

Editorial

## 1010 Jahre

Zehn Jahre sind nichts im Vergleich mit der Lebenserwartung eines Grönlandhais (400 Jahre), dem Alter der Menschheit (rund 300.000 Jahre) oder gar dem unseres Planeten (ca. 4,54 Milliarden Jahre). Im Leben eines Menschen sind 3.650 Tage jedoch ein stattlicher Zeitraum.

Für Stefan und mich stehen die vergangenen zehn Jahre auch für 40 Ausgaben der ft:pedia. Oder, anders gezählt: für knapp 2.650 Seiten geballtes fischertechnik-Wissen, verfasst mit Herzblut und Hingabe von Fans für Fans. Die Zahl der ft:pedia-Autoren ist inzwischen auf über 50 angewachsen, und der Umfang der jährlich vier ft:pedia-Ausgaben hat 2020 ein „All-Time-High“ von 375 Seiten erreicht. Auch die Nachfrage ist überwältigend: Über eine halbe Million Downloads, im Mittel also 52.000 pro Jahr.

An eine solche Erfolgsgeschichte war eigentlich nicht zu denken, als Stefan und ich uns am 11.01.2011 zum ersten Mal trafen. Als nach kürzester Anwärmphase unsere Kindheits-Schlüsselerlebnisse mit fischertechnik auflebten, wir begannen, unserer Begeisterung für das fischertechnik-System hemmungslos freien Lauf zu lassen und dabei – vielleicht ein klein wenig nostalgisch-verklärt – das gesunkene Niveau der Baukästen sowie deren zurückgehende Verbreitung in den Kinderzimmern des 21. Jahrhunderts beklagten, geschah zweierlei: Während von uns vollständig unbemerkt die Stunden verrannen (was später bei unseren Familien nur auf begrenztes Verständnis stieß), nahm gleichzeitig eine verwegene Idee konkrete Gestalt an.

Dirk Fox, Stefan Falk

Wie wäre es, etwas wie die alten Clubhefte (der 70er Jahre...) mit ihren Modellen zum Nachbauen, praktischen Tipps & Tricks und Hintergrundwissen mit Unterstützung einiger aktiver Fans wieder aufleben zu lassen – und damit zu zeigen, was mit fischertechnik wirklich möglich ist? Den Konstrukteuren der herausragenden, aber flüchtigen Modelle auf den Conventions so eine Plattform für deren Dokumentation und zugleich technikbegeisterten jungen Menschen sowohl elementares Technikwissen zum Nachschlagen als auch Ideen und Anreize für eigene kreative technische Konstruktionen zu bieten?

Gedacht, getan. Der Name war schnell gefunden – enzyklopädisch sollte das sein, was wir zum Leben erwecken wollten. Mondän. Universell. Natürlich mit ISSN, archiviert in der Deutschen Nationalbibliothek. Und pünktlich – viermal jährlich, am letzten Samstag des Quartals.

Die Dokumentenvorlage stand am 12. Januar 2011, die ersten Autorenanfragen gingen an den folgenden Tagen heraus. Wenig später erhielten wir unsere ISSN 2192-5879. Und am 26.03.2011 erschien die erste Ausgabe: Sechs Beiträge, 36 Seiten.

Alles Weitere ist Geschichte. Helft mit, dass die ft:pedia noch eine lange vor sich hat.

Beste Grüße,  
Euer ft:pedia-Team

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

## Inhalt

1010 Jahre.....	2
Inhaltsverzeichnis der ersten 10 Jahre ft:pedia .....	4
Sonderbeilage: Alle Mini-Modelle im Überblick .....	5
Mini-Modelle (26): Zentrierwinkel .....	6
Wendefahrzeug Wennichanstoßewendeich .....	8
Eine dreizeigrige Uhr .....	13
Pneukran .....	20
Schrägseilbrücke und Hängebahn (Teil 2) .....	26
Das Mini-Speichenrad.....	37
Richtungsanzeige mit LEDs .....	40
Planetengetriebe (2) .....	42
Uhr mit Planetengetriebe .....	46
Eingänge vermehren.....	55
fischertechnik-Roboter mit Arduino (Teil 2): 2,4-GHz- Fernbedienung mit dem PS2-Gamepad.....	58
Bluetooth-Modul für den ftDuino.....	64
ft-Soccer – IR-Sensor für den RoboCup-Junior-Wettbewerb „Soccer“ .....	68
Rollende Steine – die ftcSoundBar .....	73
Elektronik-Module (Teil 5) .....	80

## Impressum

<http://www.ftpedia.de>

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

**Autoren:** Christian Bergschneider, Gerhard Birkenstock, Daniel Canonica, Arnoud van Delden, Stefan Falk, Dirk Fox, Hans-Christian Funke, Stefan Fuss, Ralf Geerken, Björn Gundermann, Till Harbaum, Christian Lauff, Kurt Mexner, Rüdiger Riedel, Tilo Rust, Oliver Schmiel, Harald Steinhaus.

**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.

In eigener Sache

## Inhaltsverzeichnis der ersten 10 Jahre ft:pedia

Stefan Falk

*Die erste Dekade ft:pedia ist „voll“. Insbesondere für Liebhaber der von Rüdiger Riedel gedruckten Bände mit vollständigen Doppeljahrgängen bieten wir zum Ausdruck auch eine Sonderbeilage mit einem Gesamt-Inhaltsverzeichnis [1] über die ersten 10 Jahre ft:pedia zum Download.*

Dirk Fox und ich erinnern uns immer wieder gerne an unser Zusammentreffen Anfang 2011 in seinem Büro. Wir waren – wie andere fischertechniker auch – etwas unglücklich darüber, dass der zentrale Eckpfeiler der fischertechnik-Welt – das Vermitteln von technischem Wissen – viel zu sehr im Hintergrund schlummerte. Und es war völlig klar, dass das so nicht bleiben konnte und sollte, und dass wir das ändern werden. Jetzt. Aufschieben war keine Option mehr.

Viele Ideen schwebten durch die von Begeisterung und Schwärmen getränkte Büroluft: Sollen wir Workshops für Kinder machen? Sollten wir Geld in die Hand nehmen, Schulen ausstatten und Lehrer schulen? Sollten wir freie Räume mieten und darin Workshops für jedermann (jedes Geschlechts, natürlich) anbieten und bewerben?

Das ist mittlerweile – und nur Dank Dirks schier unglaublicher Wirbelwindigkeit in dieser Sache – tatsächlich Realität geworden. Workshops für Lehrer machen diese fit für unzählige fischertechnik-AGs in vielen Schulen. Sponsoren und die fischertechnik GmbH ermöglichen die Ausstattung dieser Schulen mit reichhaltigem Konstruktionsmaterial. In MINT-Ferientagen, Wettbewerben und anderen Veranstaltungen zeigen die Kinder und Jugendlichen stolz, was sie mit all dem erschufen – und legen

so oftmals den Grundstein für das spätere Ergreifen eines technischen Berufs.

Die erste „Maßnahme“ zur Beseitigung der uns schmerzenden Umstände war aber, wie einige Leser der Erstausgabe ganz richtig vermuteten, den Geist der 1970er Clubhefte und hobby-Begleitbücher auf aktuellem Stand der fischertechnik wieder aufleben zu lassen. Und zwar in Reinkultur, ohne Beiwerk.

So wurden die ersten Autoren für den Plan begeistert, die ersten Artikel geschrieben, redigiert, gesetzt – und im März 2011 unter bis dahin strengster Geheimhaltung als ft:pedia der fischertechnik-Welt präsentiert. Die blieb uns und der immer noch wachsenden Autorenschar bis heute treu und belohnt uns Quartal für Quartal mit vielen Downloads aller Ausgaben und manchmal auch mit Feedback, Kritik und Anregungen...

... wahrlich Ansporn also für die nächsten vierzig Ausgaben mit fast fünfhundert Beiträgen auf zweieinhalbtausend Seiten.

### Quellen

- [1] Stefan Falk: *Sonderbeilage Gesamt-Inhaltsverzeichnis 2011-2020*. Im [Download-Bereich](#) der ft Community, 2020.

In eigener Sache

## Sonderbeilage: Alle Mini-Modelle im Überblick

Rüdiger Riedel

*Viele, viele kleine Modelle sind in den letzten 10 Jahren von fleißigen und kreativen Fans der fischertechnik-Welt für die ft:pedia erfunden worden. Jetzt werden sie zusammenfassend präsentiert.*

Es braucht so wenig, um einem Kind Freude zu bereiten. Mitbringsel für die Kinder oder Enkel müssen nicht mit Schokolade versüßt sein! Ein kleines fischertechnik-Püppchen, ein Traktor, ein Roboter mädchen zum Zusammenbauen, ein Hubschrauber, das erfreut alle (Schul-)Kinder und zeigt ihnen, wie einfach diese Modelle aufgebaut sind.

Die Konkurrenz aus Fernsehen, Internet und Computerspielen lässt unseren Kindern und Enkeln immer noch genug Raum für diese kleinen Dinge, für kreative Technik.



Abb. 1: Da geht's lang

Das fischertechnik-Püppchen zeigt uns den Weg zu allen bisher gebauten Mini-Modellen der ft:pedia und sagt:

„Ich bin nicht allein!“

Alle Mini-Modelle werden in einer Sonderbeilage [1] noch einmal originalgetreu vorgestellt für alle, die Freude haben am Kleinen und Pfiffigen.

Übrigens: Das fischertechnik-Püppchen in Abb. 1 zeigt uns nicht nur den Weg zu vielen Mini-Modellen, sondern auch die Nord-Süd-Richtung. Die Achse 110 ([31031](#)) trägt zwei Bausteine 15 Magnet ([108278](#)). Im Kopf ist ein Stabmagnet 4 · 10 mm versteckt. Ein zweiter im Hut hält es an einem Perlonfaden, damit es sich frei drehen kann.

### Quellen

- [1] Rüdiger Riedel: *Sonderbeilage Mini-Modelle*. Im [Download-Bereich](#) der ft Community, 2020.

Modell

## Mini-Modelle (26): Zentrierwinkel

Gerhard Birkenstock

*Eine große Scheibe soll auf die Drehmaschine. Für die einzuspannende Achse muss der Mittelpunkt gefunden werden. Kein Zentrierwinkel zur Hand? Mit fischertechnik ist das Problem in wenigen Minuten gelöst.*

### Grundlagen

Legt man in ein Prisma einen runden Gegenstand, wird dieser sich immer mittig ausrichten.

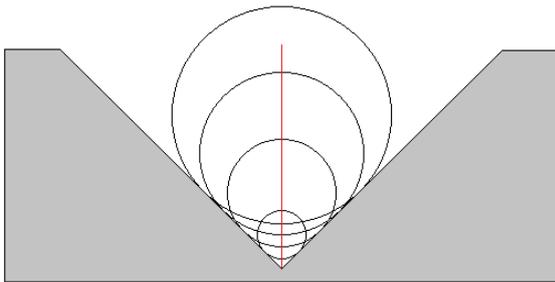


Abb. 1: Prinzip eines Prismas

Da der Radius sich rechts und links immer im gleichen Abstand ablegt, kommt der Mittelpunkt des runden Körpers auf einer Senkrechten zu liegen. Dies ist mit einer roten Linie in Abb. 1 dargestellt.

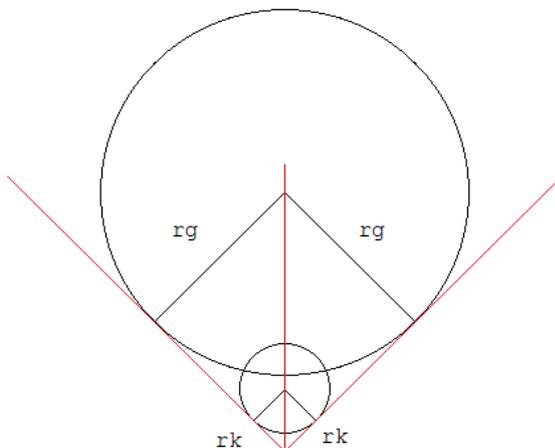


Abb. 2: Schemazeichnung nach Abb. 1

Schematisch sieht das so wie in Abb. 2 dargestellt aus. Es entsteht eine Winkelhalbierende. Auch andere Winkel kleiner als  $90^\circ$  würden funktionieren. Jedoch wird das Anlegen dann unhandlicher. Ein größerer Winkel als  $90^\circ$  macht das Anlegen ungenauer. Somit nimmt man das Optimum, die  $90^\circ$ .

Mit Hilfe dieses Wissens kann man sich ein geometrisches Hilfsmittel herstellen: den so genannten Zentrierwinkel. Mit dessen Hilfe ist man in der Lage, die (rote) Mittellinie auf einem Kreis einzuzeichnen.

Beim Anlegen an den Kreis bildet der „rechte Winkel“ die Grundlage. Auf einer der Wangen setzt man mit  $45^\circ$  einen weiteren Schenkel auf. Mit diesem Schenkel wird die Winkelhalbierende auf dem Kreis eingezeichnet.

Dreht man den Winkel um ca.  $90^\circ$  auf dem Kreis und zeichnet erneut eine Linie, welche die erste schneidet, hat man den Mittelpunkt des Kreises bestimmt.

### Realisierung mit fischertechnik

Die beiden  $90^\circ$ -Wangen mit fischertechnik zu realisieren ist eine der einfachen Übungen. Die beiden Wangen lassen sich mit den Steinen BS 30 sehr einfach erstellen. Auf einer der Wangen wird der Schenkel im Winkel von  $45^\circ$  mit Winkelsteinen aufgebaut.

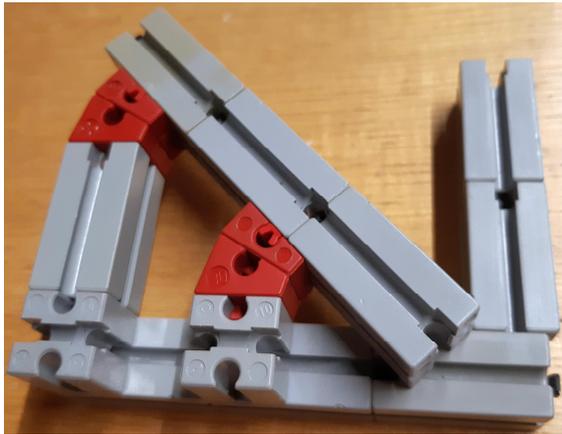


Abb. 3: Zentrierwinkel aus fischertechnik

Da die Bausteine von fischertechnik eine richtig gute Passung haben, ist die Anordnung überraschend genau. Lediglich die Dicke des Bleistiftes muss durch Verschieben der rechten Wange eingestellt werden.

In Abb. 3 ist das an der rechten Ecke gut zu erkennen.

Alle für die Konstruktion benötigten Steine sind in Abb. 3 gut sichtbar – und in der folgenden Tabelle zusammengefasst. Viel Spaß beim Anwenden!

Bauteil-Nr.	Anz.	Bezeichnung
<a href="#">32879</a>	9	Baustein 30, schwarz
<a href="#">32881</a>	2	Baustein 15, schwarz
<a href="#">31982</a>	2	Federnocken
<a href="#">31011</a>	2	Winkelstein 30°
<a href="#">31981</a>	2	Winkelstein 15°

Modell

## Wendefahrzeug Wennichanstoßewendeich

Rüdiger Riedel

*Vor 60 Jahren waren Batterien teuer, aber manche Kinder hatten Großeltern, die sie ihnen kauften. Damit wurden seltsame Autos und schienenlose, große Lokomotiven angetrieben, die bei jedem Anstoß an der Wand und an Möbeln die Fahrtrichtung wechselten und weiterfuhren. Die Funktion war sehr geheimnisvoll.*

### Das Wendegeheimnis

Bei meinem Freund Ronald fuhr eine Westernlokomotive durchs Wohnzimmer, hin und her, ohne anzuhalten von Wand zu Schrank und Stuhlbein. Wie das wohl funktionierte? Zu meinem Glück hatte er viele Spielsachen und manches war auch kaputt in seiner Spielkiste. Und so konnte ich das Geheimnis um die Wendefunktion lüften.

Es ist prinzipiell ein Dreirad mit Antrieb auf das Einzelrad. Die Antriebsachse kommt von oben durch das Chassis. Ein Ritzel treibt ein Kronrad, das auf der Achse des Antriebsrades sitzt. Die Lagerung des Rades ist vom Chassis entkoppelt, es kann sich in alle Richtungen drehen. Wird das Rad blockiert, dreht sich der Drehschemel so lange, bis das Rad wieder antreiben kann.

