers angezeigt.

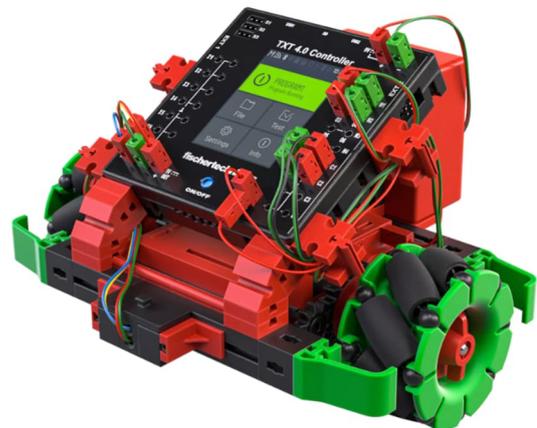
OmniwheelX2.pdf

Abb. 14: Fahrzeug 1 mit Omniwheels

- **omniwheels_x2_forward_backward_time**
Das Fahrzeug soll eine bestimmte Zeit geradeaus nach vorne und wieder zurückfahren. Anschließend soll das Programm stoppen.
- **omniwheels_x2_forward_backward_dist**
Erweitert das Programm um eine impuls-gesteuerte Rückfahrt.
- **omniwheels_x2_forward_rot90_dist**
Der Ablauf erhält zusätzlich eine 90 Grad Drehung.

- **omniwheels_x2_square**
In dieser Aufgabenstellung soll das Fahrzeug ein Rechteck abfahren.
- **omniwheels_x2_linefollower**
Fahrt auf der Vorlage als Linienverfolger

OmniwheelX4.pdf



Abb. 15: Fahrzeug 2 mit Omniwheels

- **omniwheels_x4_move**
Bewegungstest
- **omniwheels_x4_forward_backward_time**
Das Fahrzeug soll eine bestimmte Zeit geradeaus nach vorne und wieder zurückfahren. Anschließend soll das Programm stoppen.
- **omniwheels_x4_forward_backward_dist**
Das Fahrzeug soll über eine bestimmte Distanz (impulsgesteuert) bewegt werden.
- **omniwheels_x4_forward_rot90_dist**
Der Ablauf erhält zusätzlich eine 90 Grad Drehung.
- **omniwheels_x4_forward_diagonalright_forward**
Diagonalfahrt
- **omniwheels_x4_forward_sideways_right/left**
Fahrt vorwärts, seitwärts rechts / links

OmniwheelX4sensoren.pdf



Abb. 16: Omniwheels-Fahrzeug mit Sensoren

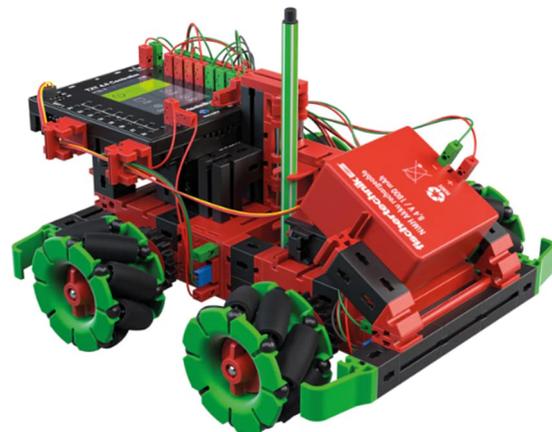
- **omniwheels_x4_sensor_linefollower**
Einer Spur folgen mit Hilfe des IR-Spursensors.
- **omniwheels_x4_sensor_inefeller_ultrasonic**
Eine Spur suchen, der Spur folgen und Hindernisse mit dem Ultraschallsensor erkennen.
- **omniwheels_x4_sensor_linefollower_ultrasonic_acc**
Die Entfernung zum Hindernis wird gemessen und die Geschwindigkeit dem Abstand entsprechend angepasst.
- **omniwheels_x4_sensor_servomove**
Testprogramm für die Kamera in Verbindung mit dem Servomotor.
- **omniwheels_x4_sensor_linefollower_camera_trafficsign**
Farberkennung mit der Kamera, Anzeige der Farbflächen auf dem Display, seitliches Einparken, warten am Zebrastrifen.
- **omniwheels_x4_sensor_inefeller_camera_pd**
Spurerkennung mit Kamera und PD-Regler. Modell soll an der schwarzen Line des Parcours entlangfahren.
- **omniwheels_x4_cam_move**
Kamerabewegung

Omniwheel_x4_ballroboter.pdf*Abb. 17: Ballwurf-Roboter*

- **omniwheels_x4_ball_simple**
Ball laden und über den Servo feuern
- **omniwheels_x4_ball_gui**
Seitlich fahren, drehen, Ball laden und abschießen unter Verwendung von Buttons auf dem Display
- **omniwheels_x4_target_detect**
Ziele (rote, gelbe oder grüne Tafeln) erkennen, Roboter am Ziel ausrichten. Die Auswahl der Farbe wird über die Buttons eingegeben. Roboter sucht das ausgewählte Ziel.
- **omniwheels_x4_target_balldetect**
Rote, gelbe und grüne Tafeln werden erkannt und abgeschossen – alle drei nacheinander
- **omniwheels_x4_ball_voice_control**
Fernsteuerprogramm mit Spracheingabe – Sprachsteuerung der Fahrbefehle sowie des Abschusses

Fussballroboter.pdf*Abb. 18: Fußballroboter*

- **omniwheels_x4_soccer_ballfollower**
Das Fußballfeld wird ohne Tor aufgebaut. Der Roboter soll sich im Kreis drehen und einen Ball suchen. Liegt der Ball im Schussbereich der Lichtschranke, wird er weggeschossen.
- **omniwheels_x4_soccer_goal**
Auf dem Fußballfeld wird ein Tor aufgebaut. Dieses und der Ball wird vom Roboter mit der Kamera erfasst und der Ball wird ins Tor geschossen.

Malroboter.pdf*Abb. 19: Malroboter*

- **omniwheels_x4_paint**
Über das Display sowie mit der Sprachsteuerung können verschiedene Figuren gezeichnet werden.

Hinweis: Die Dateien sind nicht in dieser Reihenfolge aufrufbar.

3. Autonomous Driving



Abb. 20: Zusatzset für autonomes Fahren zum Robotics TXT 4.0 Base Set

Im Folgenden werden die Programme der Grundlagen- und Erweiterungsbausätzen aufgelistet.

- **Phyton**: Programme mit diesem Symbol arbeiten auch mit Python-Code.
- **Cloud**: Programme mit diesem Symbol benötigen die Einrichtung einer ft-Cloud.
- **(A1/A1)**: Programme mit dieser Erweiterung sind in einer zusätzlichen Datei beschrieben. Diese können unter [2] als ZIP-Datei heruntergeladen werden. Dabei bedeuten (A1... den Namen der Datei, .../A1) die Nummer der Aufgabe bzw. .../E1) die Nummer des Experiments).
- **Autonomous Parking**
Automatisches Einparken (A4/E1)
- **Display_Counter_Encoder**
Impulzzähler Encodermotor (A1/A1)
- **Distance_Test**
Distanzzähler Encodermotor (A1/A2)
- **Odometer**
Kilometerzähler (A1/A2c)
- **Tachometer_and_Odometer**
Tachometer und Kilometerzähler (A1/A3)
- **Tachometer_and_Odometer_Thread**
Tachometer und Kilometerzähler im Thread ausführen (A1/E1)
- **Taxi_Meter**
Taxameter (A1/E2)
- **Brake_Assist**
Bremsassistent (A2/A3)
- **Emergency_Stop**
Not-Halt (A2/A1)
- **Lane_Keeping_Assist_P_Control**
Spurhalteassistent (A2/E1)
- **Lane_Keeping_Assist_PD_Control**
Spurhalteassistent (A2/E2)
- **Servo_Calibration**
Geradeausfahrt (A2/A2)
- **Speed_Control**
Tempomat (A2/A4)
- **Automatic_Lighting**
Automatische Lichtfunktionen in Threads
- **Blinker**
Blinkertest (A3/A3)
- **Blinker_with_Warning_Lights**
Blinkertest mit Warnlicht (A3/E1)
- **Headlight**
Abblendlicht (A3/E2)
- **Reversing_Light**
Rückfahrlicht (A3/A1)
- **Stoplight**
Bremslicht (A3/A2)
- **Stoplight_hangover**
Bremslicht mit Nachleuchten (A3/A2b)
- **Stoplight_proportional**
Bremslicht mit proportionaler Helligkeit (A3/A2c)
- **Find_Parking_Space**
Parklücke finden (A4/A1)
- **Parking_Manoeuvre**
Einparkmanöver (A4/A2)
- **Tachometer**
Tachometer (A1/A3)

4. IoT



Abb. 21: Zusatzset für IoT zum Robotics TTS 4.0 Base Set (Smart Home)

- **Sensorstation**
Sensordaten an Cloud übermitteln
- **IoT_Barometer**
Anzeige Luftdruck, Luftfeuchtigkeit, Temperatur (A1/A2)
- **IoT_Barometer_with_NTC_resistor**
Anzeige Luftdruck, Luftfeuchtigkeit, Temperatur mit Sensor und NTC (A1/E1)
- **IoT_Metrological_Station**
Metrologische Station (A1/A1)
- **IoT_MQTT**
Metrologische Station über MQTT (Temperatur, Luftfeuchte und Druck) [Phyton Cloud](#)
- **IoT_MQTT_Barometer**
Metrologische Station über MQTT (Temperatur, Luftfeuchte und Druck) [Phyton Cloud](#) (A1/E2)
- **IoT_MQTT_Barometer_with_Webcam**
Metrologische Station über MQTT inkl. Webcam [Phyton Cloud](#) (A1/E3)
- **IoT_Webcam**
Bild mit Kamera aufnehmen [Phyton](#)
- **IoT_CO2_Signal_Light**
Luftqualitätsmessung (CO₂-Ampel) (A2/E2)
- **IoT_Illuminance**
Anzeige Beleuchtungsstärke (A2/E1)
- **IoT_Indoor_Air_Quality**
Anzeige Luftqualität (A2/A2)
- **IoT_Indoor_Climate**
Anzeige Temperatur und Luftfeuchtigkeit (A2/A1)
- **IoT_Loudness**
Anzeige der Lautstärke auf GUI und Kontrolllampen (A2/E3)
- **IoT_MQTT_Indoor_Air_Quality**
Anzeige Luftqualität [Phyton Cloud](#)
- **IoT_Camera_Motion_Control**
Grundstellung Kamera
- **IoT_Init_Camera_A**
Grundstellung Variante A (A3/A1c)
- **IoT_Init_Camera_B**
Grundstellung Variante B (A3/A1c)
- **IoT_Init_Camera_Pan**
Grundstellung Variante C (A3/A1c)
- **IoT_Init_Camera_Tilt**
Grundstellung Kamera (A3/A1)
- **IoT_Microphone_Level**
Geräuschaktivierung (A3/A3)
- **IoT_Surveillance_Camera**
Kameraüberwachung [Phyton Cloud](#) (A3/A2)
- **IoT_Surveillance_Camera_Dashboard_Control**
Steuerung über das Cloud-Dashboard [Phyton Cloud](#) (A3/E2)
- **IoT_Surveillance_Camera_Dashboard_Control_Test**
Kameraüberwachung mit Fernsteuerung Test [Phyton Cloud](#)
- **IoT_Surveillance_Camera_Motion_Detection**
Bewegungserkennung [Phyton Cloud](#) (A3/A4)
- **IoT_Surveillance_Camera_Noise_Detection**
Geräuscherkennung mit Schwellenwert [Phyton Cloud](#) (A3/A3)
- **IoT_Surveillance_Camera_Voice_Control**
Sprachsteuerung [Phyton Cloud](#) (A3/A1)
- **IoT_Test_Dashboard_Control**
Test zwischen IoT und Dashboard [Phyton Cloud](#)