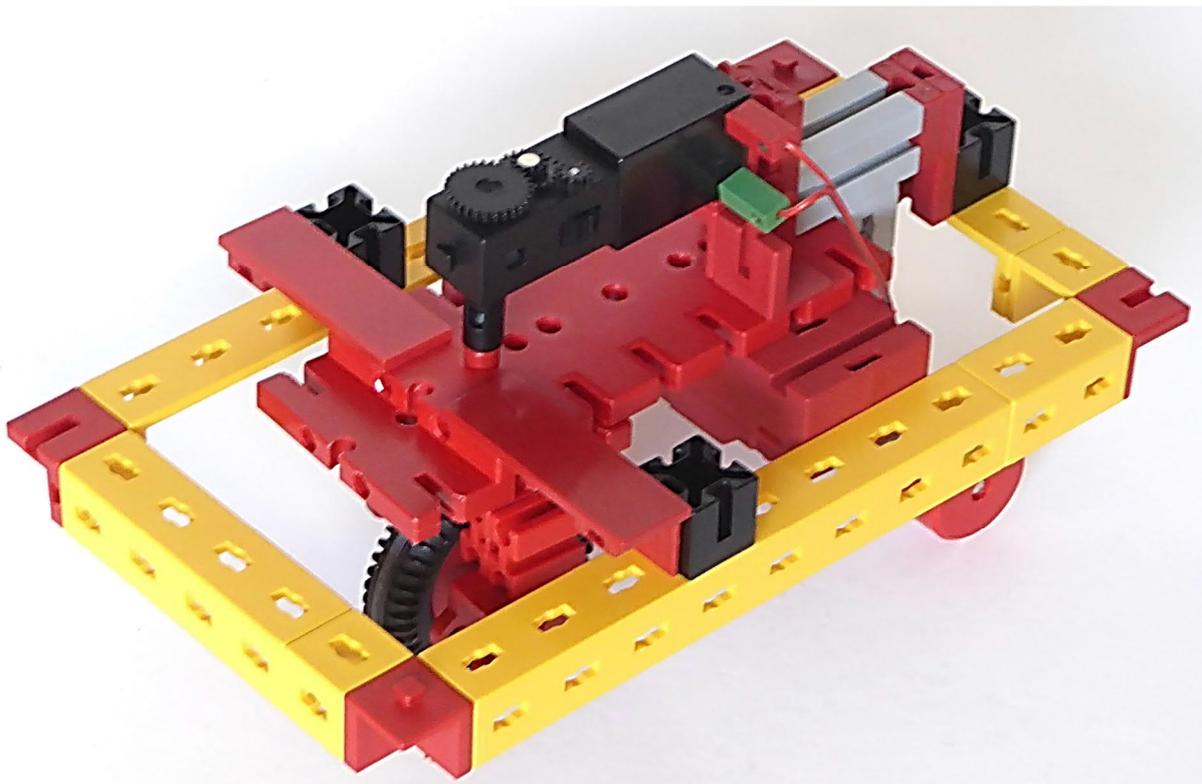


Abb. 1: Wennichanstoßewendeich

Von fischertechnik gibt es das Kronrad auf dem Zahnrad Z40-32 ([31022](#)), aber das ist recht groß und ein Antrieb mit einem Z10 braucht viel Platz.

Die „Kaulquappe“ von Harald Steinhaus in [1], Abb. 5, zeigt eine bessere Möglichkeit: Ein Rastkegelzahnrad ([35061](#)) treibt einen Reifen 45 · 12 ([31018](#)) mit Gummiring 45 ([36051](#)), siehe Abb. 2.

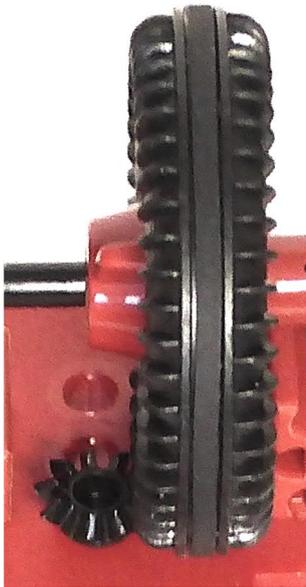


Abb. 2: Kegelzahnrad und Reifen 45

Der Drehschemel wird nicht mittig gelagert. Wir nehmen ein Loch weiter außen, um eine gewisse Geradeauslauf-Eigenschaft zu erreichen.

## Bauanleitung

Die Bauanleitung des Wendefahrzeugs lehnt sich an die Bilderrätsel einer bekannten Möbelfirma an und sollte somit leicht nachvollziehbar sein.

Die Verbindung zwischen Chassis (Grundplatte 90 · 45) und Drehschemel (Grundplatte 45 · 45) wird nur durch die Achse des Rast-Kegelzahnrades hergestellt.

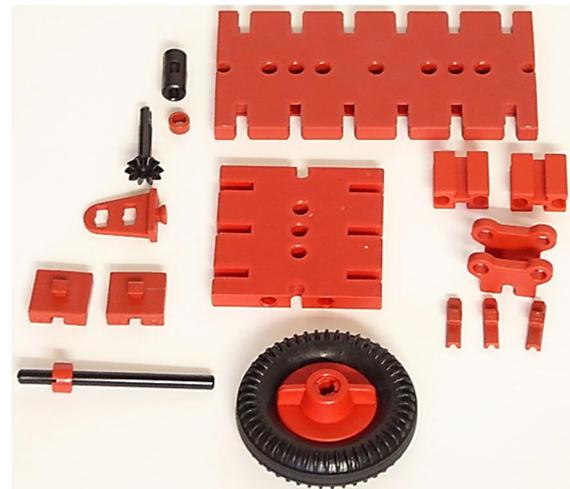


Abb. 3: Der Anfang

Teile in Abb. 3:

Nr.	Bezeichnung	Anzahl
<a href="#">36576</a>	Grundplatte 90 · 45	1
<a href="#">36593</a>	Grundplatte 45 · 45	1
<a href="#">31982</a>	Federnocken	3
<a href="#">37468</a>	Baustein 7,5	2
<a href="#">37237</a>	Baustein 5	2
<a href="#">38253</a>	S-Kupplung 15 2	1
<a href="#">32085</a>	Rollenbock 15	1
<a href="#">35061</a>	Rast-Kegelzahnrad Z10 m1	1
<a href="#">31597</a>	Abstandsring 3	1
<a href="#">35073</a>	Rastkupplung 15	1
<a href="#">38414</a>	Kunststoffachse 40	1
<a href="#">37679</a>	Klemmbuchse 5	1
<a href="#">31018</a>	Reifen 45 · 12	1
<a href="#">36051</a>	Gummiring 45 37 · 2,5	1
<a href="#">31015</a>	Flachnabe 25 komplett	1



Abb. 4: Schritt 2

Durch Verschieben der Klemmbuchse auf der Achse sowie der S-Kupplung und des Rollenbocks greift das Kegelzahnrad sauber in die Zähne des Reifens ein.

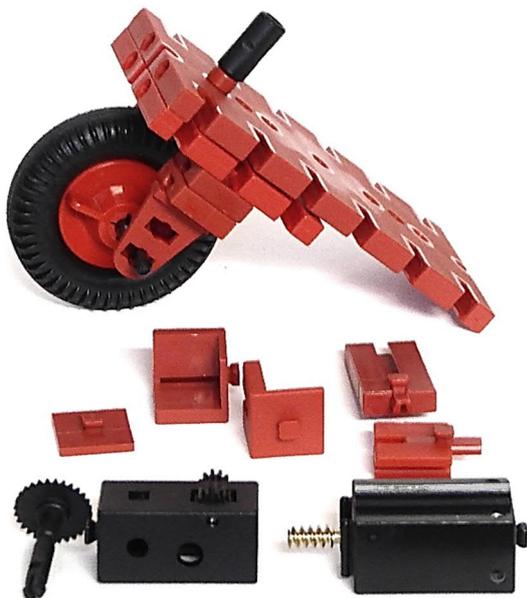


Abb. 5: Schritt 3

Weitere Teile in Abb. 5:

Nr.	Bezeichnung	Anzahl
<a href="#">38246</a>	Bauplatte 15 · 15 1Z	1
<a href="#">38240</a>	V-Baustein 15 Eck	2
<a href="#">31982</a>	Federnocken	2
<a href="#">35049</a>	Baustein 5 15 · 30	1
<a href="#">37468</a>	Baustein 7,5	1
<a href="#">137096</a>	XS-Motor	1
<a href="#">31078</a>	U-Getriebe m0,5 mini	1
<a href="#">31082</a>	Rastachse + Zahnrad Z28	1

Für die waagerechte Ausrichtung ruht der Motor auf einem V-Baustein 15 Eck mit darauf liegender Bauplatte 15 · 15 1Z.

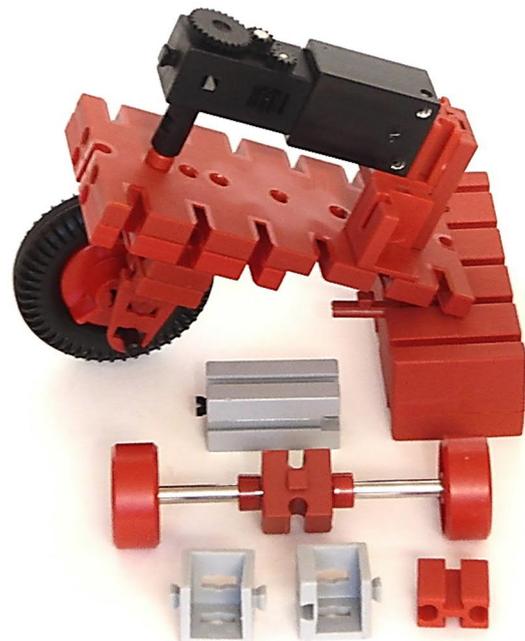


Abb. 6: Schritt 4

Die zusätzlichen Teile in Abb. 6:

Nr.	Bezeichnung	Anzahl
<a href="#">135719</a>	Batteriekasten	1
<a href="#">31982</a>	Federnocken	2
<a href="#">31003</a>	Baustein 30	1
<a href="#">37468</a>	Baustein 7,5	1
<a href="#">31040</a>	Metallachse 90	1
<a href="#">37679</a>	Klemmbuchse 5	2
<a href="#">32064</a>	Baustein 15 Bohrung	1
<a href="#">36581</a>	V-Rad 23 · 10	2
<a href="#">36300</a>	S-Winkelträger 15	1
<a href="#">36304</a>	S-Winkelträger 15 2Z	1

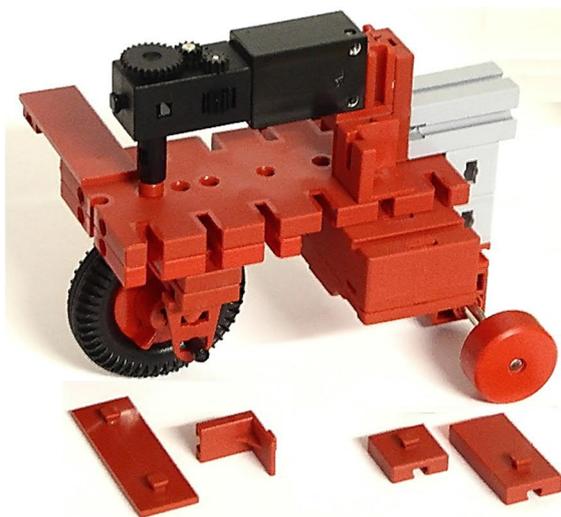


Abb. 7: Schritt 5

Zusätzliche Teile in Abb. 7:

Nr.	Bezeichnung	Anzahl
<a href="#">38423</a>	Winkelstein 10 · 15 · 15 rot	2
<a href="#">38242</a>	Bauplatte 15 · 45 2Z	2
<a href="#">37237</a>	Baustein 5	1
<a href="#">35049</a>	Baustein 5 15 · 30	1

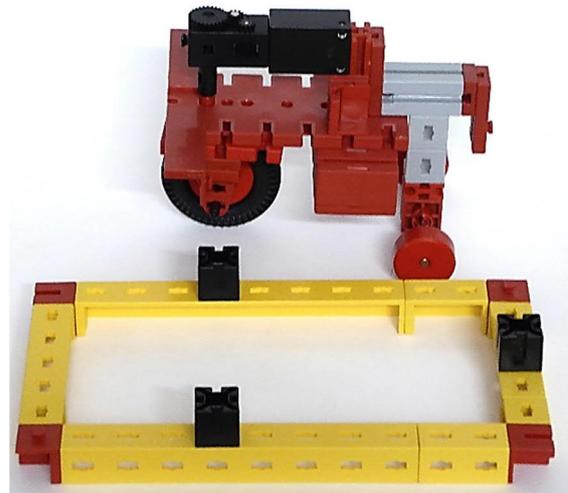


Abb. 8: Schritt 6

Zum Schluss die Teile für den Rahmen:

Nr.	Bezeichnung	Anzahl
<a href="#">38240</a>	V-Baustein 15 Eck	4
<a href="#">36294</a>	S-Winkelträger 120	2
<a href="#">36297</a>	S-Winkelträger 60	2
<a href="#">36299</a>	S-Winkelträger 30	2
<a href="#">35053</a>	S-Winkelträger 15	2
<a href="#">32850</a>	S-Riegelstein 15 · 15	3

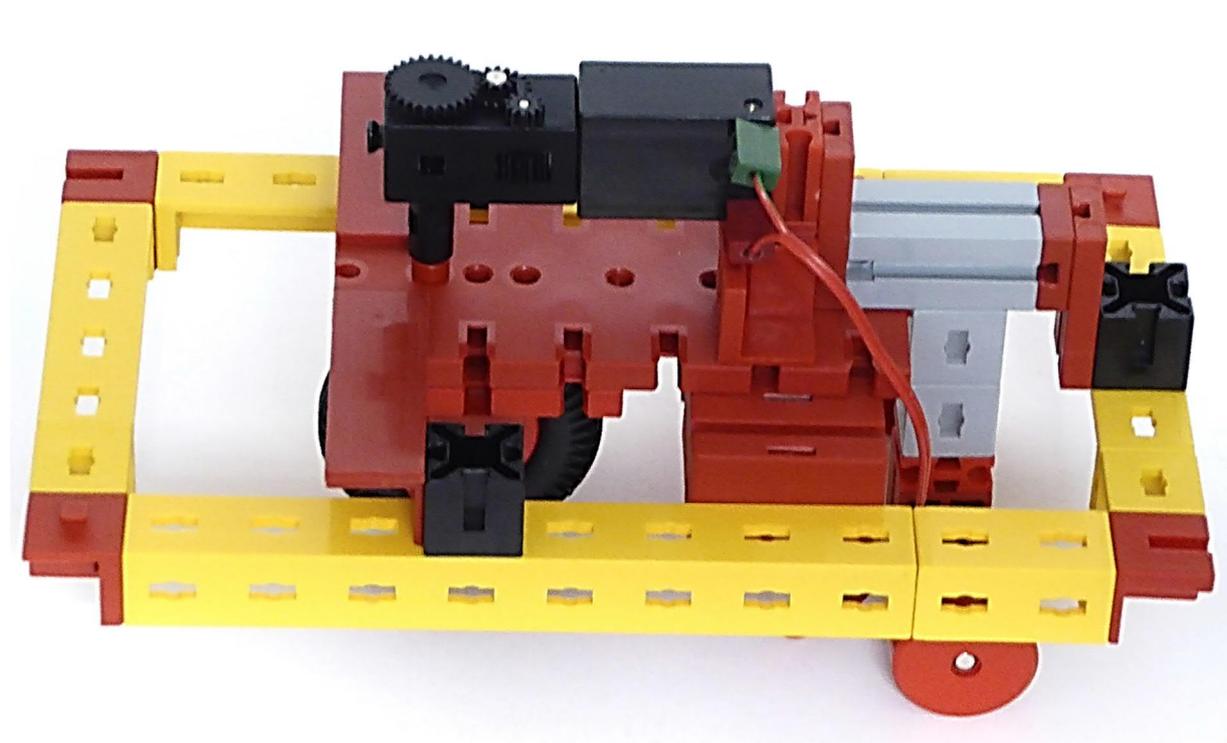
Jetzt noch den Motor mit dem Batteriekasten verbinden, Batterie oder Akku einsetzen und es kann losgehen. Ob der Motor rechts- oder linksherum dreht, ist nicht egal; es kann unterschiedliche Fahreigenschaften ergeben. Probiert es aus!

Eine Variante des Rahmens seht ihr in Abb. 10, sie erinnert ein wenig an einen Fisch.

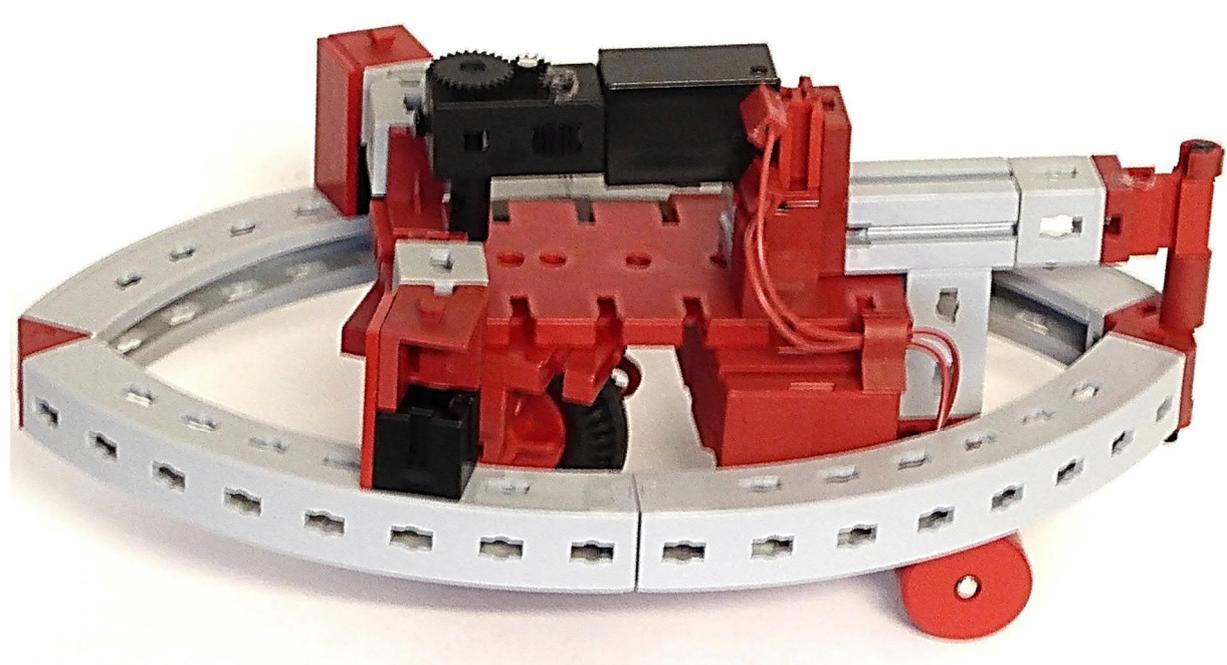
Verblüffend ist die Leichtigkeit, mit der das Fahrzeug bei einer Berührung wendet.

## Quellen

- [1] Harald Steinhaus: *Kaulquappen (Teil 11)*. ft:pedia 2/2020, [S. 12-16](#).
- [2] [Bauanleitung mot1 \(39118\)](#). Fischer-Werke, 1975.



*Abb. 9: Fertig ist das Wendefahrzeug*



*Abb. 10: Die Fischform*

Modell

## Eine dreizeigrige Uhr

Ralf Geerken

*Es ist erst ein paar Wochen her, dass ich ein mir bisher unbekanntes fischertechnik-Bauteil in den Händen hielt – nämlich das Mini-Speichenrad (35171), welches in einem eigenen Artikel [1] beschrieben ist. Es ist das Innenteil, also der Käfigdeckel, des alten Differentials (31043). Bei der Suche nach einer Anwendungsmöglichkeit stieß ich auf die Möglichkeit das Mini-Speichenrad in den neuen Differentialkäfig einzupassen. Es entstand dann recht schnell die Idee eine Uhr mit drei Zeigern zu konstruieren.*



Abb. 1: Die dreizeigrige Uhr

Das Geheimnis dieser Uhr ist natürlich, wie die drei Zeiger auf einer Achse montiert werden konnten, und es liegt in diesen Teilen:



Abb. 2: Die Teile der Minutenachse

Vor einem Nachbau empfiehlt es sich aber, alle Bilder dieses Beitrags zu betrachten.

Bevor das Differential-Z20 auf den Käfig gerastet wird, darf der Stundenzeiger nicht vergessen werden. Also bitte erst die Drehscheibe aufsetzen (siehe Abb. 15) und dann das Z40 mit den drei Riegelscheiben als Abstandshalter aufsetzen.

Das Z20 muss unbedingt in die Nuten des Differentialkäfigs einrasten. Jetzt sollte sich der Unterbau des Stundenzeigers, also die Drehscheibe mit verbundenem Z40, sehr leicht drehen lassen.

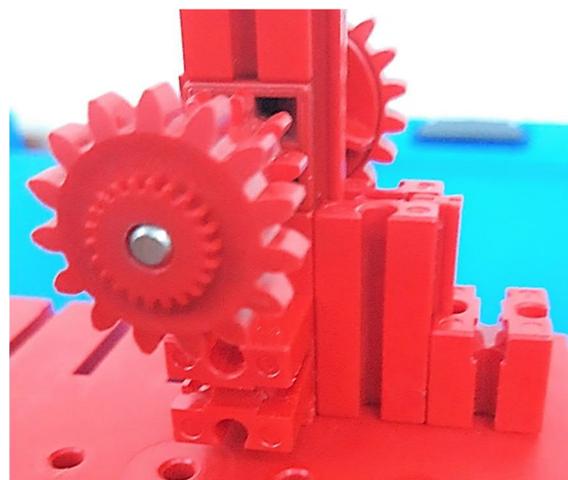
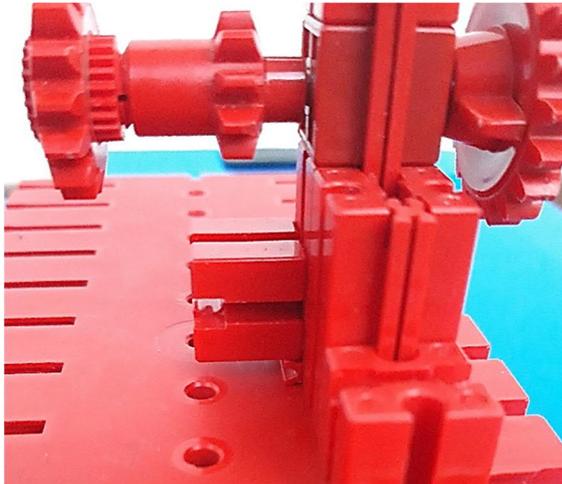


Abb. 3: Die Rückseite der Minutenübertragung

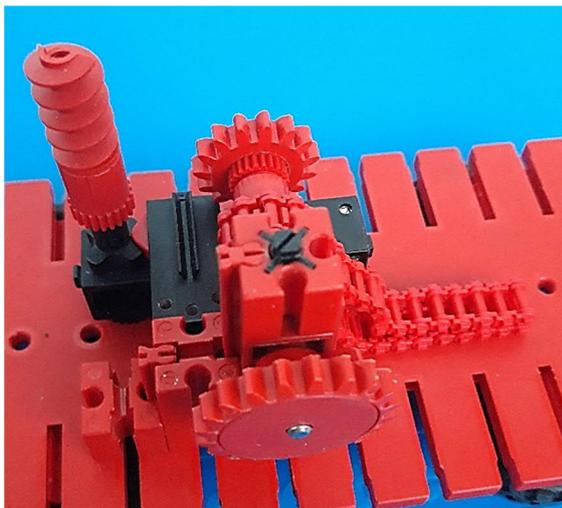
Das Rad 14 legt man danach einfach in die andere Seite des Käfigs hinein und drückt dann mit relativ viel Kraft das Mini-Speichenrad hinterher dazu. Das Rad 14 dient hierbei als Abstandshalter, damit das

Speichenrad nicht zu weit nach innen gedrückt werden kann.

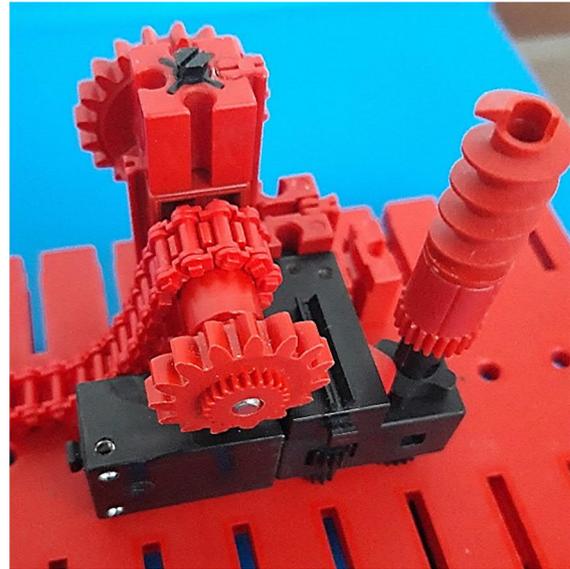


*Abb. 4: Die Minutenübertragung von der Seite*

Das Z15 und das Z20 schließen ziemlich genau mit der Metallachse 60 ab. Hierbei kommt es nicht auf ein Zehntel-Millimeter an, sondern eher darauf, dass die Achse frei drehen kann. Am linken Baustein 15 in Abb. 4 ist ein Federnocken zu sehen. Dieser dient dazu, dass man den Baustein 15 nicht zu tief montiert. Ansonsten würden nämlich die Zahnräder des U-Getriebes auf der Grundplatte schleifen.



*Abb. 5: Die Vorderseite der Minutenübertragung*



*Abb. 6: Die Rückseite der Minutenübertragung mit Motor*



*Abb. 7: Die Stundenübertragung*

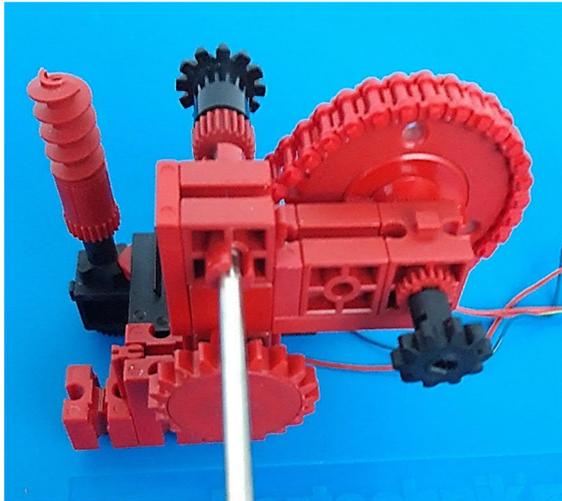


Abb. 8: Sekunden- und Minutenachse

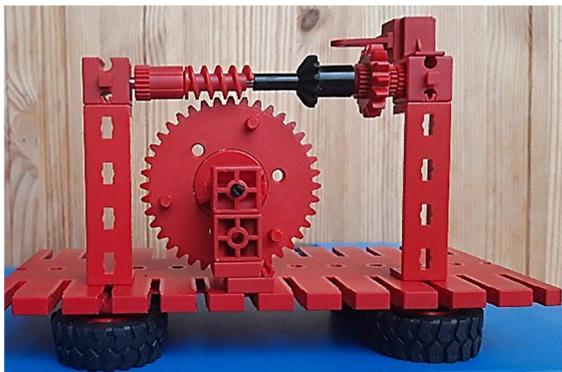


Abb. 9: Die Minuten-Untersetzung Rückseite

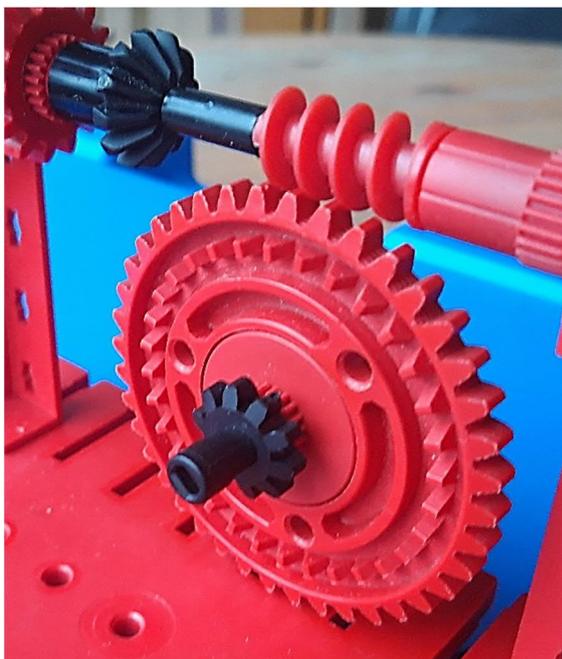


Abb. 10: Die Minuten-Untersetzung Innenseite

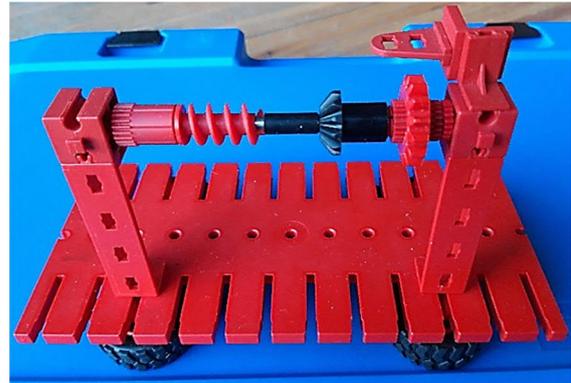


Abb. 11: Die Minutenuntersetzung nur die Achse

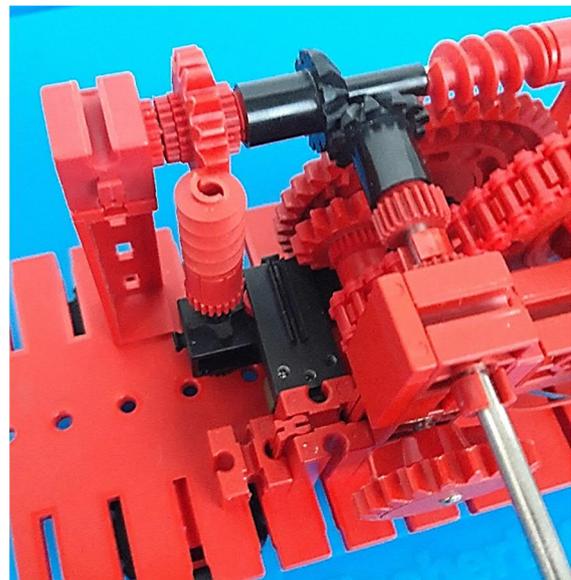


Abb. 12: Die Sekunden-Antriebseinheit von oben

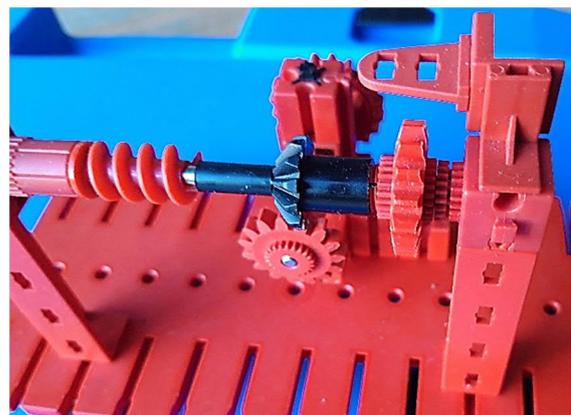


Abb. 13: Die Minutenuntersetzung nur die Achse

In Abb. 13 gibt es die Besonderheit zu beachten, dass die beiden Riegelscheiben rechts neben dem Z15 mit ihren Noppen

zueinander gekehrt sind. Damit sind sie etwas schmaler und der Abstand passt besser zum anderen Kegelzahnrad.

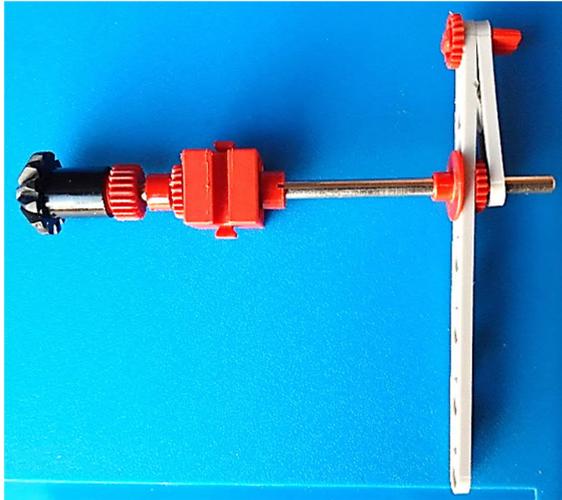


Abb. 14: Der Sekundenzeiger mit Achse



Abb. 15: Stunde Minute Sekunde frei drehbar

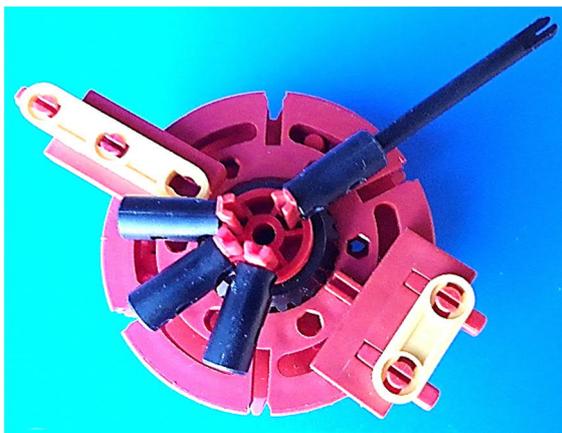


Abb. 16: Stunden- und Minutenzeiger

Bei der Konstruktion der Zeiger muss man darauf achten, dass sie sich in Waage befinden. Deswegen ist der kurze Teil der Zeiger bei fast allen Uhren zwar kürzer, aber viel dicker. Ansonsten passiert es immer wieder, dass die Zeiger an einer bestimmten Stelle hängen bleiben oder auch plötzlich kippen, weil das Übergewicht dann das Spiel in den antreibenden Zahnrädern herausdrückt.