5. Omniwheels



Abb. 22: Zusatzset für Omniwheels zum Robotics TXT 4.0 Base Set

- **Mecanum_Line_Follower_P_Controller**
Spurfolger mit Proportionalregler (A2/E1)
- **Mecanum_Coordinates_Drawing**
Malen nach Zahlen (A3/E2)
- **Mecanum_Drawing_Coordinates**
Zeichnungskordinaten
- **Mecanum_Drawing_Polygons**
Zeichnen eines N-Ecks (A3/A3)
- **Mecanum_House_of_Santa_Claus**
Zeichnen Haus vom Nikolaus (A3/A2)
- **Mecanum_Move_2_Point**
Zielpunkt ansteuern (A3/E1)
- **Mecanum_Correct_Distance**
Zielabstandskorrektur (A4/E2)
- **Mecanum_Find_Target**
Zielsuche (A4/E4)
- **Mecanum_Fire_and_Load**
Auswurfmechanismus (A4/A1)
- **Mecanum_Fire_and_Load_extended**
Auswurfmechanismus mit Füllstandsanzeige (A4/A1)
- **Mecanum_Fire_and_Load_Voice_Control**
Sprachsteuerung (A4/A2)
- **Mecanum_Turn2Target**
Zielausrichtung (A4/E3)
- **Test_Target_Distance**
Abstandsmessung (A4/E1b)
- **Mecanum_Boundary_Line**
Spurfolger (Linienerkennung) (A1/A4)
- **Mecanum_Line_Follower_digital**
Spurfolger mit Spursensor (A2/A1)
- **Mecanum_Synchronous_Navigation_Distance**
Synchroner Antrieb mit Distanz-Vorgabe (A1/A3)
- **Mecanum_Synchronous_Turning_Functions**
Synchrones Drehen (A1/A2)
- **Mecanum_Synchronous_Driving_Functions**
Synchroner Antrieb in alle Richtungen (A1/A1)
- **Mecanum_Collision_Prevention**
Hinderniserkennung mit Ultraschall (A1/E1)
- **Mecanum_Line_Follower_with_Color_Recognition_digital**
Spurfolger mit Farbsteuerung (A2/A3)
- **Mecanum_Line_Follower_with_PD_Controller**
Spurfolger mit PD-Regler (A2/E2)
- **Mecanum_Target_Reacher**
Encoder-Navigation (A1/E2)
- **Mecanum_Line_Follower_Obstacle_digital**
Spurfolger mit Hinderniserkennung (A2/A2)

6. Competition



Abb. 23: Zusatzset für Wettbewerbe zum Robotics Base Set

- **gui_sensor_gesture**
Test Gestenmodul **Python**
- **gui_sensor_combi**
Test Kombisensor

7. Robotics TXT 4.0 Base Set



Abb. 24: STEM Robotics TXT 4.0 Base Set - Grundlagen der Robotik begreifbar machen und nachhaltig verstehen

- **Barcode_Decoder**
9-stelliger Barcode (A4/A2)
- **Distance_Measuring**
Abstandsmessung (A1/A3)
- **Resistivity_Survey**
Widerstandsmessung (A1/A1)
- **Resistivity_Survey_with_Color_Code_Carbon_Film_Resistor**
Bestimmung Widerstand-Farbcode (drei Farbringe) (A1/E2a)
- **Resistivity_Survey_with_Color_Code_Metal_Film_Resistor**
Bestimmung Widerstands-Farbcode für Metallfilmwiderstände (vier Farbringe) (A1/E2b)
- **Temperature_Measuring**
Temperaturmessung (A1/A2)
- **Temperature_Measuring_SteinhartHart**
Temperaturmessung nach Steinhart-Hart (A1/E1)
- **Temperature_Measuring_with_Scale**
Temperaturmessung mit Display-Skalierung (A1/A2b)
- **Temperature_Measuring_with_Scale_and_MinMax**
Temperaturmessung mit Display-Skala mit „min“- und „max“-Wert (A1/A2c)
- **Boom_Gate_with_Encoder**
Schranke mit Magnetdecoder (A2/A3)
- **Boom_Gate_with_Pulse_Wheel**
Schranke mit Impulsrad (A2/A2)
- **Boom_Gate_with_Pushbutton_Switch**
Schranke mit Bedarfstaster (A2/E1)
- **Boom_Gate_with_Pushbutton_Switch_and_Lightbeam**
Schranke mit Durchfahrtskontrolle (A2/E2)
- **Boom_Gate_with_Pushbutton_Switch_Lightbeam_and_Counter**
Schranke mit Durchfahrt-Zähler (A2/E3)
- **Boom_Gate_with_Stop_Position_Switch**
Schranke mit Endlagentaster (A2/A1)
- **Carpark_Monitoring_with_two_Boom_Gates**
Schranke mit Belegungskontrolle (A2/E4)
- **Blinking_Light**
Blinklicht (A3/E1)
- **Boom_Gate_with_Pushbutton_Switch_Lightbeam_and_Traffic_Light**
Parkhausschranke mit Ampelsteuerung (A3/A2)

- **Boom_Gate_with_Pushbutton_Switch_Lightbeam_Counter_and_Blinking_Light**
Parkhausschranke mit Durchfahrt-Zähler und Blinklicht (A3/E2)
- **Carpark_Monitoring_with_two_Boom_Gates_and_Blinking_Lights**
Parkhausschranke mit Belegungskontrolle und Blinklicht (A3/E3)
- **Traffic_Light_with_Pushbutton_Switch**
Ampelsteuerung (A3/A1)
- **Bar_Width_Recognition**
Balkenerkennung (A4/A1)
- **Camera_Color_Decoder**
Farberkennung (A4/E2)
- **Code39_Decoder**
Code-39-Dekodierer (A4/A3)
- **Code39_Decoder_with_Error_Detection**
Fehlererkennung (A4/A4)
- **Code39_Template**
Code-39-Dekodierer (modifiziert) (A4/A3)
- **Colored_Barcode_Decoder**
Mehrdimensionaler Code (A4/E3)
- **Line_Detection**
Linienerkennung (A4/A1)
- **RGB_Colorcode**
Farberkennung (A1/E1)
- **RGB_Decoder**
RGB-Code (A4/E1)
- **Bit_Sequence_Sender**
Binärkode-Sender (A5/A2)
- **Bit_Sequence_Sender_Receive_LED**
Bitfolgesender mit Empfangsanzeige (A5/A2c)
- **Bit_Sequence_Sender_Receiver**
Bitfolgeempfänger (A5/E1)
- **Light_Signal**
Lichtsignalempfänger (A5/A1b)
- **Light_Signal_Counter**
Lichtsignalempfänger mit Zähler (A5/A1c)
- **Morse_Code_Sender**
Morse-Zeichen-Sender (A5/A3)
- **Morse_Code_Sender_Receiver**
Morse-Zeichen-Empfänger (A5/E2)
- **Morse_Code_Sender_Receiver_with_Caesar_Encryption**
Verschlüsselung-Caesar-Chiffre (A5/E4)
- **Morse_Code_Sender_Receiver_with_Vigenere_Encryption**
Verschlüsselung-Vigenère-Chiffre (A5/E5)
- **Morsecode_Template**
Programmfragment für Morsecode
- **Buggy_Driving_Test**
Einfache Geradeausfahrt (A6/A1)
- **Buggy_Driving_Test_Synchronous**
Synchroner Antrieb (A6/A2)
- **Buggy_Driving_Test_Synchronous_Distance**
Kontrollierte Fahrt mit Encodern (A6/A3)
- **Buggy_Synchronous_Turning_Calibration**
Drehen um vorgegebenen Winkel (A6/A5)
- **Buggy_Target_Reacher**
Buggy mit Encoder-Navigation (A6/E3)
- **Buggy_Turning_Test_Synchronous**
Synchrones Drehen auf der Stelle (A6/A4)
- **Buggy_Turning_Test_Synchronous_Angle**
Drehen um vorgegebenen Winkel (A6/A5)
- **Buggy_with_Pushbuttons**
Buggy mit Hinderniserkennung (A6/A6)
- **Buggy_with_Pushbuttons_and_IR_Sensors**
Buggy mit Linien-/Kantenerkennung (A6/A7)
- **Buggy_with_Pushbuttons_IR_Sensors_and_Light**
Buggy mit Frontstrahler (A6/E1)
- **Buggy_with_Ultrasonic**
Hinderniserkennung mit Ultraschall (A6/E2a)
- **Buggy_with_Ultrasonic_and_Light**
Ergänzung um Frontstrahler (A6/E2b)

- **Buggy_with_Ultrasonic_flexible_Turn**
Drehung, bis kein Hindernis bei 25 cm (A6/E2c)
- **Buggy_with_Ultrasonic_flexible_Turn_display_Angle**
Drehung mit Winkelanzeige (A6/E2d)
- **Buggy_Coordinates_Drawing**
Malen nach Zahlen (A7/E2)
- **Buggy_Drawing_Polygons**
Zeichnen eines N-Ecks (A7/A3)
- **Buggy_Driving_Functions**
Steuerungsfunktion (A7/A1)
- **Buggy_House_of_Santa_Claus**
Haus vom Nikolaus (A7/A2)
- **Buggy_Move_2_Point**
Zielpunkt ansteuern (A7/E1)
- **Drawing_Coordinates**
Zeichnungs-Koordinaten
- **Buggy_Line_Follower_digital**
Spurfolger mit Spursensor (A8/A1)
- **Buggy_Line_Follower_digital_optimized**
Optimierter Spurfolger (A8/A2)
- **Buggy_Line_Follower_with_Color_Recognition_digital**
Steuerung mit Farbflächen (A8/A4)
- **Buggy_Line_Follower_with_Obstacle_Detection_digital**
Spurfolger mit Hinderniserkennung (A8/A3)
- **Buggy_Line_Follower_with_P_Controller**
Spurfolger mit Proportionalregler (A8/E2)
- **Buggy_Line_Follower_with_PD_Controller**
Spurfolger mit PD-Regler (A8/E3)
- **Buggy_Line_Recognition**
Spurenerkennung mit Kamera (A8/E1)
- **HEX_Colorcode**
Farbwert als HEX-Code ausgeben

Quellen

- [1] fischertechnik: *Robotics Hightech*. Auf dem [fischertechnik-E-Learning-Portal](#).
- [2] fischertechnik: *fischertechnik Robotics TXT 4.0 Base Set und ADD ONs*. Auf [fischertechnik.de](#).
- [3] Axel Chobe: *Überblick der Beispielprogramme von ROBO Pro Coding für den TXT 4.0*. In Tabellenform auf [chobe.info](#).

Modell

Laserprojektion (2): Lissajous-Figuren

Arnoud van Delden

Im ersten Teil dieser Beitragsfolge habe ich beschrieben, wie man sogenannte Hypozykloiden mithilfe eines Halbleiterlasers und zweier rotierender Spiegel projiziert [5]. Diese Kurven wurden durch Kreisbewegung entlang des Umfangs eines anderen Kreises erzeugt. Wenn wir die Spiegel jedoch nicht drehen, sondern jeden in eine andere Richtung schwingen lassen, entstehen sogenannte Lissajous-Figuren. Als logische Fortsetzung der bisherigen Experimente mit Laserprojektion habe ich versucht, die Figuren rein mechanisch mit unserem Lieblingsbaukastensystem zu erzeugen.

Einführung

Die mit rotierenden Spiegeln projizierten Hypozykloiden lassen sich mathematisch in Polarkoordinaten ausdrücken. Formeln für Lissajous-Figuren liefern hingegen kartesische (x- und y-) Koordinaten. Die Figuren können durch die Übertragung von Schwingungen auf Spiegel projiziert werden. Sie werden auch nach dem Nautiker *Nathaniel Bowditch* (1773-1838) Bowditch-Kurven genannt, der sie bereits 1815 beschrieb. Die Kurven wurden später von dem Physiker *Jules-Antoine Lissajous* (1822-1880) genauer untersucht. Lissajous war Professor für Mathematik am Lyceé Saint-Louis in Paris, wo er alle Arten von Schwingungen und Wellen untersuchte. 1855 erfand er ein einfaches optisches Gerät zur Analyse zusammengesetzter Schwingungen: Er befestigte Spiegel an den Zinken zweier Stimmgabeln, die im rechten Winkel zueinander vibrierten. Wenn ein Lichtstrahl auf einen der Spiegel gerichtet wurde, prallte er zum anderen Spiegel und von dort auf einen Bildschirm zurück, wo er ein zweidimensionales Muster bildete – das Ergebnis der Überlagerung der beiden Schwingungen. Dieses einfache Gerät (ein Vorläufer des modernen Oszilloskops) war zu seiner Zeit eine Neuheit; bis dahin hing

die Erforschung von Geräuschen ausschließlich vom Hörprozess ab, also vom menschlichen Ohr. Lissajous machte es buchstäblich möglich, Klang zu „sehen“.

Wer noch mehr mathematische Hintergrundinformationen und Theorie zu den Figuren sucht, kann sich online verwöhnen lassen. Die Wikipedia-Seite bietet einen guten Ausgangspunkt [1]. Auch für diejenigen, die sich sofort an die Arbeit machen wollen, werden in der neueren Literatur gute Experimente beschrieben [2].

Verschiedene Lissajous-Figuren

Bevor ich mit mechanischer Schwingung und Projektion mit einem Lichtstrahl experimentierte, beschloss ich, einen Blick auf die Figuren auf meinem Oszilloskop zu werfen. Im XY-Modus eines Oszilloskops bestimmt ein Signal die Y-Ablenkung und das andere Signal die X-Ablenkung, die normalerweise den Zeitablauf darstellt. Auf diese Weise wird der Frequenzzusammenhang der beiden Signale sichtbar gemacht.

Stellt man auf diese Weise zwei Sinuswellen dar, erscheinen bald erkennbare Lissajous-Figuren auf dem Bildschirm. In Abb. 1 sind oben die beiden im Zeitbereich

dargestellten Sinuswellen und darunter im XY-Modus die Lissajous-Figur zu sehen.

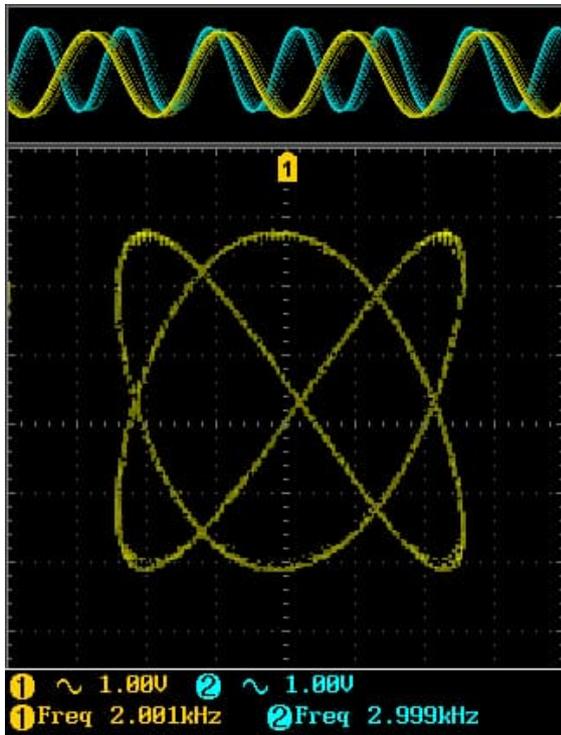


Abb. 1: Lissajous-Figur im Verhältnis 2:3 auf dem Oszilloskop

Wenn ihr kein Oszilloskop und keinen Wellengenerator habt, findet ihr online verschiedene Lissajous-Simulatoren [3], mit denen ihr virtuell experimentieren könnt. Durch Variation der Phasenverschiebung zwischen zwei Wellen derselben Frequenz wird beispielsweise deutlich, dass es sich bei der schrägen Linie auf dem Display tatsächlich um die Seitenansicht eines geneigten Kreises handelt.

Einstellbare Frequenzverhältnisse

Für die Projektion mit einem Laserstrahl müssen zwei Spiegel unabhängig voneinander modulierbar sein. Abb. 2 [4] zeigt die Figuren, die (theoretisch) mit dem Getriebe aus Teil 1 dieser Beitragsfolge möglich sein sollten [5]. Wie bei den Hypozykloiden hängt die endgültige projizierte Form vom Schwingungsverhältnis der beiden beweglichen Spiegel zueinander ab.

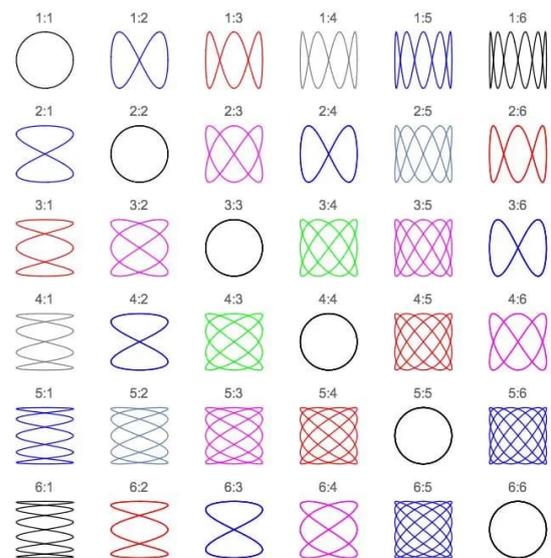


Abb. 2: Lissajous-Figuren, die theoretisch mit dem Getriebe aus [5] projiziert werden können

Die Bewegung jedes Spiegels sollte theoretisch möglichst rein in einer Richtung erfolgen. Die Herausforderung besteht darin, den störenden Einfluss von Bewegungen in andere Richtungen durch Vibrationen und Spiel in den Getrieben so weit wie möglich zu begrenzen.

Am naheliegendsten ist es, die oszillierende Bewegung mit einem Hebel auszuführen. Wird sie zudem über eine azentrisch angeordnete Welle an einem rotierenden Rad angetrieben, läuft die Bewegung direkt nach einer reinen Sinuswelle ab. Zu diesem Zweck habe ich die in Abb. 3 gezeigte Konstruktion auf der Abtriebsseite des Getriebes aufgebaut.

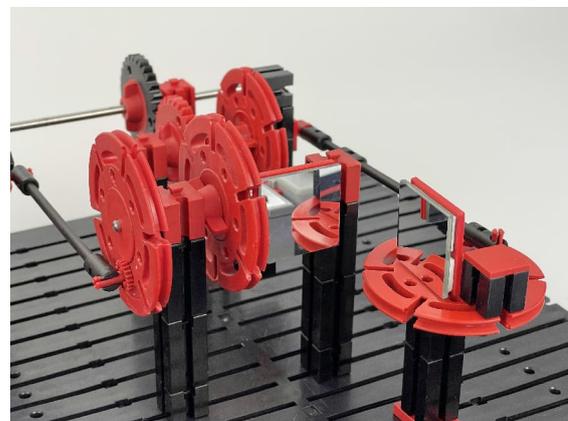


Abb. 3: Die beiden Schwingenspiegel

Der Ausschlag des Hebels sollte nur gering sein. Schließlich muss der durch die erste Spiegelbewegung abgelenkte Laserstrahl weiterhin vollständig auf den zweiten Spiegel fallen. Bei dieser Methode kann die Auslenkung des Spiegels durch Verschieben des Drehpunkts der Pleuelstange eingestellt werden. Dieser kann bei Bedarf nahe der Drehachse liegen.

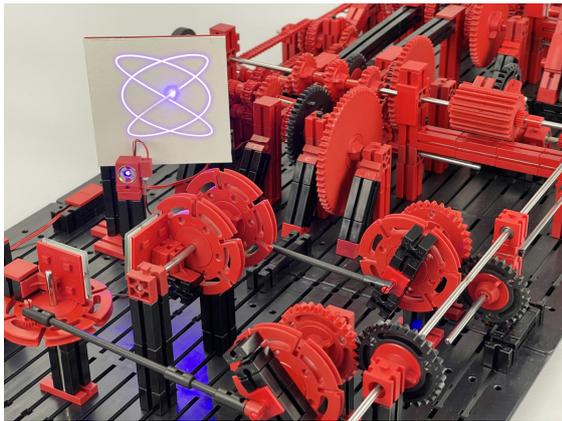


Abb. 4 - Fake-news! Diese Realität wurde besser dargestellt, als ich sie sehen konnte...

Leider ist die resultierende Projektion nicht mehr als ein sich sanft bewegender Lichtpunkt. Die Kopplung von Physik und Optik konnte sehr gut demonstriert werden, allerdings erwies sich eine maximale Schwingungsfrequenz von 25 Hz erwartungsgemäß als zu niedrig, um gut erkennbare Figuren zu projizieren. Die opportunistische Fotomontage von Abb. 4 vermittelt hoffentlich einen Eindruck davon, was selbst mit den langsamsten Verschlusszeiten nicht fotografiert werden konnte. Auf den von mir gemachten [verzögerten Filmbildern](#) lässt sich jedoch überprüfen, dass die Lissajous-Figuren durch den Lichtpunkt beschrieben werden.