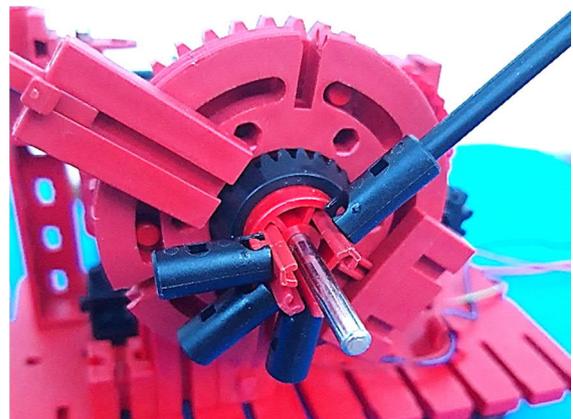


Abb. 17: Stunden- und Minutenzeiger



Abb. 18: Die drei Zeiger

Die Uhr lässt sich leicht stellen, wenn man den links oben in Abb. 19 sichtbaren Baustein 15 mit Bohrung ganz an den Rand des Federnockens zieht – gerade soweit, dass er noch von ihm gehalten wird. Dann können nämlich das Z40 und die beiden Kegelzahnräder frei drehen. Der Minutenzeiger

kann dann von Hand auf die korrekte Uhrzeit eingestellt werden, wobei sich der Stundenzeiger mit dreht.

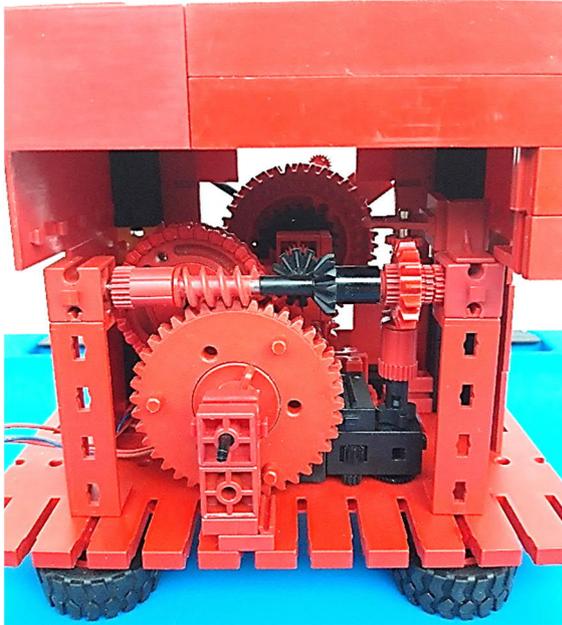


Abb. 19: Die Rückseite

Auf die Beschreibung des Gehäuses bzw. des Zifferblattes verzichte ich. Diese Teile sind auch nicht in der Teileliste aufgeführt. Hier kann jeder seiner Fantasie freien Lauf lassen. Das eckige Gehäuse mit Zifferblatt ist lediglich am Federnocken des vorderen Baustein 30 befestigt. Man kann es jederzeit austauschen gegen ein neues. Dann sieht die Uhr ganz anders aus – tickt aber noch genauso.

Der Motor wird über ein sehr einfach gehaltenes ROBOPRO-Programm jede Sekunde um 1/4 Umdrehung weitergedreht. Er dreht so lange, bis der Mini-Taster einmal gedrückt und wieder freigegeben wird, also einen 1-0-Sprung an I1 des Controllers verursacht hat. Er stoppt dann so lange, bis er wieder seinen Sekundentakt erhält. Die Uhr hört sich damit an, wie eine alte Küchenuhr (und ist leider auch genauso laut).

Mit einem geeigneten Sketch ist die Uhr sicherlich auch über einen ftDuino zu betreiben. Wie gesagt, das Programm ist sehr einfach gehalten.

Die Teileliste:

Artikel	Bezeichnung	Anzahl
<a href="#">31031</a>	Achse 110, Metall	2
<a href="#">31034</a>	Achse 30, Metall	1
<a href="#">31032</a>	Achse 60, Metall	1
<a href="#">31124</a>	Aufnahmeachse, rot	4
<a href="#">32330</a>	Bauplatte 3,75 · 15 · 30, mit 1 Langnut	1
<a href="#">38428</a>	Bauplatte 5 · 15 · 30, mit 3 Teilnuten	2
<a href="#">116252</a>	Baustein 15, neurot	3
<a href="#">32064</a>	Baustein 15 mit Bohrung	8
<a href="#">116251</a>	Baustein 30, neurot	2
<a href="#">37237</a>	Baustein 5, rot	3
<a href="#">31411</a>	Differentialkäfig, schwarz	1
<a href="#">31019</a>	Drehscheibe 60, rot	1
<a href="#">38240</a>	Eckstein 15, rot	1
<a href="#">31982</a>	Federnocken	11
<a href="#">31015</a>	Flachnabe, einflügelig komplett	3
<a href="#">31001</a>	Grundplatte, rot 180 · 90	1
<a href="#">35981</a>	Hülse mit Scheibe 15 mm, rot	1
<a href="#">37157</a>	Impulszahnrad 4	1
<a href="#">35146</a>	Kegelzahnrad, schwarz neue Bauform	2
<a href="#">37679</a>	Klemmbuchse 5	3
<a href="#">38253</a>	Kupplungsstück zweifach, rot	1
<a href="#">36819</a>	Lagerhülse 15, schwarz	1

Artikel	Bezeichnung	Anzahl
<a href="#">35171</a>	Mini-Speichenrad, auch unter Rad 17 mm freilaufend zu finden	1
<a href="#">37783</a>	Minitaster, schwarz	1
<a href="#">137096</a> , <a href="#">31062</a>	Motor XS, Motor Classic geht auch	1
<a href="#">36573</a>	Rad 14, schwarz unbedingt freilaufend also mit größerem Innendurchmesser	1
<a href="#">36581</a>	Rad 23mm, rot	4
<a href="#">35064</a>	Rastachse 45	3
<a href="#">36227</a>	Rastadapter	4
<a href="#">36248</a>	Rastkettenglied	38
<a href="#">35945</a>	Rastritzel Z10	2
<a href="#">31082</a>	Rastritzel Z28 m0,5	1
<a href="#">142251</a>	Reifen 35, weicher Gummi	4
<a href="#">36334</a>	Riegelscheibe	11
<a href="#">36323</a>	S-Riegel 4	1
<a href="#">37858</a>	Schnecke klemmbar	2
<a href="#">35113</a>	Spannzange, rot	3
<a href="#">38544</a>	Statikstrebe 15, gelb	1
<a href="#">38538</a>	Statikstrebe 30, mit Loch gelb	1
<a href="#">31848</a>	Strebenadapter	4
<a href="#">31078</a>	U-Getriebe für Minimotor, schwarz	1
<a href="#">31060</a>	Verbindungsstück 15	1
<a href="#">31061</a>	Verbindungsstück 30, hier tut es auch ein 15er	1
<a href="#">31330</a>	Verbindungsstück 45	2
<a href="#">38423</a>	Winkelstein 10, rot	1
<a href="#">78728</a>	Winkelträger 60, rot	2

Artikel	Bezeichnung	Anzahl
<a href="#">31047</a>	Zahnrad 10, rot	1
<a href="#">35695</a>	Zahnrad 15, rot mit Verzahnung innen und außen freilaufend rot	2
<a href="#">35983</a>	Zahnrad 20, Deckel für Differentialkäfig schwarz	2
<a href="#">36264</a>	Zahnrad 30, rot	1
<a href="#">31022</a>	Zahnrad 40, rot	2
<a href="#">31915</a>	Zangenmutter	2

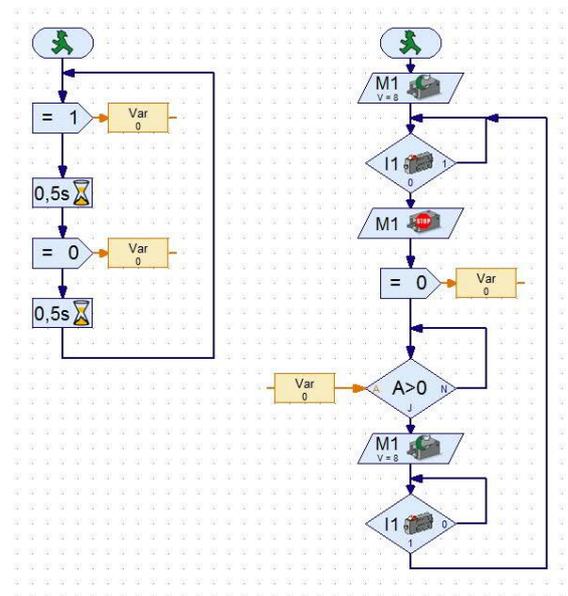


Abb. 20: Das RoboPro-Programm

Und weil's so schön war, gibt's zum Schluss nochmal dieselbe Uhr mit einem anderen Zifferblatt (Abb. 21).

## Quellen

- [1] Ralf Geerken: *Das Mini-Speichenrad*. ft:pedia 4/2020 (in dieser Ausgabe).



*Abb. 21: Die Uhr mit anderem Zifferblatt*

Modell

## Pneukran

Daniel Canonica

*Ein Pneukran vereint die Möglichkeit zum Bewegen von schweren Lasten mit Mobilität und schneller Einsatzbereitschaft. Durch ständig verbesserte Teleskoparme und modulare Gegengewichte werden enorme Höhen und Lastgrenzen erreicht.*



Abb. 1: Großer und leistungsfähiger Mobilkran [1]

Bei mobilen Kränen unterscheidet man zwischen solchen mit Raupenantrieb und straßentauglichen mit einer größeren Anzahl von Achsen.

Die eingesetzten Teleskoparme haben typischerweise etwa drei bis acht Stufen. Bei kleiner Auslenkung aus der Vertikalen können mehrere 100 Tonnen Last gehoben und bewegt werden.

### Das Modell

Dem dargestellten Modell gingen viele Prototypen und Versuche voran. Die Herausforderung beim Entwurf ist, den Teleskoparm samt Aufhängung und Drehkranz relativ leicht und doch möglichst robust und belastbar zu bauen. Abb. 2 zeigt den beinahe voll ausgefahrenen Teleskoparm.

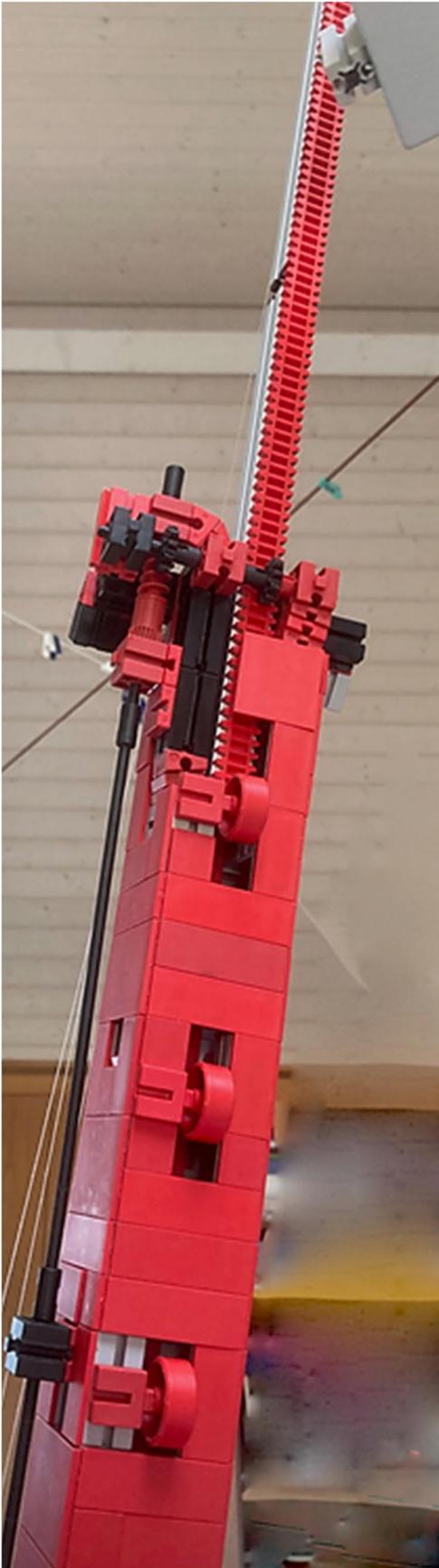


Abb. 2: Teleskoparm

Der Auszug besteht aus zwei Alus mit je 21 cm Länge. Auf der Unterseite wird er über Rollen geführt, seitlich und oben mit Statikteilen und flachen Bauplatten gehalten.

Ohne dass er zu wackeln beginnt, kann der Auszug bis auf etwas mehr als 30 cm Länge ausgefahren werden.

Im Folgenden betrachten wir die Konstruktion des ganzen Modells Schritt für Schritt.

### Antrieb

Der Antriebsmotor ist mit Verbindungsstücken 30 oder 45 zwischen den zwei tragenden Alus befestigt (Abb. 3).

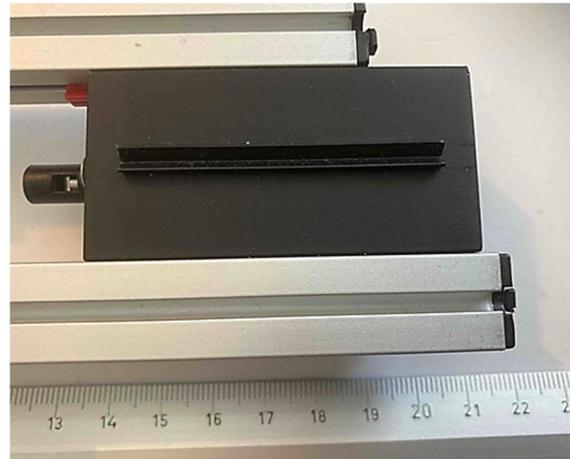


Abb. 3: Antriebsmotor

Ein weiterer, quer montierter Alu dient zur optimalen Stabilität und zur Befestigung der ausschwenkbaren Stützen:



Abb. 4: Rahmen

Der Antrieb erfolgt über eine kurze Welle über die Kette auf die beiden hinteren Starrachsen (Abb. 5 – Sicht von unten).

Der Drehkranz liegt nur 5 mm über den Alus und ist mit 4 BS5 sehr stabil befestigt. Die Drehung erfolgt über einen XS-Motor, welcher die Kraft gut übertragen kann, solange der Teleskoparm nicht zu stark ausgelenkt ist (Abb. 6).

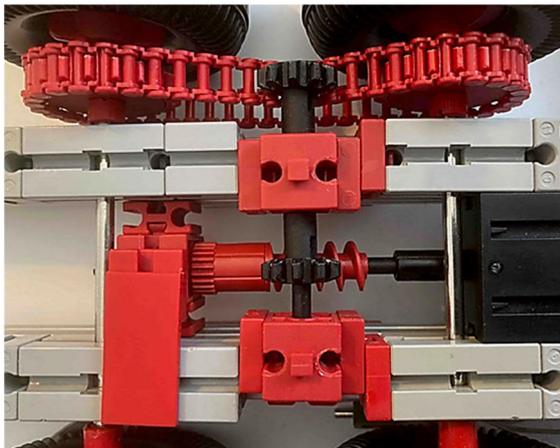


Abb. 5: Antrieb Hinterachsen



Abb. 6: Drehkranz mit Motor

In Abb. 7 sehen wir nun die Aufhängung des Teleskoparms. Der hintere Aufbau auf dem Drehkranz muss ebenfalls ziemlich robust sein, um den ganzen Oberwagen und die Ballastgewichte zu tragen.



Abb. 7: Aufhängung Teleskoparm

## Lenkung

Die vorderen zwei Achsen werden proportional ausgelenkt. Abb. 8 und 9 zeigen die Lenkung von oben und von unten.