Schnellere Spiegelschwingung

Durch den Einsatz zweier Einzelmotoren lässt sich die Geschwindigkeit der oben beschriebenen Hebelmethode natürlich erheblich steigern. Eine volle Hin- und Herbewegung des Spiegels entspricht jedoch einer Umdrehung des Antriebsrads mit dem Hebel, und ein kurzer Versuch zeigte, dass

sich die Kräfte und Vibrationen im Modell dann erheblich summieren können. Deshalb suchte ich nach einer Methode, bei der weniger Zahnräder und Massen in Bewegung zu setzen sind und die auch mit weniger Drehpunkten und Hebeln umgesetzt werden kann. Auch hier kam der 3D-Drucker zum Einsatz [6]. Der endgültige Prototyp einer Rolle mit ovalem Querschnitt, die die sinusförmige Bewegung auf die Spiegelplattform überträgt, ist in Abb. 5 dargestellt.

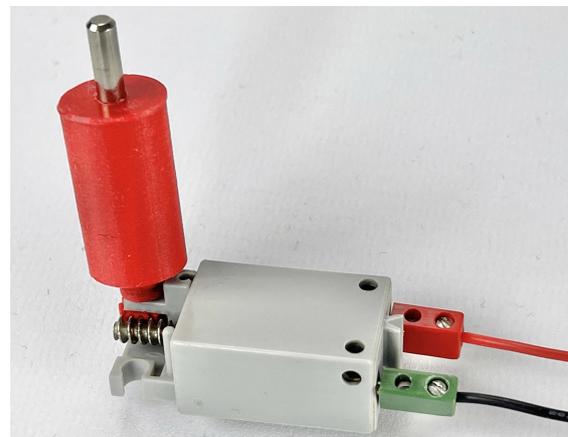


Abb. 5: Ovale Rolle zur Montage auf Schneckenwelle

Die Rolle kann direkt auf die Schneckenwelle ([31069](#)) eines Schneckengetriebes für den Mini-Motor oder S-Motor (Abb. 6) geschoben werden. Durch die Möglichkeit, die Oberfläche der Walze durch Führungen auf der Plattform abzutasten, wird die oszillierende Bewegung möglichst direkt und mit relativ geringer Auslenkung übertragen. Darüber hinaus werden beide Plattformen von einem eigenen Motor angetrieben, sodass auf das Getriebe verzichtet und das Ganze auf einer kleinen Bauplatte untergebracht werden kann. Dadurch werden immerhin 60 Zahnräder und 16 Kettenübersetzungen eingespart. Für die Motorsteuerung habe ich wieder meinen Zauberling [7] ausgewählt, der bereits eine Doppelmotorsteuerung enthält. Auch mit einem ftDuino, fischertechnik TXT-Controller oder zwei separaten Motorcontrollern lässt sich natürlich etwas Ähnliches aufbauen.

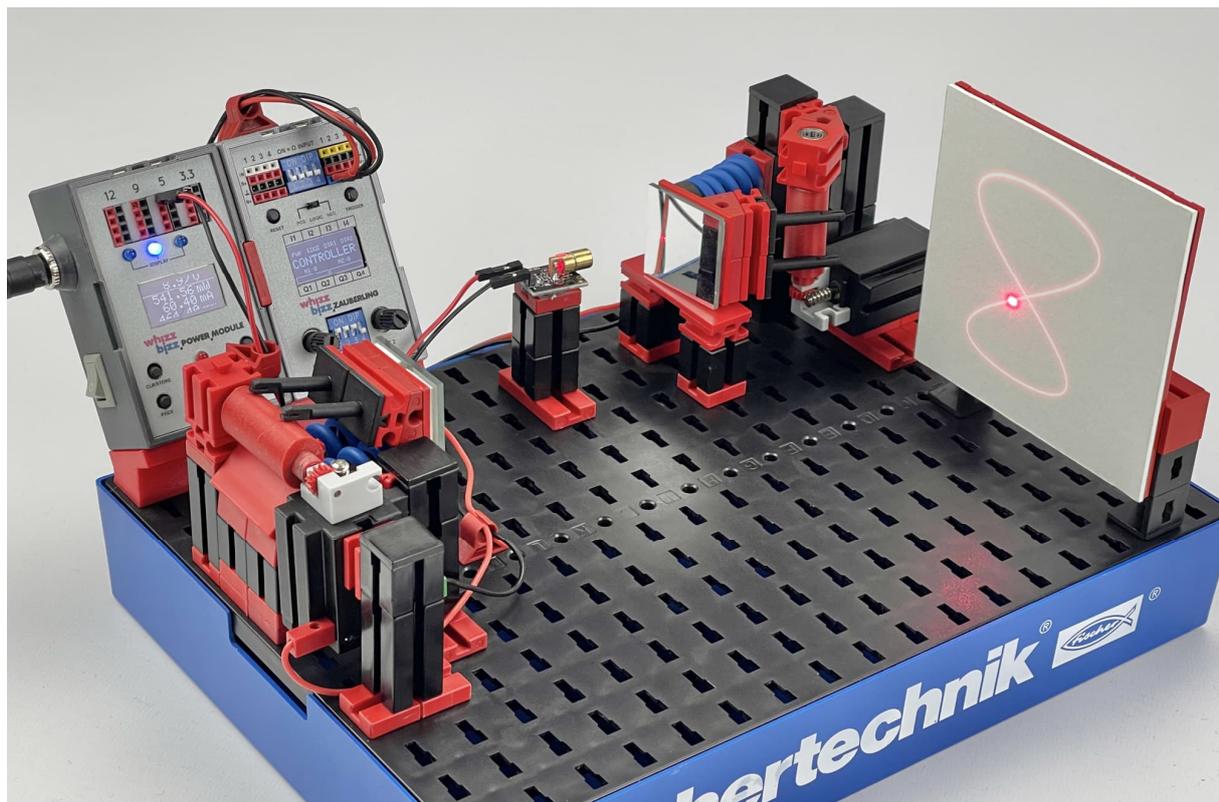


Abb. 6: Separate Motoren für die x- und y-Auslenkung (Foto illustrativ manipuliert)



Abb. 7: Draufsicht auf die X-Ablenkung:
Die Rolle befindet sich unter dem Kugellager

Mehr zu den Experimenten findet ihr auf der Projektseite meiner Website [8]; dort findet sich auch ein Link zum Video.

Durch Experimentieren mit den verschiedenen Drehzahlverhältnissen kann theoretisch der gesamte Bereich der in Abb. 2 gezeigten Lissajous-Figuren durchlaufen werden. Es erwies sich jedoch als wesentlich schwieriger als bei den Hypozykloiden [5], die Figur wirklich stabil zum Stehen zu bringen. Aufgrund von Spiel und Vibrationen

stimmen auch die Auslenkrichtungen nicht zu 100 %.

Der X-Spiegel könnte auf einem Tisch abgestützt werden und „vibrieren“. Die Y-Umlenkung schwingt frei hängend an den Federn (31892) und ist daher etwas empfindlicher gegenüber störenden Einflüssen. Es lohnt also durchaus zu prüfen, ob die Federung der Spiegelplattformen in diesen Punkten verbessert werden kann.

Fazit

Mit etwas Experimentieren ist es gelungen, Lissajous-Figuren auf rein mechanische Weise mit einem Halbleiterlaser zu projizieren. Für stabile Projektionen sind jedoch relativ hohe Geschwindigkeiten erforderlich, bei denen die Bewegung des Spiegels pro Motor möglichst rein auf eine Achse begrenzt wird. Der Einfluss von Vibrationen und Spiel führt schnell zu einer etwas „zitterigen“ Projektionslinie. Zudem

schränkt die relativ geringe Schwingungsfrequenz die Erkennbarkeit der Figuren ein.

Dennoch lässt sich der Schluss ziehen, dass das Projizieren von Lissajous-Figuren mit einem Laser eine unterhaltsame und kreative Möglichkeit ist, die Eigenschaften von Wellen und Vibrationen zu untersuchen. Es lassen sich komplexe und schöne Muster erstellen, die sowohl visuell beeindruckend als auch wissenschaftlich informativ sind.

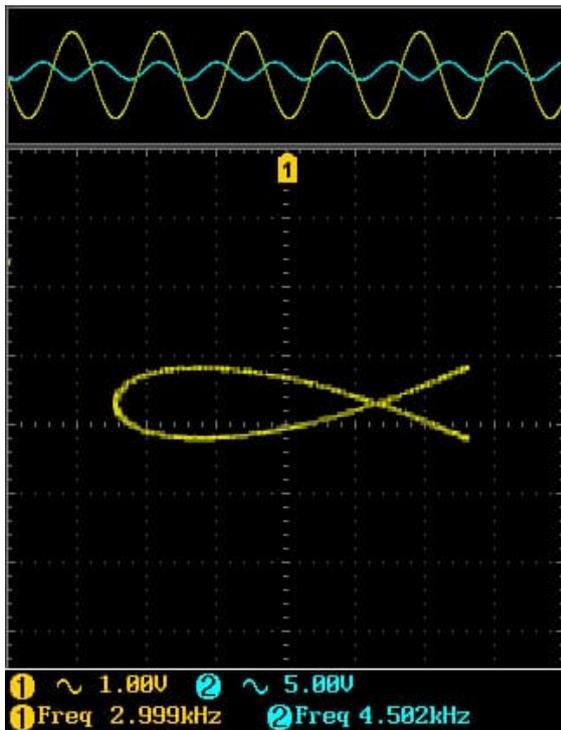


Abb. 8: Eine ganz besondere Lissajous-Figur erscheint im Verhältnis 3:4,5 (Könnte es ein Zufall sein, dass 3 und 4,5 in das fischertechnik-Raster fallen?)

Für die professionelle Laserprojektion scheinen die hier untersuchten mechanischen Lösungen wenig brauchbar zu sein. Dabei kommen spezielle Lasergalvanometer zum Einsatz, die eine wesentlich schnell-

lere und genauere Modulation der Spiegel ermöglichen. In seiner einfachsten Form enthält ein solches „Laser-Galvo-Spiegelsystem“ lediglich zwei kleine Gleichstrommotoren, die mit einem symmetrischen Eingangssignal direkt bewegt werden können. Vielleicht wäre es ein mögliches zukünftiges Experiment, zu untersuchen, ob die fischertechnik-Motoren mit der richtigen elektronischen Steuerung dafür eingesetzt werden können. Es würde den störenden Einfluss der Mechanik beseitigen, uns aber vor neue Herausforderungen stellen. Schließlich sollen die Spiegel mit möglichst reinen Sinussignalen angesteuert werden, deren Frequenz sich genau steuern lässt.

Quellen

- [1] Wikipedia: [Lissajous-Figuren](#).
- [2] Thomas Püttmann: *Mathematik verstehen mit fischertechnik*. Kapitel 25: „Der Harmonograph“. dpunkt Verlag, 2022.
- [3] academo: [Lissajous-Simulator](#).
- [4] ThatsMaths: [Hearing Harmony. Seeing Symmetry](#).
- [5] Arnoud van Delden: *Laserprojektion (Teil 1): Hypozykloide*. [ft:pedia 1/2023](#), S. 60–69.
- [6] Printables: [STL-Modell zum Herunterladen](#).
- [7] Arnoud van Delden: *Der Zauberling (Teil 4): Die Weiterentwicklung*. [ft:pedia 1/2022](#), S. 71–79.
- [8] Whizzbizz: [Projektseite](#).