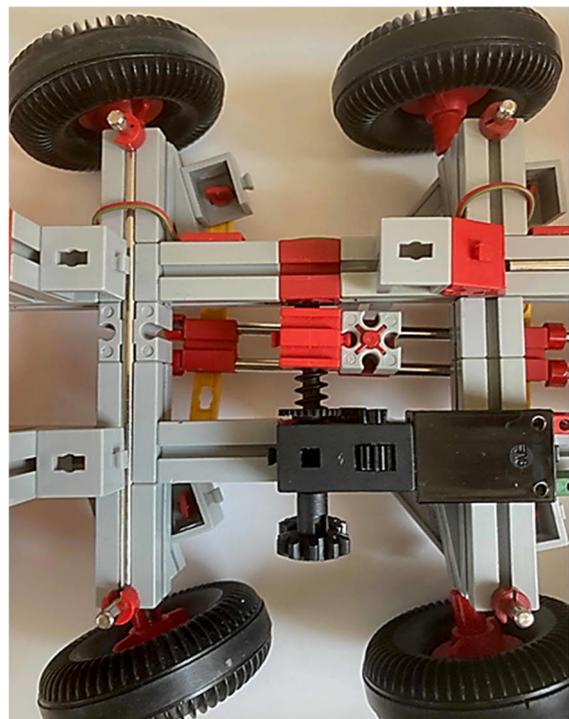


Abb. 8: Lenkung von oben

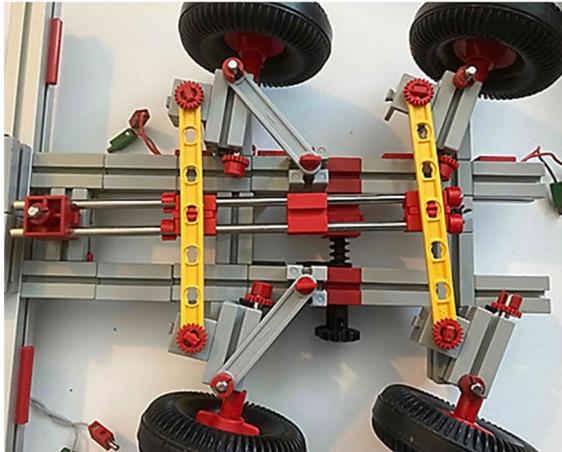


Abb. 9: Lenkung von unten

## Teleskoparm

Nun schauen wir uns den Aufbau des Teleskoparms genauer an. Abb. 10 zeigt den Auszug mit den Zahnstangen.

Im Teleskoparm gleitet der Auszug sehr gut. Er wird auf der oberen Seite durch mehrere V-Bauplatten 15 · 90 geführt. Seitlich sind Winkelsteine 30 angebracht, damit er sich beim Zurückfahren nicht verhakt.

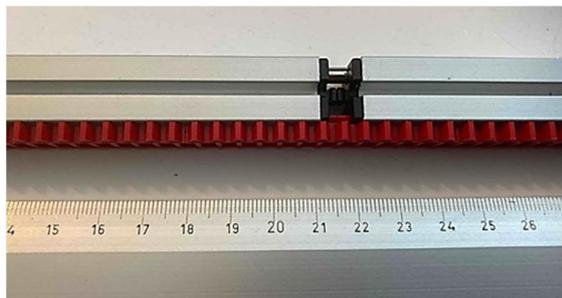


Abb. 10: Auszug Teleskoparm

Auf der Unterseite wird der Auszug über Rollen geführt (vgl. Abb. 11).

Abb. 12 zeigt uns, wie der Teleskoparm ausgefahren wird. Abb. 13 zeigt den ganzen Teleskoparm von unten. Der äußere Arm ist ca. 36 cm lang. Voll ausgefahren erreicht er eine Höhe von etwa 70 cm.



Abb. 11: Führung im Teleskoparm

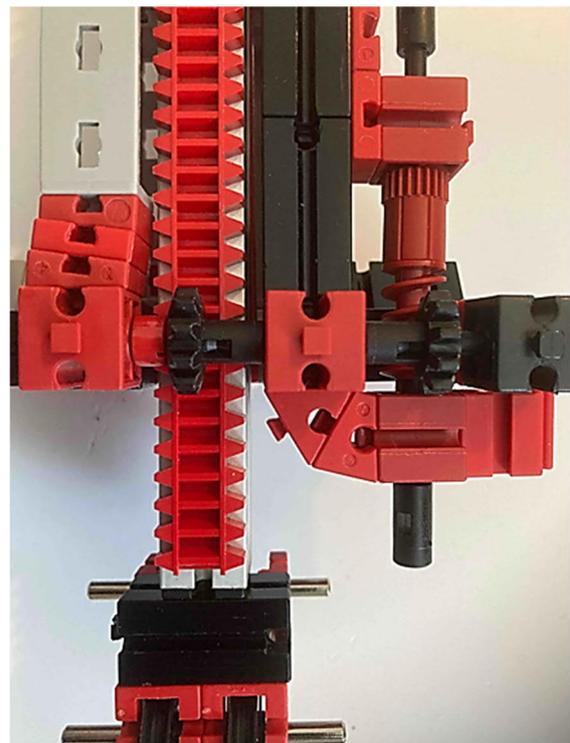


Abb. 12: Getriebe des Teleskoparms



Abb. 13: Teleskoparm von unten

Auch das Ausfahren des Teleskoparms erfolgt durch einen XS-Motor (ersichtlich in Abb. 16), dieser hat sich aber in der Praxis als eher zu schwach erwiesen.

### Aufrichten des Teleskoparms

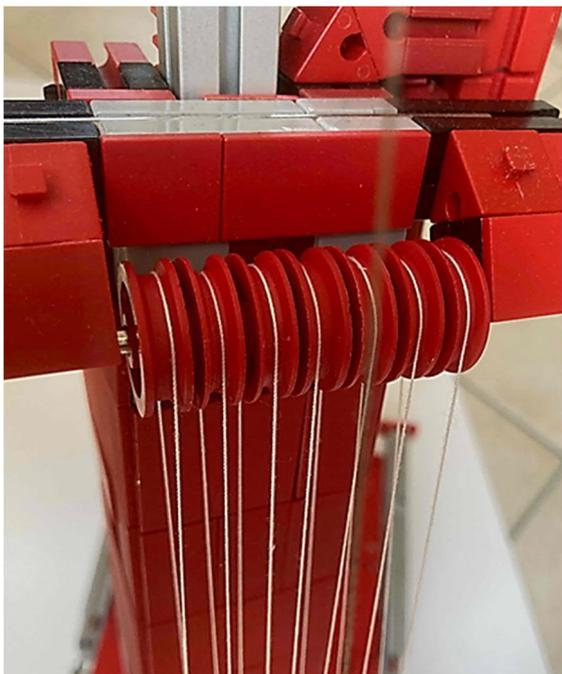


Abb. 14: Flaschenzug mit Befestigung am oberen Ende des Arms

Zum Aufrichten des Arms ist eine erhebliche Kraft nötig. Mit einem 7-fachen Flaschenzug und einem XM-Motor funktioniert dies erstaunlich gut, sogar bei (teilweise) ausgefahrenem Arm.

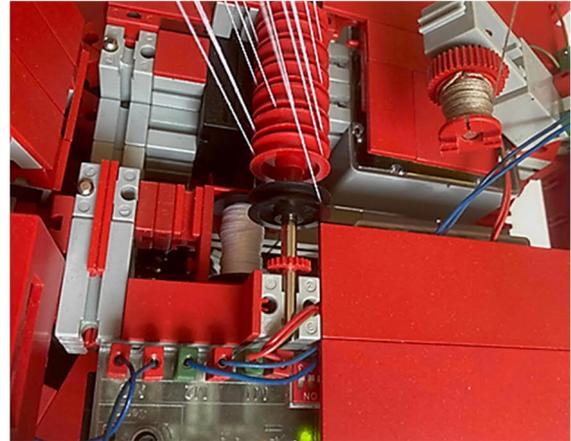


Abb. 15: Hebevorrichtung

Abb. 14 zeigt die Befestigung am oberen Ende, Abb. 15 den Motor mit dem Flaschenzug, sowie den (kleinen) Lastmotor, die Steuerelektronik und die Gegengewichte unter der Abdeckung.

### Oberwagen

Nun sehen wir den Oberwagen als Ganzes (Abb. 16). Der Oberwagen ist nur etwas mehr als 30 mm hoch (unten in der Mitte ist eine Abstützung für die Gegengewichte).

Ganz rechts auf dem Bild erkennt man die um 90° ausschwenkbaren Stützen.



Abb. 16: Oberwagen

Der Oberwagen hat natürlich eine eigene Kabine für den Kranführer (Abb. 17).

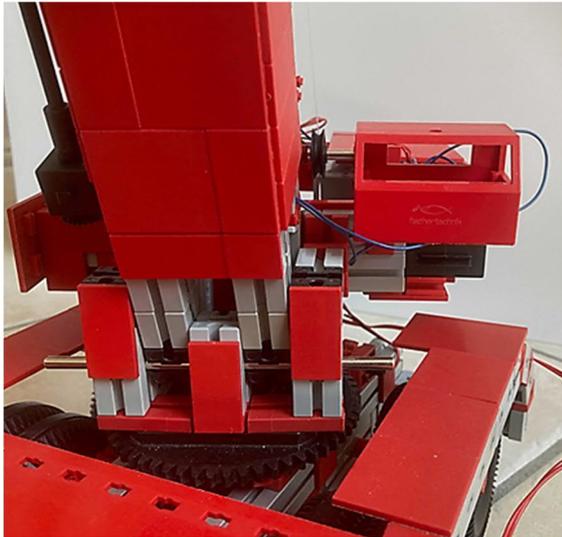


Abb. 17: Kranführer-Kabine

## Fazit

Das ganze Modell funktioniert recht gut und ist belastbar gebaut. Dank des stabilen Drehkranzes und den Gegengewichten kann der voll ausgefahrene Teleskoparm problemlos bis auf etwa 60° aus der Vertikalen ausgelenkt werden.

## Quellen

- [1] Migl, Alexander: *Liebherr-Fahrzeugkran zur Bergung verunglückter Rennwagen am Circuit de Monaco*. [Wikimedia Commons](#), 2019 (CC BY SA 4.0).

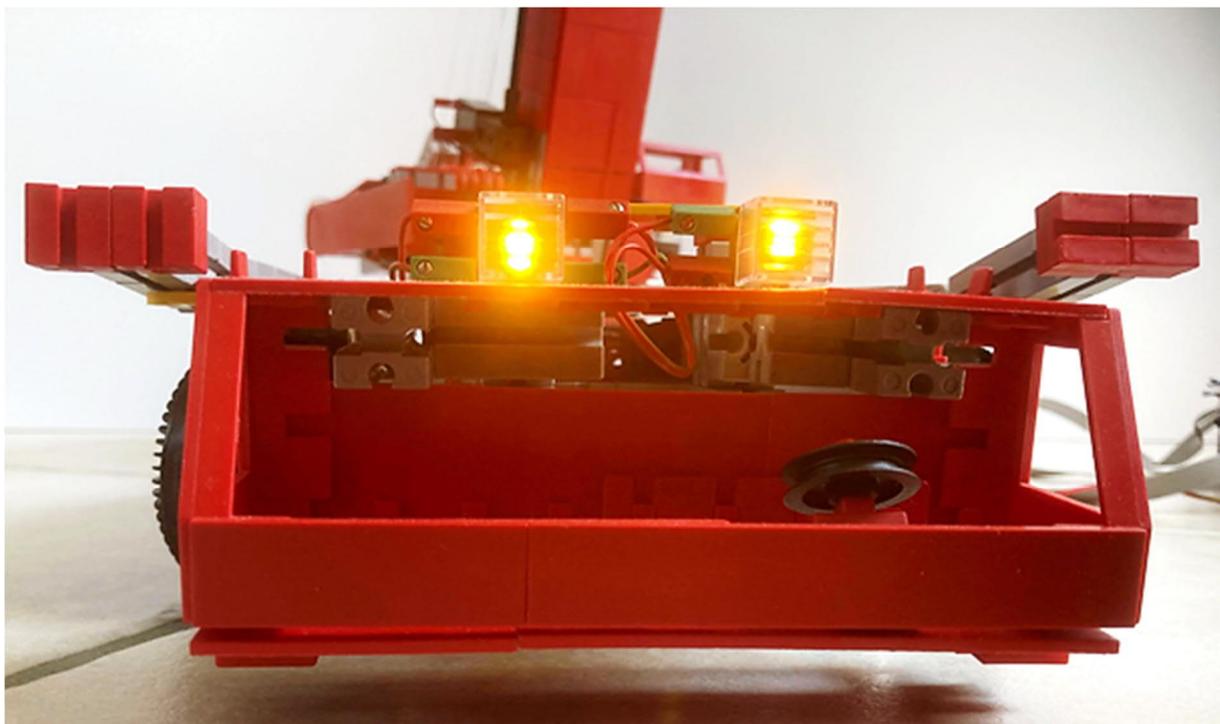


Abb. 18: Das Modell von vorne

Modell

## Schrägseilbrücke und Hängebahn (Teil 2)

Tilo Rust

*Nachdem ich im ersten Teil [4] die Architektur und Statik des Bauwerkes erläutert habe, befasst sich dieser Teil mit dem Antrieb und der Regelung für die Hängebahn an unserer Schrägseilbrücke.*

### Rahmenbedingungen

An der Fahrbahn der Schrägseilbrücke hängt eine Gondel auf Laufschiene. Sie ist mit jeweils drei Rädern von jeder Seite eingehängt, sodass die Halteseile der Brücke zwischen dem Fahrwerk hindurch gleiten. Da die Fahrbahn aufgrund der Konstruktion labil aufgebaut wurde, sich also vertikal verbiegt und dabei in der Länge variiert, scheidet ein Antrieb mit einem umlaufenden Seil aus. Die nötige Seilspannung, welche erforderlich wäre, um die Reibung auf dem Antriebsrad sicherzustellen, würde die Fahrbahn verformen und damit die Seillänge zusätzlich verändern. Die Kraft wäre (in Experimenten ermittelt) sogar so hoch, dass die Türme zueinander gezogen werden würden. Zudem reicht ein kleiner Stoß an einen der Türme z. B. auf einer Ausstellung, um deren Entfernung zu verringern und so die Seilspannung unter den Grenzwert zu bringen (siehe Video [2]). Zusätzlich wäre der Auf- und Abbau umständlicher, da mindestens eine Achse entfernt werden müsste.

### Zweiseil-Technik

Die Lösung bietet ein Antrieb aus zwei gegenläufigen Seilen. Hierbei zieht ein Seil („Unterseil“) die Gondel in Richtung Talstation (in welcher sich der Antrieb befindet) während das „Oberseil“ über der Fahrbahn (auf ihr liegend) zur Bergstation

führt, dort um eine Welle umgelenkt wird und somit die Gondel zur Bergstation zieht.

Beide Seile werden auf eigenen Rollen aufgewickelt, die mit Hilfe eines Getriebes gleich schnell laufen. Um die Seile gegen zu starken Durchhang zu spannen, befindet sich im Unterturm („Schacht“) der Talstation ein Spanngewicht, welches in das Oberseil eingehängt ist.

### Dilemma

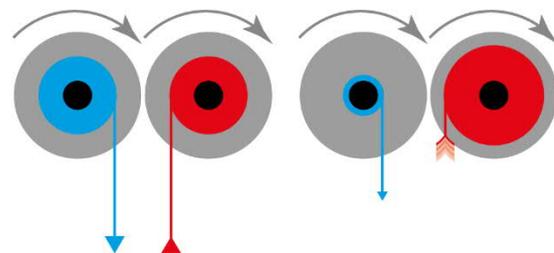


Abb. 1: Laufunterschiede

Laufen beide Rollen gleich schnell, spult also eine Rolle genau so viel Seil ab, wie die andere dieses aufrollt, ist alles in Ordnung (links). Doch nach einer Weile verringert sich der Spulendurchmesser einer Rolle (da Seil abespult wurde), während auf der anderen Rolle der Durchmesser wächst. Damit wird pro Umdrehung weniger Seil abgegeben, als aufgenommen wird; das Seil verkürzt sich (rechts).

Das Spanngewicht im Schacht wird dadurch nach oben gezogen – die Spannung selbst bleibt konstant, da das Gewicht sich nicht ändert.

Je nach Auslegung der Rollendimensionen und des Flaschenzuges am Gewicht könnte man damit die Seillängenänderung ausgleichen oder in der Grenze der Schachtlänge halten. Aber: Das wollen wir doch gar nicht – dann fehlte ja der Lerneffekt!

Also benutzen wir absichtlich schmale Rollen (damit sich die Durchmesser schnell ändern) und eine relativ kurze einfache Seilzuglänge im Schacht. Dadurch forcieren wir das nächste Problem (absichtlich): die Seillänge muss während (!) der Fahrt geregelt werden.

## Messgröße

Als Messgröße bietet sich die Position des Spannunggewichtes an. Wird das Seil „viel“ zu kurz, wird es „zu hoch“ gezogen – und entsprechend umgekehrt. Das Gewicht könnte einen Kontakt auslösen, der dann über das (Gleichlauf-)Getriebe die Spulen gegeneinander verdreht (s. u.).

Die Probleme dabei sind zwei Effekte.

- Zum einen würde die Regelung bei Auslösen des Kontaktes sofort Seil nachgeben, das Gewicht nach unten fahren, der Kontakt abfallen und das Getriebe angehalten (um kein Seil mehr nachzugeben). Doch dann beginnt der Prozess von neuem recht schnell, je nach Geschwindigkeit der Komponenten. Das würde zu einem Aufschaukeln, einer Resonanzschwingung führen. Die Gondel schwingt, die Seile schwingen, das Gewicht schwingt, im Extremfall die Türme – ganz schlecht.
- Zum anderen ruckelt die Gondel beim Fahren durch Fahrbahnunebenheiten selbst etwas. Das System kommt damit schon in unregelmäßige Schwingungen. Dadurch würde ein sicheres Auslösen des Kontaktes schwer realisierbar. Beobachtet man das Spannunggewicht bei der Fahrt, sieht man, wie es mehrere Zentimeter „hüpft“ (siehe Video [2]).