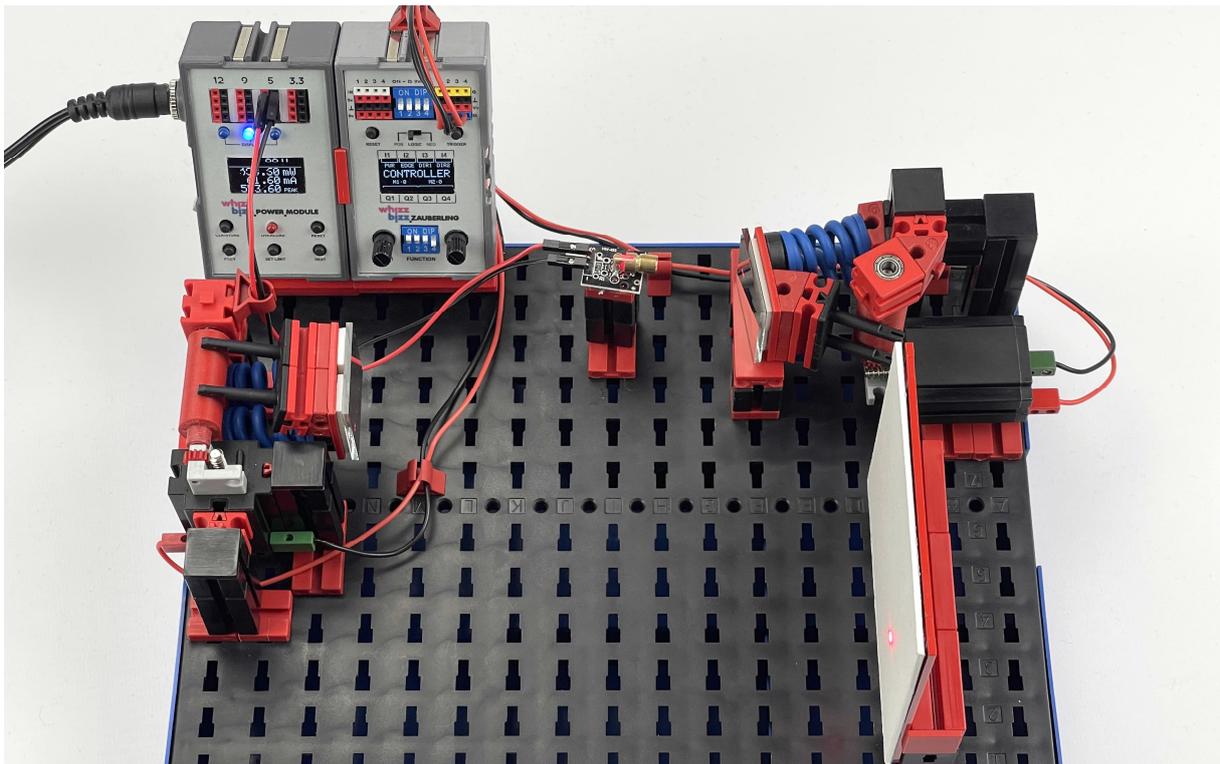


Abb. 9: Stark vereinfachte Konstruktion

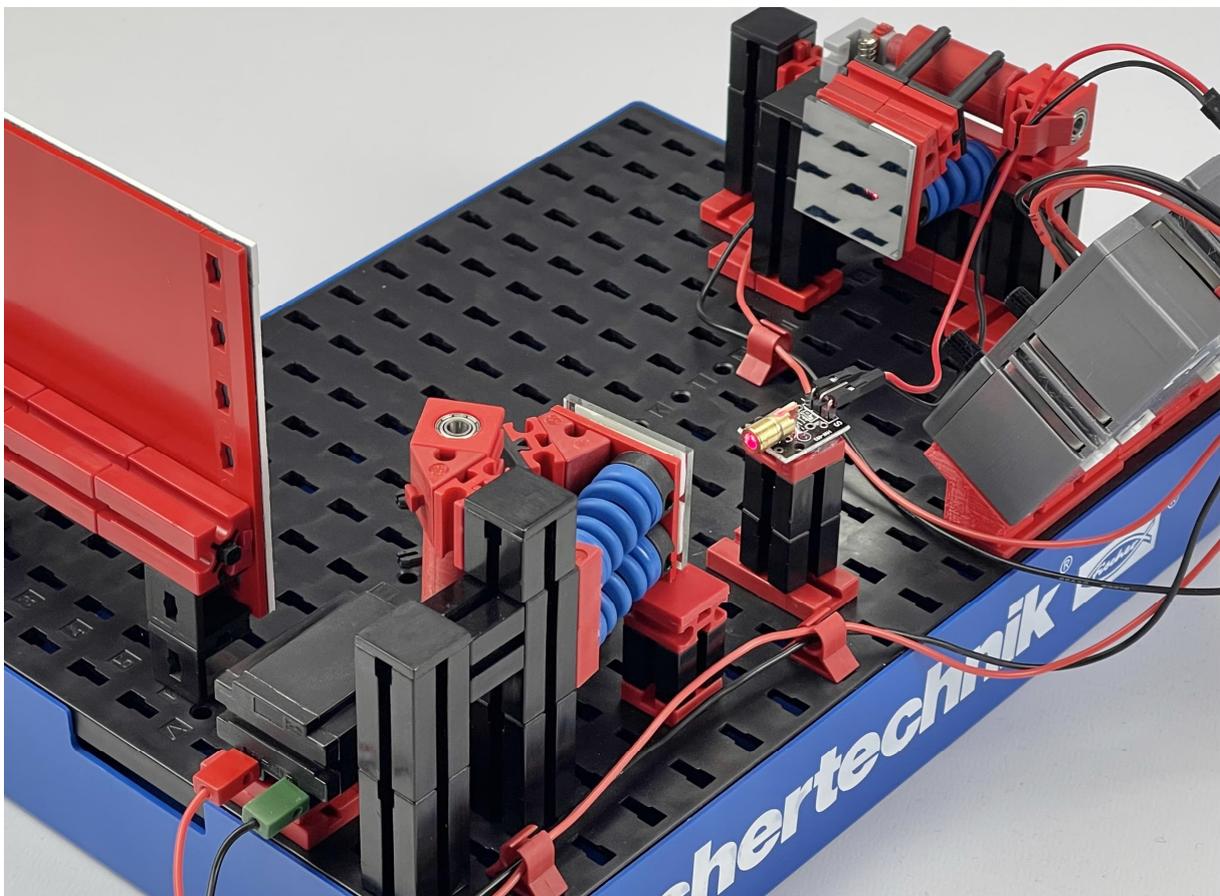


Abb. 10: Blick „hinter die Kulissen“

Computing

Balancier-Plattform

Florian Bauer

In diesem Beitrag wird eine kardanische Balancier-Plattform vorgestellt, die in zwei Richtungen gekippt werden kann. Es wird gezeigt, wie man diese Plattform manuell und mit einem Computer ansteuern kann, um eine Kugel zu balancieren.

Die Idee

Die Idee zu diesem Aufbau kam mir auf der Elektronikmesse embedded World 2023 [2]. Dort habe ich einen Tilt-Maze-Aufbau gesehen, mit dem man mittels eines Joysticks eine Kugel durch ein Labyrinth vom Start zum Ziel dirigieren kann (Abb. 1). Ich muss zugeben, dass mich in den 90er Jahren das Spiel Marble Madness [3] auf dem Amiga sehr begeistert hat. Vielleicht habe ich mir deshalb zu Hause überlegt, wie man eine echte Kugel mit fischertechnik steuern kann.



Abb. 1: Computer-steuerbare Tilt-Maze-Plattform auf der embedded World 2023

Aufbau

Eine Idee, um eine Plattform in zwei Richtungen kippen zu können, besteht darin, sie in der Mitte auf einem Kugelgelenk zu

lagern. Alternativ kann man anstatt eines Kugelgelenks auch ein Kardan-Gelenk verwenden. Anstatt die Plattform in der Mitte zu lagern kann man sie auch kardanisch aufhängen [4]. Diesen Ansatz habe ich dann weiterverfolgt. Letztendlich ist dabei der Aufbau herausgekommen, der in Abb. 2 gezeigt ist. Für eine verbesserte Stabilität wurde der Rahmen mit Statik-Elementen verstärkt.

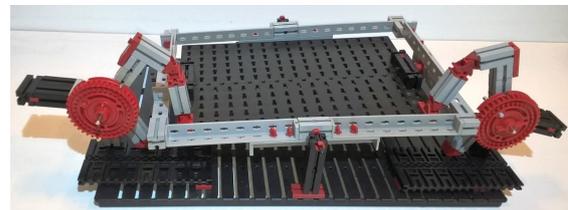


Abb. 2: Kardanisch gelagerte Tilt-Plattform aus fischertechnik

An den Drehscheiben mit den proximalen Kipp-Armen befinden sich Gegengewichte, die die Plattform ohne externes Haltemoment nahezu im Gleichgewicht halten. Die proximalen Arme sind jeweils mit einem 15er Stein an den Drehscheiben befestigt und mit Federnocken gegen Herausrutschen gesichert.

Ansteuerung

Als nächstes stellt sich die Frage, wie die Plattform in zwei Richtungen gekippt werden kann. Inspiriert von dem Modell auf der Messe wollte ich die Anlenkung über zwei Arme links und rechts der Plattform realisieren. Jeder Arm besteht jeweils aus

zwei Teilarmen: einem proximalen Arm, der gedreht wird, und einem distalen Arm, der den proximalen Arm mit der Plattform verbindet.

Wenn sich beide Arme gegensinnig nach innen bewegen, wird die Plattform nach vorne unten gedrückt. Wenn sie sich gegensinnig nach außen drehen erfolgt die Kippung nach hinten. Analog für die andere Richtung: Bewegen sich die Arme gleichsinnig nach rechts bzw. links wird die Plattform nach links bzw. rechts gekippt.

Eine kleine Herausforderung bei der Ansteuerung ist, dass die Lager an beiden Enden der distalen Arme in zwei Freiheitsgraden beweglich sein müssen. Im fischertechnik-Forum wurde der Vorschlag gemacht, Kardan-Gelenke für die Lager zu verwenden. Diesen Vorschlag habe ich auch umgesetzt. Er funktioniert sehr gut und ist auch in dem in Abb. 2 gezeigten Modell realisiert.

Alternativ kann man die Bewegungsfreiheitsgrade durch Kugelgelenke realisieren. Da fischertechnik meines Wissens momentan keine solchen Gelenke im Programm hat, habe ich ein Bauteil konstruiert, das mit fischertechnik kompatibel ist, und es mit einem 3D-Drucker gedruckt (Abb. 3) [6].

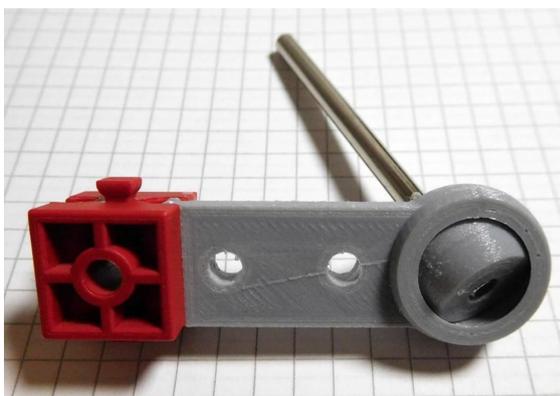


Abb. 3: Kugelgelenk für fischertechnik mit drei Freiheitsgraden

Die Gelenke zwischen proximalen und distalen Armen sind nur als Kipp-Lager mit einem Freiheitsgrad ausgeführt. Streng genommen sollte man hier auch Kugellager

einsetzen. Es geht aber auch so, da die Flexibilität der Konstruktion die entstehenden minimalen Verkippungen aufnehmen kann. Eine Plattform mit diesen Gelenken ist in Abb. 4 zu sehen.



Abb. 4: Tilt-Plattform mit Kugelgelenken aus dem 3D-Drucker

Kinematik

Beim Aufbau der Tilt-Plattform handelt es sich um einen parallelen Mechanismus. Seine Kinematik kann durch eine sog. „Loop Closure Equation“ [14] beschrieben werden. Dies ist eine trigonometrische Gleichung, die die Ansteuerwinkel A und B der Aktoren mit den Kippwinkeln der Plattform X und Y in Beziehung setzt. Für die Closure Equation berechnet man einerseits die Position der Lager auf der Plattform $P[i]$ in Abhängigkeit von den Kippwinkeln X und Y . Andererseits benötigt man die Position $Q[i]$ der Lager auf den proximalen Armen in Abhängigkeit von den Aktor-Stellwinkeln A und B . Die geometrischen Abstände von $P[i]$ und $Q[i]$ müssen der Länge der distalen Arme entsprechen. Damit erhält man zwei trigonometrische Gleichungen, die (A, B) mit (X, Y) in Beziehung setzen und die simultan gelöst werden müssen.

Bei der sog. Vorwärts-Kinematik gibt man A und B vor und berechnet daraus die Kippwinkel X und Y , während man bei der Inversen Kinematik X und Y vorgibt und A und B ausrechnet. Die Plattform kann nur um kleine Winkel in jede Richtung gekippt werden. Es stellt sich heraus, dass der Zusammenhang zwischen A, B und X, Y für kleine Winkel näherungsweise linear ist:

$$A = a * (X + Y) / 2$$

$$B = b * (X - Y) / 2$$

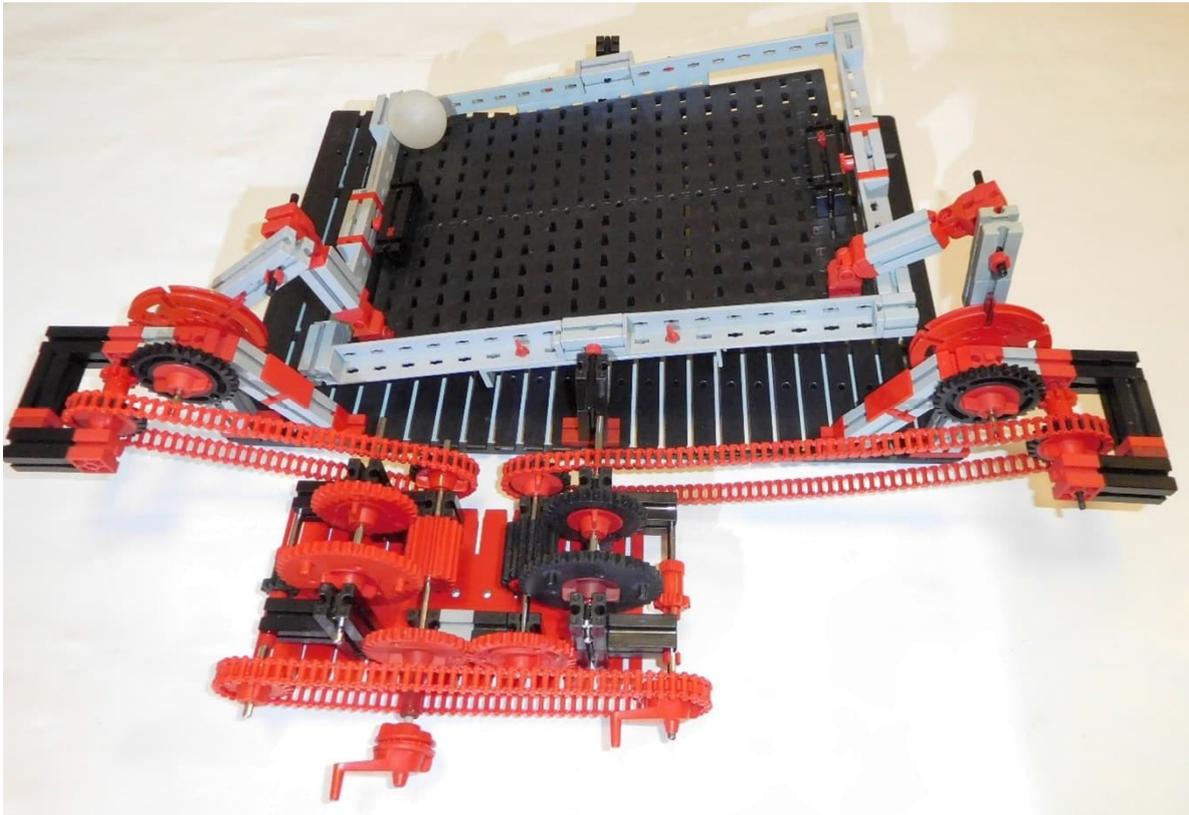


Abb. 5: Manuell steuerbare Tilt-Plattform mit „Differential-Mixer“, um die Kipp-Achsen der Plattform zu entkoppeln. Linke Kurbel: gegensinnige Bewegung der Arme für Kippung der Plattform um die X-Achse nach vorne bzw. hinten. Rechte Kurbel: gleichsinnige Bewegung der Arme für Kippung der Plattform um die Y-Achse nach links bzw. rechts.

Die Skalierungsfaktoren a und b hängen von der konkreten Geometrie ab und können entweder durch Rechnung oder durch Eichung ermittelt werden.

Manuelle Ansteuerung über ein Mixer-Differential-Getriebe

Für eine „benutzerfreundliche“ manuelle Ansteuerung, bei der ein X-Stellsignal in eine Kippung um X und ein Y-Stellsignal in eine Kippung um Y gewandelt wird, braucht man einen „Mixer“, der den X- bzw. Y-Stellwinkel in gleichsinnige bzw. gegensinnige Winkeländerungen A und B für die Aktoren umsetzt. Diese Operationen kann man mit Differential-Getrieben erreichen. In Abb. 5 ist eine Tilt-Plattform mit einem mechanischen Mixer gezeigt.

Mit diesem Getriebe lassen sich X- und Y-Richtung unabhängig voneinander ansteuern und man kann damit auch ohne

Mikrocontroller – ganz analog – eine Kugel auf der Plattform bewegen.

Elektronische Ansteuerung mit Modellbau-Servos

Als nächstes kann man sich fragen, wie man die Plattform mittels Mikrocontroller ansteuern kann. Dafür benötigt man Aktoren, die die gewünschten Stellwinkel realisieren.

RC-Modellbau-Servos eignen sich sehr gut dafür. Ihr Stellwinkel wird über ein PWM-Signal (Puls-Weiten-Modulation [12]) eingestellt. In dem Modell in Abb. 7 habe ich Modellbau-Servos als Aktoren (Abb. 6) eingesetzt. Für ihre Ansteuerung habe ich einen Arduino-Mega mit einem Motor-Shield L293D [15] verwendet. Die Verschaltung ist in Abb. 6 gezeigt.

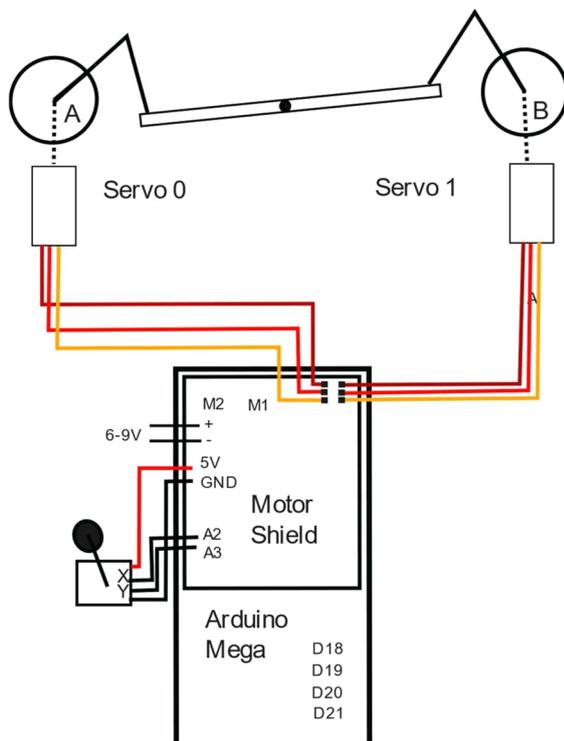


Abb. 6: Verdrahtungsplan für Tilt-Plattform-Steuerung über Joystick mit Servos

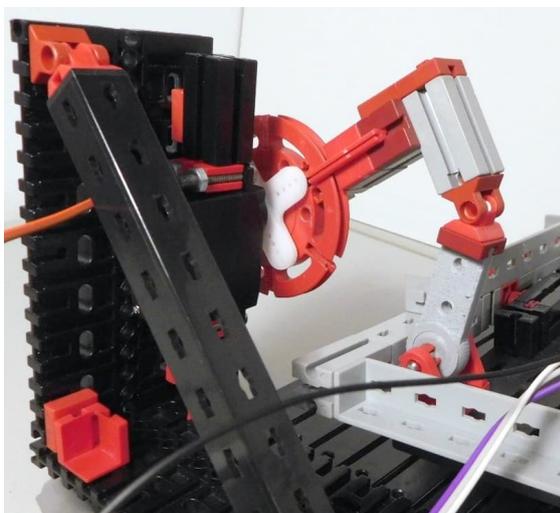


Abb. 7: Aktor mit Modellbau-Servo. Die Drehscheibe ist mittig auf das Steuerkreuz des Servos geschraubt. Zusätzlich ist die Drehscheibe an einer Strebe des Kreuzes mit einer Schraube fixiert um Verrutschen zu vermeiden. Der Servo ist mit M4-Schrauben an Standard-fischertechnik-Teilen fixiert.

Der Arduino liest die X- und Y-Winkel von einem 2D-Potentiometer-Joystick aus und berechnet den PWM-Wert für das Stell-Signal der Servos unter Berücksichtigung

der Mixer-Beziehung. Wichtig ist, dass man die Stellwinkel auf $\pm 15^\circ$ begrenzt, da sonst die Plattform auf der Grundplatte aufsetzt.