## Hysterese

Schnell käme man auf die Idee, eine „Zeitverzögerung“ einzubauen, also sobald der Kontakt geschlossen wird, noch eine Zeit zu warten, bis Seil gegeben wird. Und beim Öffnen wieder etwas Zeit zur Verzögerung. Das ist nett gedacht, aber technisch genau falsch. Denn die Seilmenge, die gegeben werden muss, und die Geschwindigkeit, in der der Fahrbetrieb die Seillänge verändert, sind nicht konstant. Mal wird nur ganz wenig Seil gebraucht, mal geht es rasend schnell (je nach unterschiedlichen Spulendurchmessern). Wenn also die Zeit zu kurz wäre, würde sich das System nur noch mehr aufschwingen. Ist die Zeit zu lang, könnte es sein, dass das Gewicht am Ende des Schachtes ankommt und damit die Anlage Schaden nimmt. Zu diesem Zweck sind Endabschaltungen enthalten, die einen kompletten Stillstand der Anlage erzwingen. Sowohl im Falle eines Seilrisses (Gewicht fällt nach unten), als auch im Fall, dass das Seil klemmt (Gewicht wird nach oben gezogen). Dann ist der Notbetrieb nötig – siehe unten.

Die Lösung ist eine *mechanische Hysterese*. Dazu löst das Gewicht den Kontakt nicht direkt aus, sondern nimmt eine Steuerkette mit, welche den Kontakt betätigt. Der Trick: die Kette wird mit großem Spiel mitgenommen.

Am Gewicht befindet sich also ein Mitnehmer, der die Kette mit nach oben zieht. Wird nun Seil gegeben, dann fährt das Gewicht nach unten, der Mitnehmer greift aber einige Zentimeter ins Leere, bevor er die Kette dann wieder mitnimmt und so den Kontakt abschaltet. Vorteil dieser Methode: Die Regelung setzt sofort ein, läuft aber mindestens so lange, wie die Wegstrecke der mechanischen Hysterese ist. Unabhängig von der dazu benötigten Zeit ist so mechanisch sichergestellt, dass Seil nachgegeben wird.

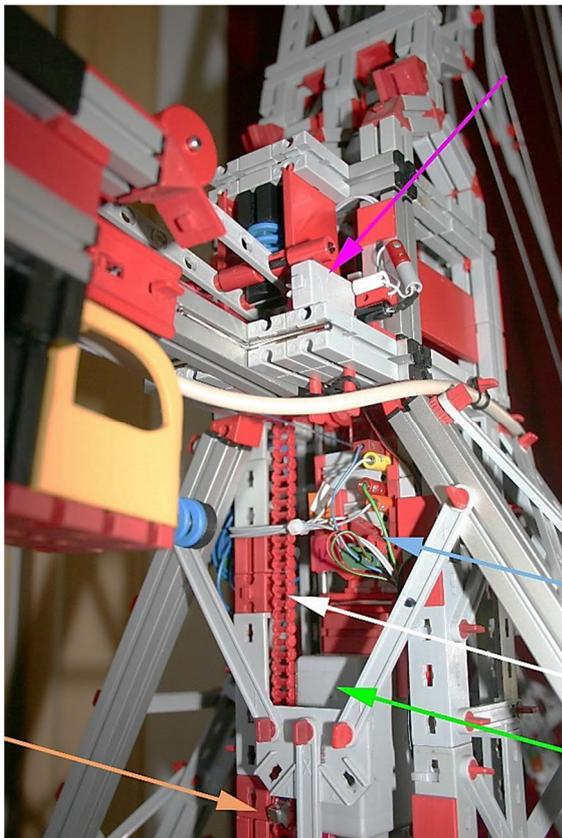


Abb. 2: Elektrik im Schacht

In Abb. 2 ist alles auf einmal zu erkennen:

- Die Gondel (links) wird mit dem blauen Seil (ganz dünn unterhalb des Seriellen Kabels zu erkennen, welches aus der Gondel zu kommen scheint) zur Talstation gezogen.
- Das Gewicht im Schacht (grüner Pfeil) sorgt für die Seilspannung. Bei Längenschwankung nimmt es die Steuerkette (weißer Pfeil) mit. Ein Neodymmagnet darauf (orangener Pfeil) löst die Reedkontakte aus (im BS15 mit Loch, Höhe des weißen Pfeils). Damit wird die Seillänge ermittelt und das Relais gesteuert.
- Endschalter (hellblauer Pfeil) lösen den Notstopp bei einem Fehler aus.
- Der Endschalter (lila Pfeil) welcher von der Gondel ausgelöst wird, dient als Endlage-Erkennung in der Station und hat mit der Seilspannung nichts zu tun – nur mit dem Antrieb.

## Kontaktprobleme

Unser Test hat gezeigt, dass das Auslösen von Minitastern Probleme bereitet. Das Gewicht selbst hängt komplett reibungsfrei im Schacht, geführt an Stahlstangen. Die Kette benötigt etwas Kraft für die Bewegung, was sich mit einem Faktor zwei durch den Flaschenzug auf die Kraft der Spulen überträgt. Käme nun ein Minitaster dazu, so könnte das Gewicht verkanten oder die Kette blockieren. Dieses Risiko gehen wir nicht ein. Deswegen verwenden wir Reedkontakte, die berührungsfrei auslösen. Dazu ist auf der Gegenseite der Steuerkette ein kleiner Neodymmagnet montiert. Er löst je einen Reedkontakt für zu kurzes oder zu langes Seil aus. Zur Sicherheit sind diese sogar doppelt ausgeführt und parallelgeschaltet.



Abb. 3: Reedkontakte

Die üblichen fischertechnik-Reedkontakte (links) sind schwer zu befestigen und unsauber zu positionieren. Es gibt zwar einige Tricks und Reedkontakthalter, aber wir haben uns entschieden, professionelle Industrieware zu verwenden. Diese gibt es in Durchmessern von 4 mm und 5 mm, als Öffner, Schließer oder Wechsler, sind wasserdicht und präzise. Mit BS7,5, BS30 mit Bohrung und BS15 mit Loch (4 mm)

oder Gelenkwürfel-Klaue (5 mm) u. a. sind diese Teile bestens zu positionieren.

## Doppellösung

Aus der Verwendung eines Reedkontakts folgt ein weiteres Problem: Diese Kontakte können nur eine Leitung schalten. Wir benötigen aber beide Leitungen (Plus und Minus) und diese dazu auf drei Stellungen:

- Seil geben
- Stillstand (spannungsfrei; Seillänge ist richtig)
- Seil kürzen

Wir benötigen einen Polwendeschalter mit Nullstellung – und das mit einem einzigen Reedkontakt?

Des Rätsels Lösung ist ein Relais. Also haben wir einen „Selbstbau-Silberling“ entworfen. In einer Kassette 60 · 60 mit Deckel befinden sich nun zwei Relais mit jeweils fünf Umschaltkontakten sowie einem Sperrkontakt (max. Schaltstrom 2 A). Löst also nun ein Reedkontakt aus, so zieht eines der Relais an und gibt mit zwei Kontakten Spannung in der richtigen Polung auf den Motor, welcher die Seillänge verändert. Löst der andere Reed aus, wird das andere Relais tätig und die Spannung wird auf den Leitungen verpolt aufgegeben.

Damit nicht beide Relais gleichzeitig anziehen (z. B. weil ein „Spielkind“ mit einem Magnet die beiden Reedkontakte auslöst), ist der sechste Kontakt der Relais als Sperrkontakt ausgeführt. Ist also ein Relais angezogen, wird die Versorgung des anderen Relais durch Öffnen des sechsten Kontaktes verhindert. (Anm.: Um den Silberling später in anderen Projekten verwenden zu können, gibt es auch eine „Override“-Ansteuerung, die es prinzipiell ermöglichen würde, beide Relais komplett unabhängig zu verwenden oder den sechsten Kontakt als Selbsthaltung zu benutzen.)

Doch damit nicht genug: Ist ein Relais angezogen, liegt die Spannung auch an den

Kontakten des zweiten Relais an – allerdings verpolt. Dadurch, dass dessen Öffner nun aktiv sind, käme es zu einem Kurzschluss. Wieder eine Lösung: Dessen Zuleitungen werden über die zwei weiteren (Öffner-)Kontakte des ersten Relais geführt und sind damit in diesem Moment spannungsfrei (die sogenannte „elektrische Verschränkung“).

Somit bleibt ein Kontakt an jedem Relais übrig – verwenden wir ihn einfach für eine Anzeige auf der Fernbedienung.

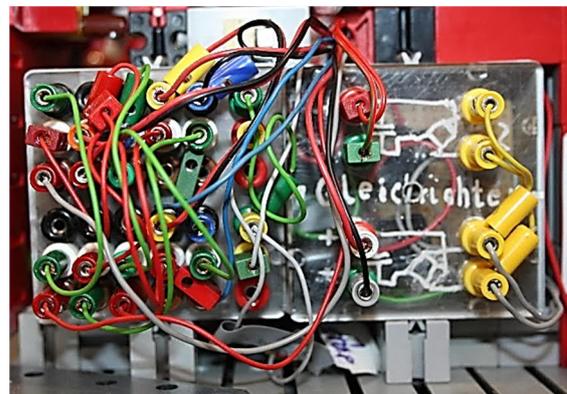


Abb. 4: Die „Selbstbau-Silberlinge“: Links das Relais 5 · UM (mit 36 Buchsen auf der Oberseite, daher das Kabelgewirr) und rechts der Brückengleichrichter

## Getriebe

Wie schon erwähnt, muss diese Regelung während der Fahrt durchgeführt werden. Damit müssen zwei mechanische Bewegungen „addiert“ oder überlagert werden. Zum einen die Fahrbewegung (ein Seil abrollen, das andere aufrollen) und zum anderen die Seillängenveränderung, bei der eine Spule schneller als die andere läuft.

Mechanisch ist das ein bekanntes Problem, das in jeder Raupe oder jedem Panzer gelöst ist. Das Gleichlaufgetriebe erlaubt es, beide Ketten solcher Fahrzeuge in die gleiche Richtung zu bewegen und zum Lenken deren Laufgeschwindigkeiten gegeneinander zu versetzen. Im Bilderpool gibt es dazu dutzende Lösungen. Das gleiche Getriebe setzen wir hier ein.

Damit ist ein Motor für den Antrieb „Fahrt“ zuständig, während der andere Motor über Relais, Reedkontakte, Steuerkette und Spanngewicht die Seillängen regelt. Im Video [2] ist schön zusehen, wie eine Spule fast stehen bleibt, während sich die andere bewegt.

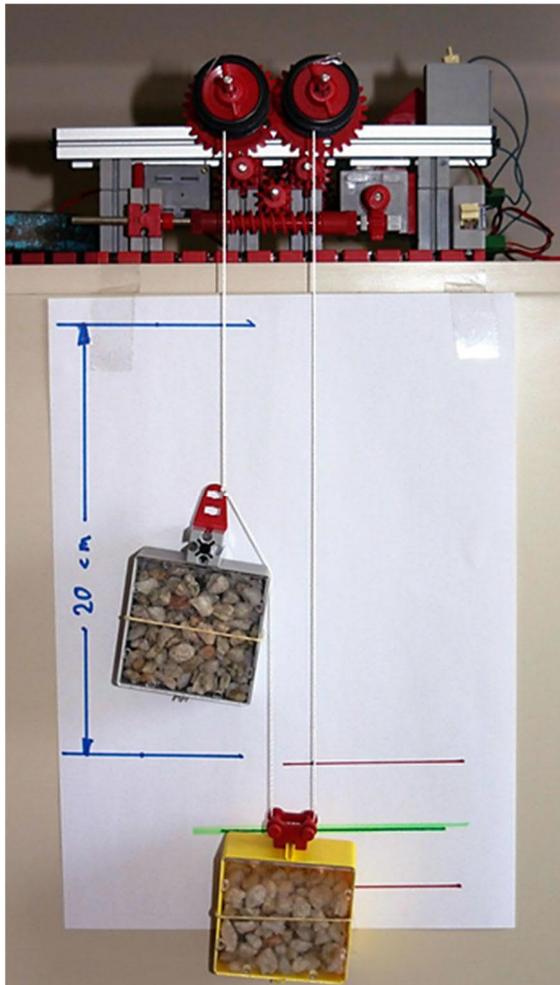


Abb. 5: Das Gleichlaufgetriebe

Die Seillängenänderung ist (absichtlich) extrem: während sich die Gondel (graue Kasette) nur 20 cm bewegt, ändert sich die Seillänge am Spanngewicht (gelbe Kasette) bedingt durch die absichtlich eng bemessenen Spulendurchmesser erheblich. Man bedenke: unser Modell besitzt eine Fahrbahnlänge von 360 cm.

## Automatik vs. Handbetrieb

Über die Kabelfernsteuerung (s. u.) ist der Betrieb in zwei Modi möglich:

Der *Automatikbetrieb* lässt die Gondel von einer Station in die andere durchfahren. Währenddessen regelt der beschriebene Regelkreis die Seilspannung selbsttätig. Am Turm angekommen unterfährt die Gondel eine „Gabel“, welche einen Minitaster öffnet und so den Fahrstrom unterbricht. Wird an der Fernbedienung die Fahrtrichtung geändert, erhält der Antrieb über eine Diode am Minitaster wieder Strom und die Fahrt in die Gegenrichtung beginnt.

Bei einer Störung, z. B. Verklemmen der Steuerkette, einer Entgleisung oder anderen Problemen, welche von der Anlage durch Sensoren erkannt werden, wird die Fahrt unterbrochen und die Regelung der Seilspannung angehalten. Gleichzeitig leuchtet auf der Fernbedienung eine Warnleuchte auf. Nun kann eine Klappe geöffnet werden. Dadurch werden zwei Taster ausgelöst, welche den Automatikbetrieb unterbrechen und die Steuerung auf die Notsteuerung umschaltet. Es steht ein Taster zur Verfügung, der die Gondel manuell in die vorgewählte Richtung bewegen kann, außerdem ein Polwendetaster, mit dem es möglich ist, manuell die Seillänge unabhängig von der Regelung zu erhöhen oder das Seil zu verkürzen. Dazu ist keine Fahrt nötig. Erst wenn sich die Anlage wieder in einem geordneten Zustand befindet, wird die Klappe geschlossen und der Automatikbetrieb setzt ein.

Damit ist die Anlage im Normalfall, also bei Automatikbetrieb, lediglich mit einem einzigen (Polwende-)Schalter zu bedienen.

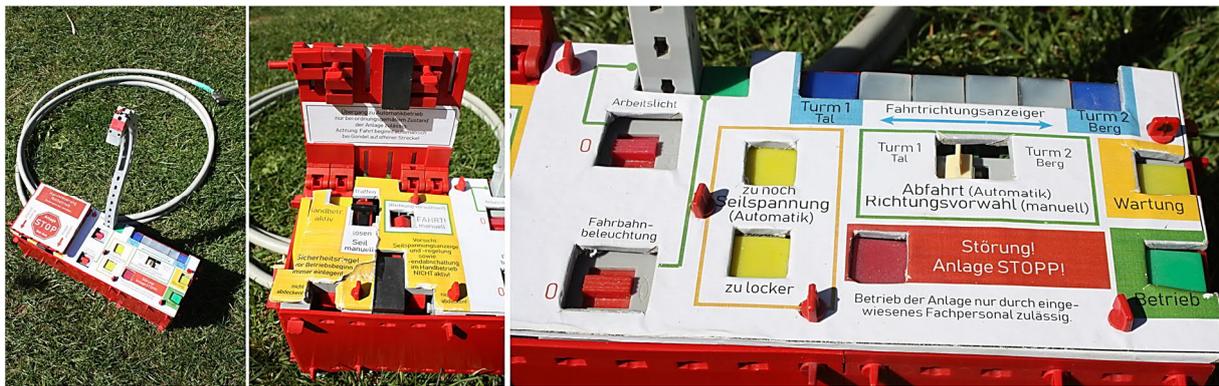


Abb. 6: Fernbedienung

## Auf- und Abbau

Dieser Handbetrieb (Notbetrieb) ermöglicht nun auch den einfachen Auf- und Abbau der Anlage, vorgeführt im Video [2]. Sobald Brücke und Fahrbahn mit Gondel stabil aufgebaut sind, werden die Seile von den Spulen abgerollt, auf denen sie zum Transport aufgewickelt waren. Dazu wird im Handbetrieb einfach „Seil verlängern“ gedrückt. Sind dann beide Seile in die Gondel eingehängt und etwas über diese manuelle Steuerung gespannt, so kann der Automatikbetrieb beginnen. Nach zwei bis drei Fahrten haben sich die Spulen ausgeglichen und ein stundenlanges reibungsloses Betrieb ist möglich. Der Abbau erfolgt entsprechend umgekehrt.