Abb. 8: Tilt-Plattform mit Servo-Aktoren

Elektronische Ansteuerung mit Encoder-Motoren

fischertechnik hat Encoder-Motoren im Programm. Können diese Encoder-Motoren anstelle der Servos verwendet werden? Um einen Gleichstrommotor als Servo betreiben zu können wird üblicherweise die Drehposition der Motorachse über einen Winkelencoder gemessen und die Motor-Winkelposition über einen PID-Regler [13] geregelt. Es gibt zahlreiche Abwandlungen und Verbesserungen für PID-Regler für Motorpositionen als auch andere Regelungstypen wie Gleit-Regime-Regler, auf die hier nicht näher eingegangen werden soll. Als Beispiel sei hier auf [16] verwiesen.

PID steht für Proportional-Integral-Differential-Regler. Ein PID-Regler hat als Input einen Sollwert und einen Istwert und liefert in Abhängigkeit von der Abweichung beider Werte voneinander einen Stellwert am Ausgang.

Die Wahl der PID-Parameter bestimmt das Verhalten des Regelkreises. Die Parameter werden so gewählt, dass sich das Ist-Signal dem Soll-Wert möglichst schnell angleicht, ohne dass es zu Regel-Schwingungen oder zu langsamem „Herankriechen“ an den Sollwert kommt.

Die 9V-fischertechnik-Encoder-Motoren besitzen standardmäßig einen Impulsausgang, der etwa 63,33 Impulse pro Umdrehung der Getriebeachse liefert.

Da die Motoren nur einen Ausgang haben, kann man die Drehrichtung nicht messen. Man kennt zwar bei der Ansteuerung die Polarität der Drehrichtung. Da der Rotor des Motors ein Trägheitsmoment besitzt, kehrt er bei Umpolung nicht sofort um, sondern läuft erst noch etwas in die ursprüngliche Richtung. Zudem kann sich der Motor durch ein von außen wirkendes Drehmoment auch kurzzeitig in die andere Richtung bewegen. Beide Effekte führen zu Fehlzählungen, die sich im Laufe der Zeit schnell addieren. Somit funktioniert dieser Ansatz nicht zuverlässig. (Es sei hier angemerkt, dass der Nachlauf-Effekt unkritisch ist, wenn man dem Motor bis zum vollständigen Stillstand Zeit gibt: Man wartet einfach eine Zeit, bis keine Impulse mehr ankommen.) Für einen Betrieb des Gleichstrommotors als Servo in einem Regelkreis ist dieses Abwarten aber inakzeptabel.

Besser ist es, dem Motor einen zweiten Impulsausgang zu geben, der zusammen mit dem ersten Signal ein echtes Quadratur-Signal erzeugt. Damit lässt sich die Drehrichtung des Motors zuverlässig erkennen und es treten theoretisch keine Fehlzählungen auf.

In der Praxis kann es leider doch zu einigen (im Allgemeinen wenigen) Fehlzählungen kommen, wenn das Programm zu langsam ist oder Störungen vorliegen. Eine Möglichkeit ist, den Zähler für den Motor bei Detektion der Nullposition zurückzusetzen. Dazu muss aber ein weiteres Signal (Z-Signal) erzeugt und ausgewertet werden.

fischertechnik-Encoder-Motoren wie die 24V-Motoren ([144643](#) oder 168988), die einen zweiten Impulsausgang haben, eignen sich gut als schnelle und leistungsfähige Servo-Motoren. In dem Beitrag „Quadratur-Encoder für fischertechnik-Encoder-Motoren“ beschreibe ich, wie man auch die gängigen 9V-Encoder-Motoren des neueren Typs auf einen Quadratur-Ausgang umrüsten kann.



*Abb. 9: Modifizierter Encoder-Motor mit Zwei-Kanal-Quadratur-Signal-Ausgang als Aktor für die Tilt-Plattform.
Kabelfarben: braun: GND, rot: VCC (5V), orange: A-Signal, grün: B-Signal*

Abb. 9 zeigt einen Aktor mit einem modifizierten 9V-Encoder-Motor.

Ein Wort noch zu einer externen Drehwinkel-Positionsbestimmung: Wie bereits erwähnt haben die Encoder-Motoren ein eingebautes Getriebe. Man kann die Drehwinkelposition eines Motors auch am Ausgang nach dem Getriebe auswerten, zum Beispiel mit Hilfe eines magnetischen Winkel-Encoders (u. a. AS56000 [10]) oder eines optischen Encoders. Der Vorteil ist, dass man damit auch Motoren ohne eingebauten Encoder (wie XM-Motoren [135485](#)) als Aktoren verwenden kann. Ich habe aber festgestellt, dass es bei PID-Regelkreisen mit dieser Konfiguration oft zu Oszillationen und Rüttelbewegungen kommt, die dann eine empfindliche Regler-Einstellung erfordern, die bei mir fast immer mit Abstrichen in der Geschwindigkeit einhergingen. Ggf. sind noch weitere Maßnahmen nötig, um die durch Reibung hervorgerufenen Limit-Oszillationen zu kompensieren [17]. Starke Rüttelbewegungen sind Gift für Motoren und Getriebe, da sie zu erhöhtem Verschleiß führen.

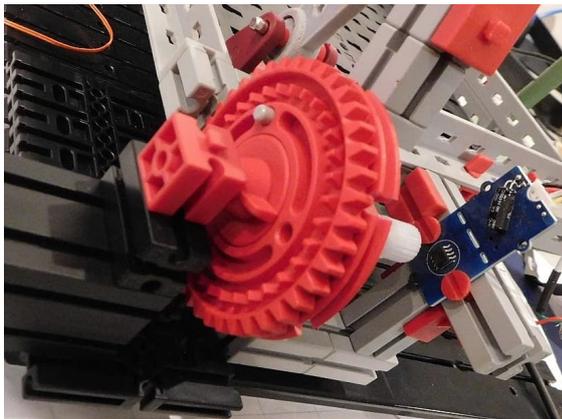


Abb. 10: Aktor mit XM-Motor und magnetischem Winkel-Encoder. Auf der Achse der Drehscheibe sitzt ein Neodym-Magnet, der sich vor dem AS5600-Rotary-Encoder dreht. Dieser Encoder wird über I2C ausgelesen.

Die Auswertung des Quadratur-Signals kann durch einen Mikrocontroller erfolgen. Es sind zwei Digital-Eingänge notwendig, die mit hoher Priorität abgetastet werden müssen, um Fehlzählungen zu vermeiden. Man setzt hier meist Interrupt-Steuerungen ein. Da der Microcontroller oft weitere Aufgaben wie Serial-I/O oder Berechnungen für den Regler übernimmt, kann es im ungünstigen Fall zu Fehlzählungen kommen. Schnelle Hardware-Counter, die die Quadratur-Auswertungen übernehmen und gleich einen digitalen Zählwert an den Mikrocontroller liefern, können hier mit einem moderaten Hardware-Aufwand Abhilfe schaffen (siehe z. B. [8]).

Mikrocontroller und Steuerprogramm

Für die Ansteuerung der Kipp-Plattform sind zwei Encoder-Motoren mit Quadratur-Ausgang notwendig. Den Verdrahtungsplan zeigt Abb. 11. Die Quadratur-Signale werden mit vier Interrupt fähigen Pins erfasst und die analogen Steuersignale für X- und Y-Richtung von einem Potentiometer-Joystick eingelesen.

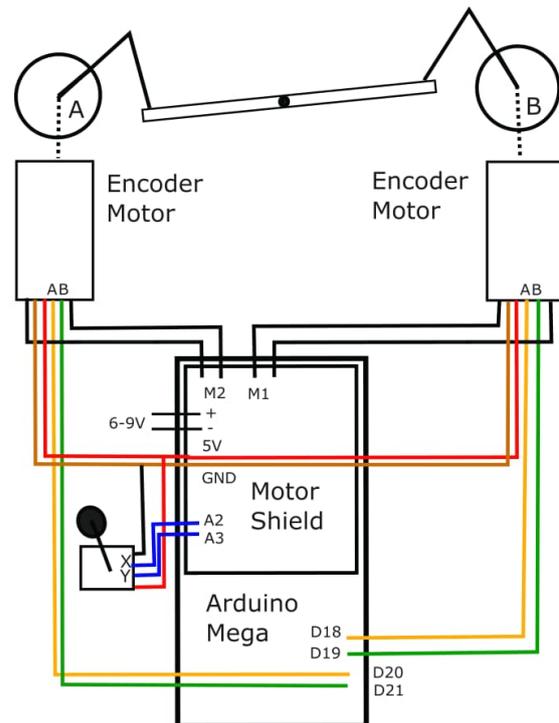


Abb. 11: Verdrahtungsplan für Tilt-Plattform-Steuerung über Joystick mit modifizierten fischertechnik-Encoder-Motoren

Balancieren eines Balls

Im Netz finden sich zahlreiche Anleitungen, wie man mittels einer Tilt-Plattform einen Ball balancieren kann. Mit dem vorgestellten Aufbau sollte so etwas ebenfalls möglich sein.

Dazu muss man die Position des Balls auf der Plattform bestimmen. Zwei gängige Methoden sind die Bestimmung über einen Touch-Screen [18] und die Bestimmung mittels Objekterkennung aus einem Videostream einer Kamera. Ich habe mich für letzteren Weg entschieden und das gut dokumentierte Projekt von Johan Link als Basis genommen [5]. Die Positions-Steuerung der Plattform übernimmt ein Arduino und die Kamera-Auswertung und Regelung ein PC mit einer USB-Videokamera. Das Projekt ist in Python geschrieben und verwendet zur Bilderkennung die OpenCV-Bibliothek. Der PC übermittelt die Stellwinkel via serieller Schnittstelle an den Arduino, der dann die Servo-Motoren ansteuert bzw. regelt.

Die in dem Programm von Johan Link verwendete dreiarmlige Plattform-Ansteuerung wurde durch die wesentlich einfachere zweiarmlige Ansteuerung ersetzt. Als Regler kommt bei dem Projekt wieder ein – diesmal zweidimensionaler – PID-Regler zum Einsatz. Dieser Regler ist ein TITO-Regler (*Two Inputs Two Outputs*), wobei man eigentlich von zwei unabhängigen PID-Reglern für jede Richtung sprechen kann.

Als Input dienen die mittels OpenCV ermittelten Ballpositionen und die vom Benutzer ausgewählte Soll-Position. Die Ausgangswerte des Reglers werden in Stellwinkel umgerechnet und an den Arduino übermittelt.

Die grafische Oberfläche ist mit Tkinter realisiert und erlaubt die Einstellung der Regelparameter über ein grafisches User-Interface (Abb. 12).

Zur Eichung der Winkelwerte habe ich eine Wasserwaagen-Handy-App verwendet. Die Werte habe ich mir dann mit Gnuplot ausgeben lassen und eine Eichgerade hineingelegt, die ich dann im Arduino hinterlegt habe, so dass die Winkelwerte möglichst genau den echten Winkeln in Grad entsprechen. Die Eichung ist aber eigentlich nicht nötig, da die Stellwerte aus dem PID-Controller mit einem experimentell bestimmten proportionalen Parameter in Winkelwerte umgerechnet werden.

Mit Middle-Klick auf den Ball im Camera-Feed wird dem Programm die Farbe des Balls mitgeteilt; das Programm übernimmt dann das Tracking des Balls und umrandet den erkannten Ball mit einem gelben Kreis. Mit Links-Klick kann man die Zielposition auf der Plattform auswählen, auf die der Ball dann gesteuert wird. Es dauert eine Weile, bis der Ball die Zielposition erreicht hat. In [1] ist ein Video des Aufbaus in Aktion zu sehen.

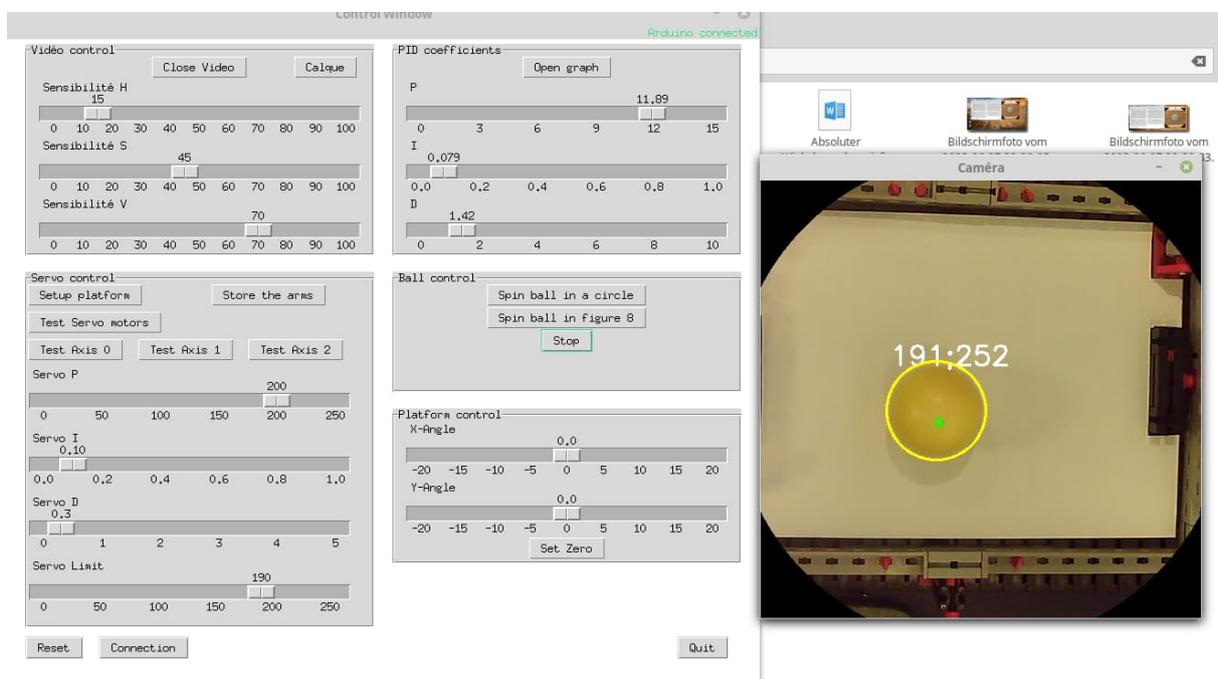


Abb. 12: Benutzeroberfläche des angepassten Python-Steuerprogramms von Johan Link zum Balancieren eines Balls auf der Tilt-Plattform. Links sieht man das UI-Panel zum Eingeben der Regelparameter, rechts das Bild der USB-Kamera. Das Steuerprogramm wird durch Middle-Klick auf den Ball „angesetzt“, mit Links-Klick kann man die Zielposition (grüner Kreis) setzen. Das Regelprogramm versucht dann, den Ball auf dieser Position zu halten. Die PID-Regelparameter kann man über die Schieberegler während des Betriebs verändern.

Ausblick

In diesem Beitrag habe ich beschrieben, wie man mit fischertechnik eine 2D-Kipp-Plattform ansteuern und damit einen Ball balancieren kann: entweder von Hand als analoges Geduldsspiel oder mittels Servos bzw. modifizierten fischertechnik-Encoder-Motoren über einen Computer. Es wäre gut, wenn fischertechnik 9V-Motoren mit Quadratur-Ausgang anbieten würde.

Ich denke, dass Optimierungen der Regelkreise oder die Verwendung von anderen Reglertypen möglich sind. Auch andere Plattform-Mechanismen wie die von Johan Link vorgestellte dreiarmlige Ansteuerung sind denkbar. In Abb. 13 ist ein erstes Modell zu sehen, das auf weitere spannende Experimente wartet.

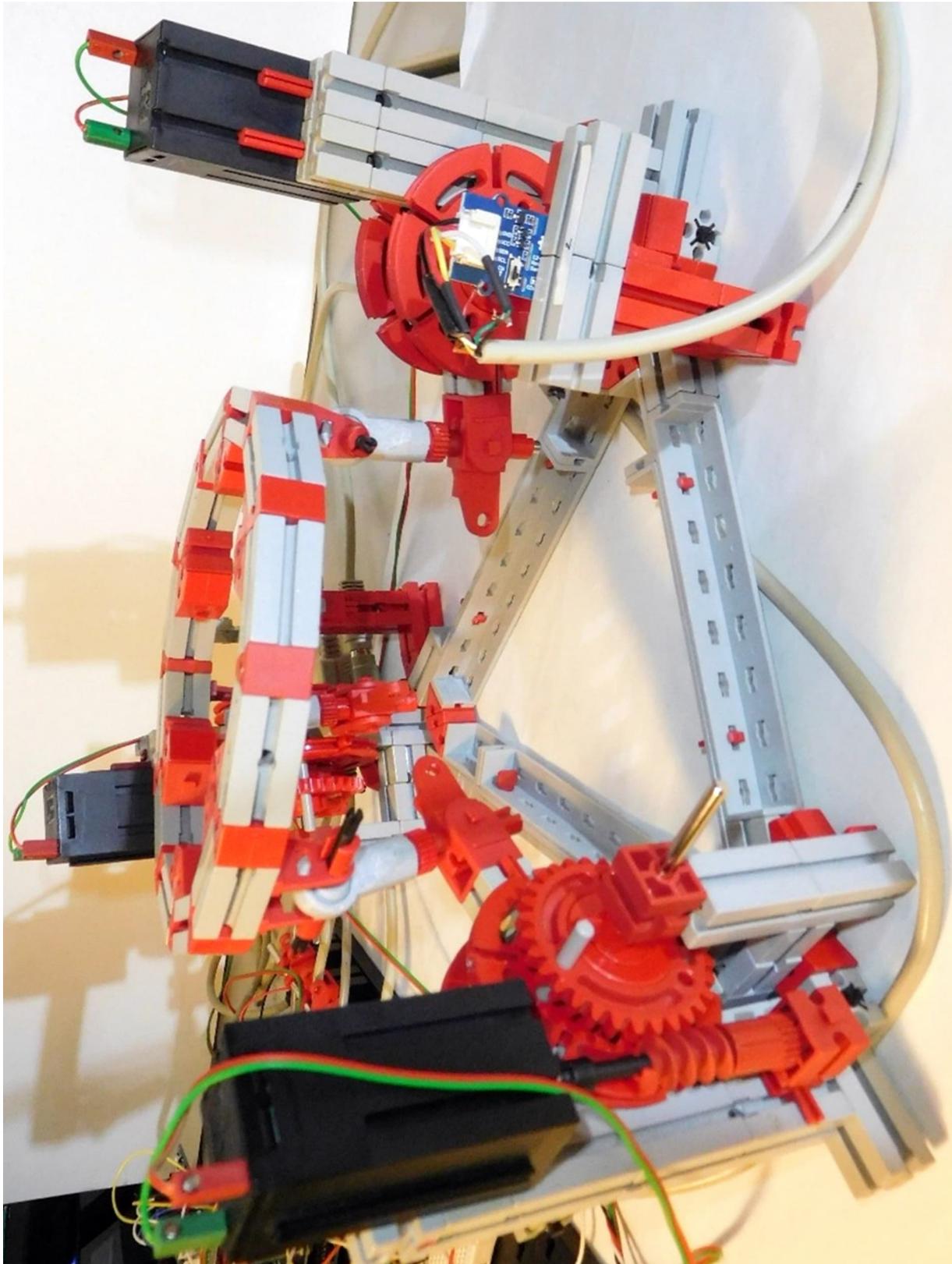
Darüber hinaus wäre es interessant, ob man das Balancier-Programm auf einem TXT-Controller zum Laufen bringen kann.

Und wer sich für die Physik des Balls auf einer Platte interessiert, dem sei der sehr ausführliche Bericht [11] empfohlen, in dem auch die Bewegungsgleichungen hergeleitet werden.

Referenzen

Die Programme finden sich auf der [Download-Seite zu dieser Ausgabe](#) der ft:pedia.

- [1] [Video der fischertechnik-Balancier-Plattform](#), Youtube.com
- [2] [embedded World Conference](#)
- [3] Wikipedia: [Marble Madness](#)
- [4] Dirk Fox: *Kardanische Aufhängung*. [ft:pedia 1/2023](#), S. 70–73.
- [5] Johan Link: [Ball-Balancing-PID-System](#), Github Repository
- [6] Florian Bauer: [fischertechnik Kugelgelenkkopf V2 \(Ball Joint\)](#), Thingiverse Repository
- [7] Florian Bauer: [fischertechnik Kugelgelenkkopf](#) (Ball Joint), Thingiverse Repository
- [8] ic-Haus: [iC-MD – 48-Bit Quadrature Counter with RS422 Receiver and SPI/BiSS Interface](#)
- [9] [AS5600-Arduino-Library](#), Github Repository
- [10] AMS: [AS5600 – 12-Bit Programmable Contactless Potentiometer](#), Datenblatt, v1.06, 20.06.2018.
- [11] Ehsan Lakzaei: [Balancing ball on a plate](#). Final Report, Technische Universität Eindhoven, 2020.
- [12] Wikipedia: [PWM](#)
- [13] Wikipedia: [PID-Controller](#)
- [14] Ashitava Ghosal: [Robotics: Advanced Concepts & Analysis, Module 4 – Kinematics of Parallel Robots](#), Indian Institute of Science, Bangalore, 2010.
- [15] Himalayasolution: [L293D motor control shield](#)
- [16] Gabriela Mamani, Jonathan Beccedas, Vicente Feliu Batlle: [Robust Position Control of a DC Motor by Sliding Mode](#), 2010
- [17] Myo Maung Maung, Maung Maung Latt, Chaw Myat Nwe: [DC Motor Angular Position Control using PID Controller with Friction Compensation](#), International Journal of Scientific and Research Publications, Volume 8, Issue 11, 2018
- [18] Arduino Team: [Ball-on-plate machine uses touchscreen position sensing](#), 14.07.2017.



*Abb. 13 Eine Tilt-Plattform mit dreiarmer Ansteuerung.
Die Plattform hat drei Freiheitsgrade: Neben der Kippung um zwei Achsen kann sie in der Höhe
verstellt werden. Als Aktoren werden hier XM-Motoren mit AS5600-Magnet-Encodern verwendet.
Die Kugelgelenke sind als spezielle 3D-Drucke [7] realisiert.*