## Kabelchaos

Ein Blick in den Maschinenraum bringt bei den Betrachtern meistens ein spontanes Lachen und Verzweifeln mit sich: Hunderte von Kabeln liegen scheinbar chaotisch auf einem Haufen. Doch dem ist nicht so. Wie ich schon in [4] und [5] schilderte, sind die Kabel geordnet und beschriftet. Doch hier kommt der Kick: Vom Trafo (vergraben rechts neben dem weißen Bereich) führt der Strom über die Fernbedienung, welche mit einem Centronics-Stecker (unterer Rand, zwischen grünem und Magenta-Bereich) und 3 m Telefonkabel angeschlossen ist, in die Talstation, dann über ein seriell Kabel zur Bergstation an die oben erwähnte Gabel, zurück zur Talstation, zur

Fernbedienung in den Polwendeschalter, zur Talstation, dort in dessen Gabel, zum Fahrmotor (linker Powermotor zwischen den Pfeilen), wieder in die Fernbedienung zur Handsteuerung, zur Talstation für die Notabschaltung, zur Fernbedienung zum Override und zurück zum Trafo. Einiges an Strecke, oder? Zusammen mit den Kabeln für die Regelung und die Beleuchtungen, für die Signallampen und die Wartungsmodi sind in diesem Modell über 240 m Litze und ca. 250 Stecker bzw. Buchsen verlegt (und fast unsichtbar).

Abb. 7 zeigt den Maschinenraum: kompakte Technik aus allen Bereichen. Der „Block“ (weißes Rechteck) leitet die Kräfte der Schrägseilbrücke in die Bodenplatte ein (vgl. [4]). Rechts daneben befindet sich der einzige (klassische) Trafo, der die komplette Anlage versorgt. Rechts daneben sind zwei Polwendeschalter zu erkennen, die zusammen mit den blauen Lampen die Anlage in „Wartung“ versetzen und Steuerbefehle nicht an die Motoren, sondern die Lampen übergeben. Rechts daneben sieht man die blaue Seilspule (linker Pfeil), den Motor für die Seillängenregelung (mit orangefarbenem Aufkleber) und darunter versteckt die zweite Spule (rechter Pfeil). Der zweite Powermotor (violetter Aufkleber) ist zuständig für die Fahrt der Gondel. Die beiden Motoren wirken über das Gleichlaufgetriebe (violette Rechteck) auf die Spulen. Links neben dem Getriebe sieht man das „Kabelchaos“, das durch das

Auffächern des Centronics- und des seriellen Kabels entsteht.

Die Selbstbau-Silberlinge (grüner Bereich) verbergen sich unter dem liegenden Polwendeschalter, der das Arbeitslicht von „Artur“ (siehe [4]) schaltet. Über diesem sitzt ein schwarzes Kästchen – das Blinkrelais für die Topplichter. Rechts unten neben dem grünen Bereich sieht man eine Lampenfassung; das ist die Steckdose für die Handlampe. Der freie Bereich daneben gibt dem Centronics-Kabel Platz.

Der Geber für den Fahrtrichtungsanzeiger (gelber Bereich) greift mit einem Z40 in ein Z10 ein, das auf der Achse der blauen Spule sitzt. Zu erkennen sind dort die schwarzen Reedkontakte für die LED-Anzeige. Der Deckel, gleichzeitig das Dach der Anlage, besteht aus den rechts aufgestellten roten Bauplatten. Sobald er geschlossen wird, deaktiviert der kaskadierbare Taster (über dem weißen Bereich, mit schwarzem Betätiger) die gelben Warnlampen (sichtbar

z. B. unter dem weißen Bereich). Der Taster ganz rechts oben (betätigt durch das blaue Federbein) schaltet bei offener Klappe die Arbeitsbeleuchtung im ganzen Gebäude.

Am oberen Ende des magentafarbenen Bereiches sieht man ein geflochtenes rot/weißes Kabel, das eine Laterne – unscharf, direkt unter den Wartungsschaltern – aufrichtet. Zur Orientierung: die Seile laufen etwa mittig aus dem rechten Bildrand unter der Bauplatte heraus zum Turm der Talstation und von dort auf die Strecke.

### Stromversorgung

Warum dieses Chaos nötig ist? Nun, meine Anforderung war, dass die Anlage nur einen einzigen Netzstecker haben darf, also einen Trafo, der alles versorgt. Ebenso dass die Fernbedienung über Kabel angeschlossen und wie die Fahrbahn komplett entfernt werden kann, damit die Steuerleitung, welche unter der Fahrbahn angebracht ist, ausgesteckt werden kann. Die Bergstation

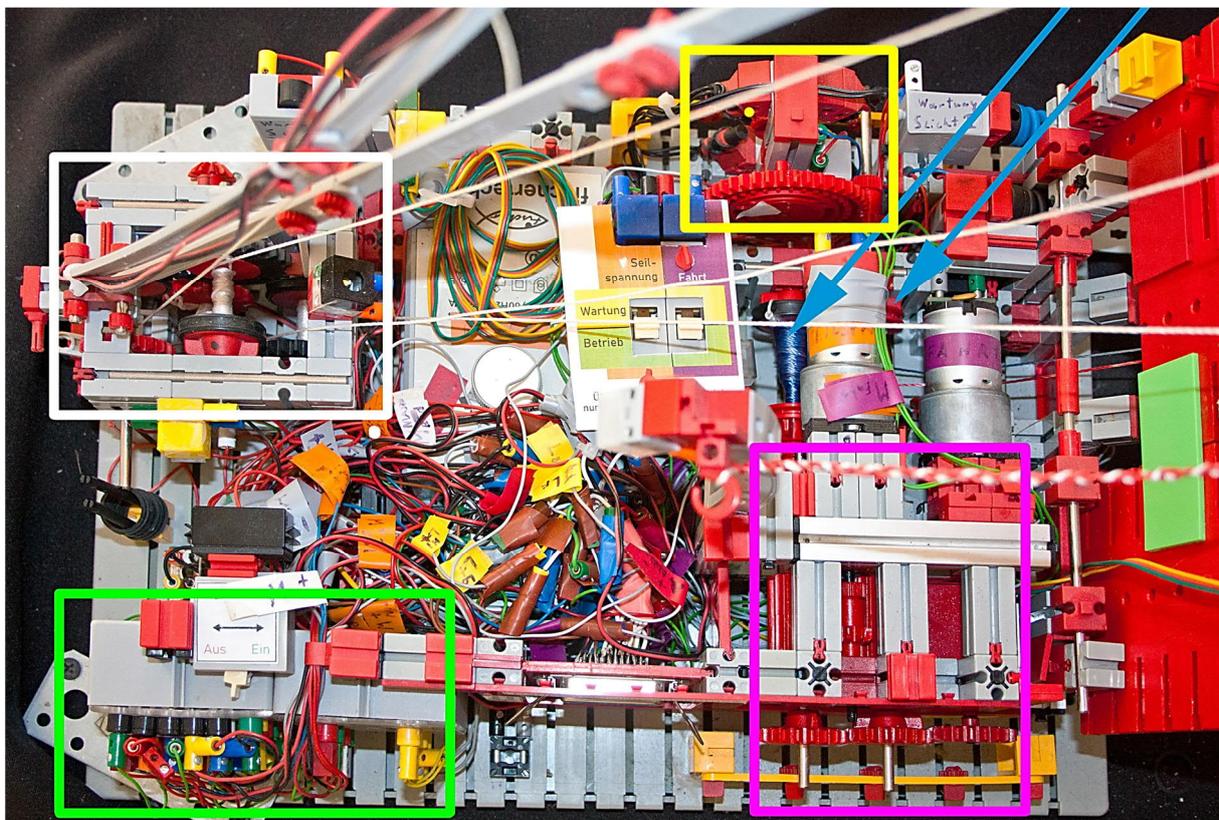


Abb. 7: Maschinenraum

muss „passiv“ sein, also Spannungsversorgung und Schaltsignale über das serielle Kabel bekommen und damit auch die Beleuchtung versorgen. Außerdem wird die Beleuchtung der Fahrbahn (die Lichterkette oben zwischen den Mastspitzen) ausschließlich von der Talstation versorgt. Die Wartungsbeleuchtung soll leuchten, wenn die Abdeckung des Maschinenraumes geöffnet wird, zusätzlich Warnlampen auch an der Fernbedienung. Der Nachtbetrieb (Arbeitslicht an der Fernbedienung) mit Beleuchtung aller technischen Geräteräume soll so möglich sein, dass wir keine Taschenlampe brauchen, wenn im Garten ein Problem mit der Anlage auftritt.

So kommen wir zu einer weiteren technischen Raffinesse: Der Trafo versorgt zwar mit seinen Gleichspannungsausgängen die Motoren, aber da die Beleuchtung zum Stromsparen aus (Selbstbau-)LEDs besteht, können wir keine Wechselspannung verwenden. Was liegt also näher als zwei Leistungs-Brückengleichrichter (5 A) mit gigantischen Kondensatoren (nur begrenzt durch die Einbaugröße) in eine weitere Kassette zu montieren. Nun ist die Anlage sogar fähig, die Beleuchtung fast 30 Sekunden am Laufen zu halten, nachdem der Netzstecker gezogen wurde. (Getreu meines Mottos „Gigantismus kennt keine Grenzen.“)

## Beleuchtungen

Die Anlage sollte einen Hauch von Realität einer professionellen Anlage widerspiegeln. Deswegen erfolgt beim Einstecken des Netzsteckers eine sofortige Aktivierung der Topplichter an den Mastspitzen. Diese zwei Lampen (!) werden über das klassische Blinkrelais von fischertechnik (es sitzt direkt über dem Relaisbaustein) synchron zum Blinken gebracht. Die Versorgung läuft über das serielle Kabel.

Sobald die Fernbedienung über den Centronics-Stecker eingesteckt wird, aktivieren sich die Kollisionslichter (rote Dauerlichter) sowohl an den Streben der Türme (versorgt ebenfalls über das serielle Kabel) als auch am Seil der Fahrbahnbeleuchtung (ausschließlich über die Talstation und das geflochtene Kabel versorgt). An der Fernbedienung leuchtet eine Betriebsanzeige.

Im Nachtbetrieb (siehe [5], S. 67) kann der Bediener die Fernbedienung mit einer Arbeitsleuchte erhellen (gebogene Laterne) und separat die Bauwerks- und Fahrbahnbeleuchtung aktivieren. Hierzu gibt es auch eine Kontrolllampe. Die elf Ultrabright LEDs werden wie die Kollisionslichter versorgt.

Sobald man die Abdeckung des Maschinenraumes anhebt, blinken dutzende gelbe Blink-LEDs, vor allem an Stellen mit Quetschgefahr, aber auch „Wartung“ an der Fernbedienung. Ist die Klappe ganz geöffnet, erhellen mehrere gut platzierte LEDs (weiß) alle technischen Elemente der Anlage (Getriebe, Fahrtrichtungsanzeiger, Kabelsalat, beide oberen Maschinenräume der Türme, sowie den Schacht von innen).

An der Fernbedienung gibt es außerdem den Fahrtrichtungsanzeiger (siehe [4] und im Video [2]), die Warnlampe für Störung und für die Spannungsregelung des Seils. Die Hinweislichter für Handbetrieb und Stationsankunft werden ebenfalls mit einer LED angezeigt.

Möglich ist diese Menge an Beleuchtung gleichzeitig an einem einzigen alten Trafo aber nur durch Verwendung von LEDs und meinem großen Gleichrichter – und den besagten Metern Kabelchaos.

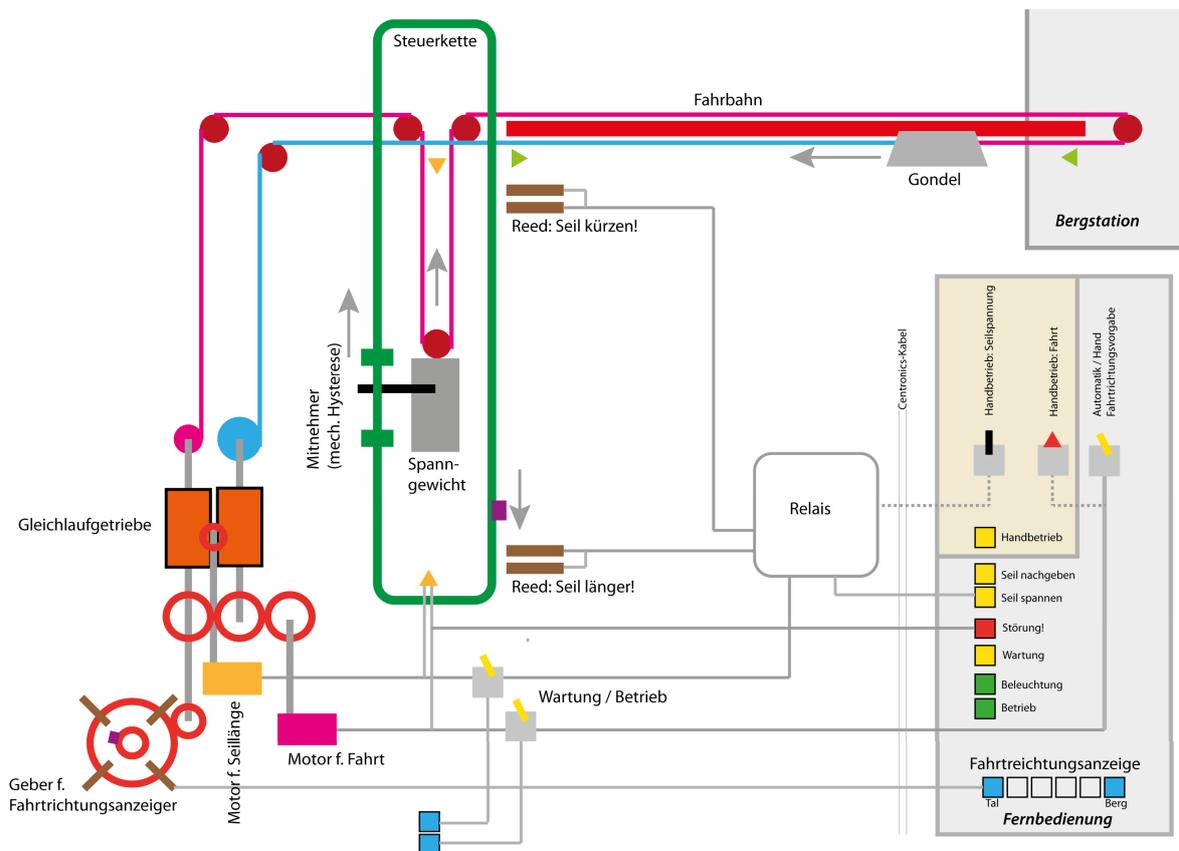


Abb. 8: Schema der Anlage

## Technik der 80er

Alles in Allem ist diese Anlage also Technik, wie sie in den 1970er und 1980er Jahren bereits möglich war (ClassicMan eben), basierend auf reiner Elektromechanik.

Natürlich haben wir auch darüber nachgedacht, die Anlage einmal mit meinem (relativ neuen) ftDuino und TX-Pi zu modernisieren. Doch Tests zeigen, dass die Reaktionszeiten der Regelung zu langsam sind, um einen sicheren Betrieb zu gewährleisten.

Zudem widerstrebt es mir persönlich, für das Einschalten eines Lichts einen Controller zu benutzen. Das geht auch klassisch: Schalter ein – Strom ein – Licht ein. Keep it simple.

Im Schema sind alle Komponenten der Anlage noch einmal im Zusammenspiel erkennbar.

## Kräfte

Abschließend noch eine genauere Betrachtung zu den Kräften, die in der Anlage auftreten. Diese sind additiv und enden letztlich im Getriebe/Motorblock.

Angefangen bei der Reibungskraft, die zum Ziehen der Gondel überwunden werden muss, über die additiven Kräfte der Seilreibung und Rollreibung der Umlenkrollen gelangen wir zum Spanngewicht. Dieses muss so ausgelegt sein, dass es in der Lage ist, das Seil zu spannen, wenn die Anlage stillsteht. Da das Gewicht an einer Umlenkrolle hängt, muss es doppelt so schwer sein, wie zur Überwindung der erwähnten Reibungen nötig. Zusätzlich sollte Reserve eingeplant werden. Auf der anderen Seite aber bedeutet dies, dass das Getriebe dafür ausgelegt sein muss, diese Kräfte aufzunehmen und die Rollen so übersetzt werden

müssen, dass der Motor in der Lage ist, das Gewicht ohne Probleme anzuheben.

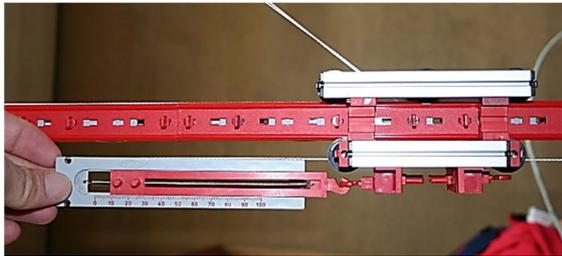


Abb. 9: Kraftmessung mittels Federwaage

Aus den ersten Versuchen ermittelten wir, dass es mit geschickter Getriebewahl möglich sein sollte, die Anlage mit den alten großen Motoren zu betreiben. Ich besaß nämlich keine modernen Powermotoren – schließlich bin ich der ClassicMan. Von der Realität aber wurden wir eines Besseren belehrt. Schnell war das Getriebe erweitert, so dass ein zweiter Motor auf der gleichen Welle für zusätzlich Kraft sorgen sollte. Doch die Schneckengetriebe fressen zu viel Kraft und die lange Schneckenstange

spreizte sich in alle Richtungen, sobald Last auftrat.

Nach der Süd-Convention 2018 sprang ich also über meinen Schatten. Dank fischerfriendsman mit seiner guten Beratung sind wir nun (zugegeben glückliche) Besitzer dreier Powermotoren. Das Getriebe wurde komplett neu gestaltet, verstärkt und verkleinert und tat von Anfang an das, was wir erwarteten – bis heute ohne Mucken. Interessant hierbei: Beide Getriebe wurden auf dem Kopf stehend aufgebaut. Sie sind als Module konzipiert und werden auf die Grundplatte nur mit wenigen Steinen eingeschoben. Alle Kräfte werden intern aufgenommen. Dies hat einen unschlagbaren Vorteil, den wir auch anderen Modellbauern empfehlen: Getriebe mit Motoren können als „Modul“ abgenommen, gewartet und repariert werden. Wir brauchen keine 20 Sekunden, um an jedes Rädchen zu kommen.

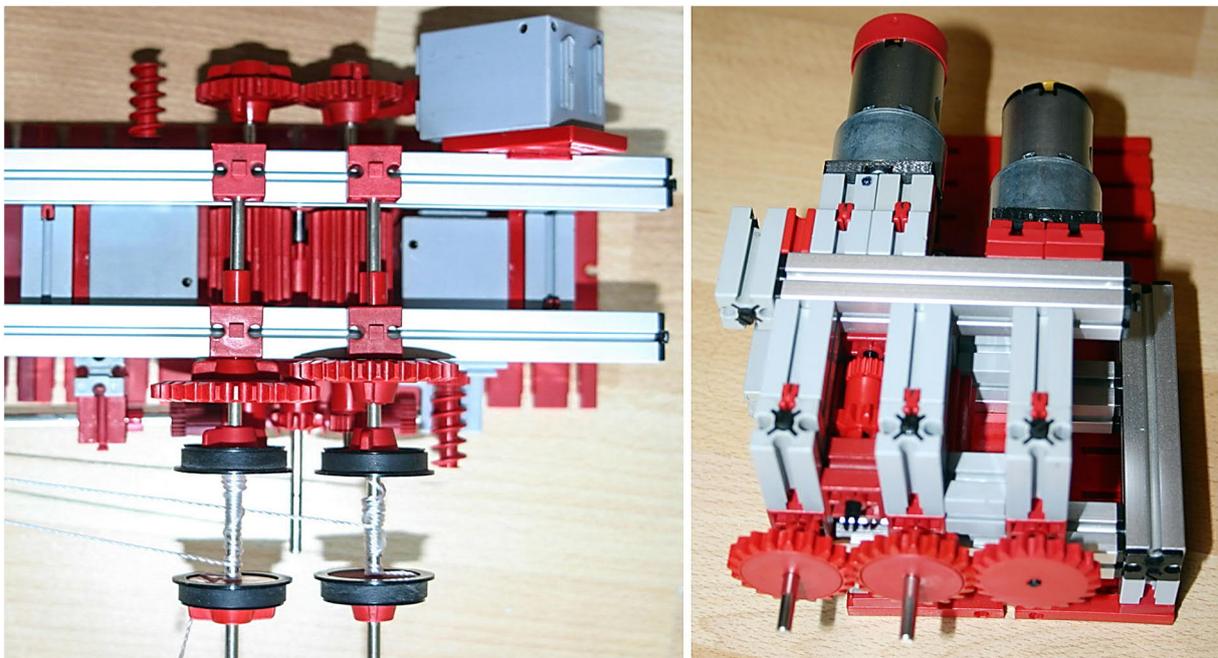


Abb. 10: Das Getriebe in der ersten Version mit den klassischen Motoren (der dritte Motor mit seiner Schnecke links oben ist noch ein Relikt aus der Entwicklung) und das wesentlich kompaktere und stabilere neue Getriebe mit den Powermotoren (hier liegen die Spulen unter den Motoren). Ein Z10 greift in beiden Versionen auf die Gehäusezähne der Differenziale und verdreht diese gegeneinander, um die Seillänge zu verändern.

## Abschied

Doch während ihr diese Zeilen lest, ist die Schrägseilbrücke und der ft:express mit seiner elektromechanischen Regelung längst Geschichte. Wie alle Bauten aus fischertechnik haben Jan und ich dieses Modell bereits wieder zerlegt. Es hat uns vier Jahre begleitet und vielen Besuchern Technik nähergebracht. In unserer 34-minütigen Video-Dokumentation [2] haben wir alles noch einmal für die Ewigkeit konserviert.

Die Teile aus unserem Privatbesitz geben meiner Frau damit wieder das Arbeitszimmer zurück und werden für ein neues Projekt eingesetzt: die 30 m lange kuppelbare Einseil-Umlauf-Seilbahn mit Hauptsponsor Doppelmayr Garaventa, die wir bis 2023 im Erlebnismuseum Fördertechnik in Sinsheim [3] aufbauen und in der nächsten ft:pedia vorstellen werden.

## Quellen

- [1] Tilo Rust: [Schrägseilbrücke \(340 cm\)](#). Bilderpool der ft:Community, 2016.
- [2] Tilo Rust: [fischertechnik Schrägseilbrücke & ft:express](#). Auf YouTube, 2020.
- [3] [Erlebnismuseum Fördertechnik Sinsheim](#) (Sponsor der Seilbahn)
- [4] Tilo Rust: *Schrägseilbrücke und Hängebahn*. Teil 1 in ft:pedia 3/2020, [S. 31-40](#).
- [5] Tilo Rust: *Verkabelung (Teil 3)*. In ft:pedia 2/2020, [S. 61-67](#).



Abb. 11: Das letzte Foto der Schrägseilbrücke in unserem Garten über den Teich, mit dem Chef-Konstrukteur Jan (11) und ClassicMan (50) bereitet auch den Besuchern Mümmeline und Bommel Freude – fotografiert von einer geduldigen Mama und Ehefrau.

Tipps &amp; Tricks

## Das Mini-Speichenrad

Ralf Geerken

*Um an das gute Innere heranzukommen, knackt man zu Weihnachten ja eigentlich Nüsse – als fischertechniker knackt man natürlich etwas anderes, z. B. ein altes Differential ([31043](#)). Aber keine Bange, hierbei ist es etwas anderes als bei der Nussschale, das Äußere nimmt dabei nämlich keinen Schaden und kann hinterher genauso gut verwendet werden wie das Innere.*

Ein bisschen vorsichtig sollte man beim Vorgang des Aufknackens aber schon sein, bei meinen ersten Öffnungsversuchen flogen mir die beiden herausschauenden Achsen nämlich buchstäblich um die Ohren. Da hatte ich das Differential einfach so in den Schraubstock geklemmt. Danach habe ich es lieber in eine geeignete Aufnahmegerät (siehe Abb. 3) gesteckt.



Abb. 1: Im Schraubstock

Wenn man jetzt den Schraubstock mit Vorsicht zudreht, verformt sich erstmal das Außenzahnrad recht stark. Das Bild, welches dabei entsteht, ist nichts für zart besaitete fischertechniker, deswegen habe ich davon erst gar keins aufgenommen. Dann springen aber plötzlich die beiden Achsen mitsamt den Rädern heraus.

Es gibt eine weitere Möglichkeit, das Innere aus dem Käfig zu holen. Dabei muss man aber wirklich sehr vorsichtig sein, denn sonst kann zumindest eine Achse dabei kaputt gehen.

Eine Achse spannt man dabei in den Schraubstock ein. Dann zieht man, bspw. mit einer Rohrzange, an der anderen Seite und wackelt dabei minimal hin und her – höchstens 10 mm nach oben und unten. Noch ein bisschen mehr Zugkraft geben, wenn man merkt, dass sich das Mini-Speichenrad aus dem Gehäuse löst. Eine Seite springt dann heraus. Die zweite Seite geht dann aber relativ problemlos, weil die Achsen in der Mitte dann nicht mehr ineinander verhaken können (logisch, eine ist ja schon draußen). Genau da liegt auch das Problem bei dieser Methode des Herauslösen – die Achsen können dabei nämlich leicht Schaden nehmen – also seid bitte vorsichtig. Mit diesen Innereien kann man bestimmt genauso viel anfangen, wie mit den Mini-Speichenrädern.

Genau um diese kleinen Räder geht es in diesem Artikel. Ich nenne sie mal Mini-Speichenräder – wobei es eigentlich Felgen sind, weil auf ihnen das Außenzahnrad (der Differentialkäfig) festgeklemmt ist.

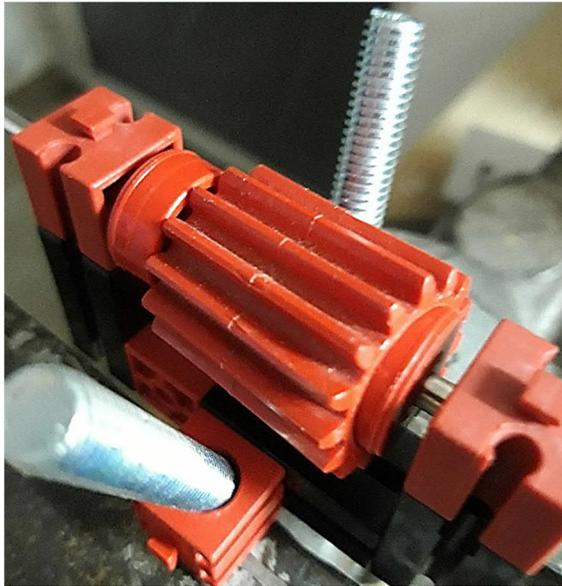


Abb. 2: Geknackt

Wie man in Abb. 2 sieht, hat das Außenzahnrad keinen Schaden genommen. Die Einzelteile des Differentials können nun gut aus dem Käfig herausgelöst bzw. genommen werden. Dabei ist zu beachten, dass alle Innenteile mit Fett verschmiert sind. Das sieht man gut an den Enden der Kegelzahnräder in Abb. 3.

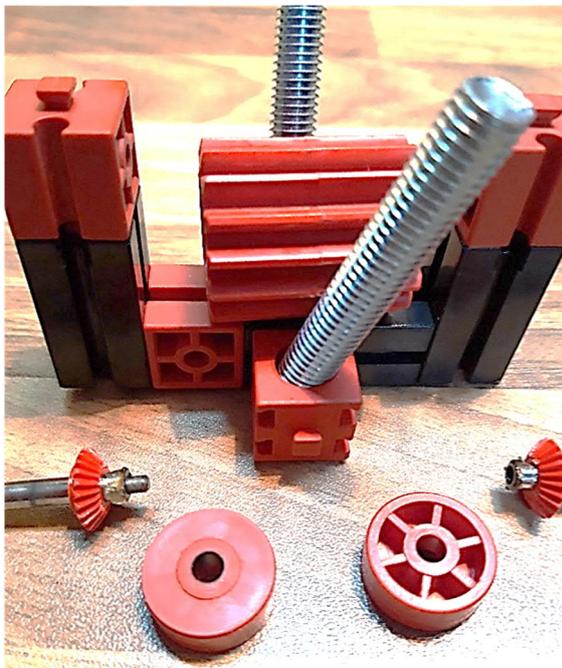


Abb. 3: Zerlegt

Ein bisschen gereinigt sehen die Teile schon etwas ansehnlicher aus. Zur weiteren Verwendung sollten sie aber erst mit warmem Wasser und Spülmittel komplett entfettet werden.



Abb. 4: Blick ins Innere

Dann sehen die Teile aber wie frisch aus dem Werk aus und können in den Fundus einsortiert werden. Das Mini-Speichenrad hat leider kein fischertechnik-Logo, aber dafür eine zugehörige ft-Teilenummer, und zwar die 35171. Diese glänzt hoffentlich in der ft-Datenbank nicht mehr lange mit Abwesenheit.



Abb. 5: Das Mini-Speichenrad

Aber was wäre ein frisch gebackenes fischertechnik-Bauteil ohne Anbaumöglichkeiten? Für irgendetwas anderes als nur eine Achsenhalterung muss man das Bauteil schließlich noch verwenden können.



Abb. 6: Gerade Anbaumöglichkeit

Der Durchmesser ist derselbe wie der des Deckels des neuen Differentials. Der Deckel hat die Teilenummer [35982](#), der neue Käfig die [31411](#). Aus diesem Grund passt das Mini-Speichenrad auch in den neuen Käfig. Was man damit dann bauen kann, beschreibe ich in einem separaten Artikel.

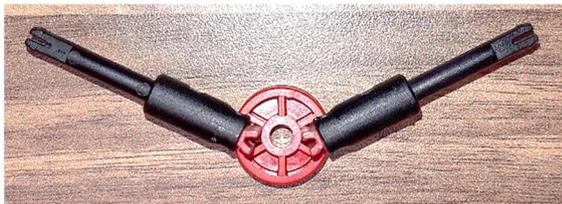


Abb. 7: Schräge Anbaumöglichkeit

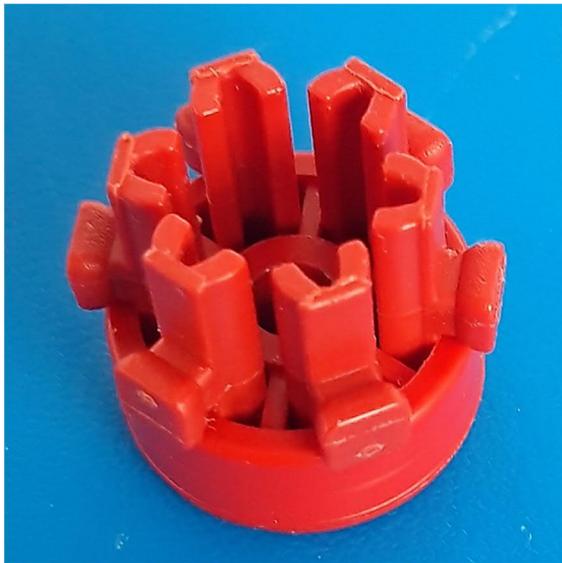


Abb. 8: Gefüllt mit Federnocken

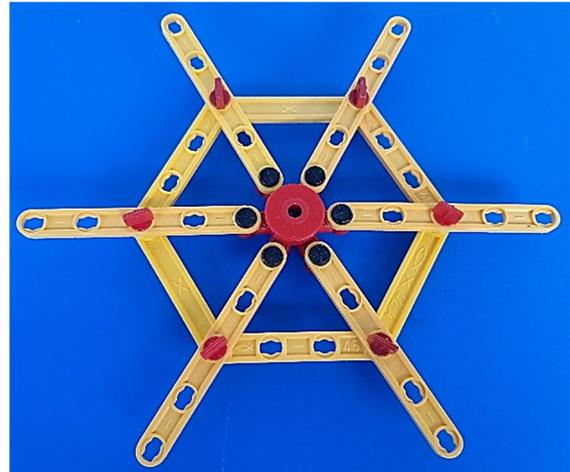


Abb. 9: Angebaute Streben



Abb. 10: Eine Detailansicht

Viel Spaß mit dem neuen Bauteil und viele frische Ideen wünscht euch euer Ralf.

Tipps &amp; Tricks

## Richtungsanzeige mit LEDs

Rüdiger Riedel

*Eine LED ist auch eine Diode, wie der Name sagt: Light-Emitting Diode. Wird sie falsch herum angeschlossen, dann leuchtet sie nicht. Aber hält sie das auch aus?*

Arbeitet man häufig mit den fischertechnik-LEDs, lässt sich eine Falschpolung nicht vermeiden. Auch durch den separaten (36708) Polwendeschalter und den am Batteriekasten (135719) befindlichen ist die falsche Polung schnell hergestellt. Doch zum Glück sind die LEDs robuster als befürchtet, und auch meine Bluetooth-Empfänger haben Umpolungen überstanden. Ich verwende LEDs von fischerfriendsman [1].



Abb. 1: LED mit Sockel [1]

Sollte doch mal ein Missgeschick passieren, gibt es preisgünstigen Ersatz. Einfach die defekte LED samt Widerstand aus dem Stecksocket herausziehen und die neue mit einer Spitzzange einfügen.



Abb. 2: Ersatz LED [1]

Aber das wird wahrscheinlich nicht passieren; alle meine Versuche mit falsch herum an 9 V Gleichstrom angeschlossenen LEDs haben dies klaglos überlebt.

Unsere fischertechnik-Motoren können durch Umpolen rechts- oder linksherum laufen. Schließen wir eine LED an die Motorklemmen an, dann zeigt sie die Laufrichtung an: Sie leuchtet bei der einen Drehrichtung, bei der anderen ist sie dunkel.

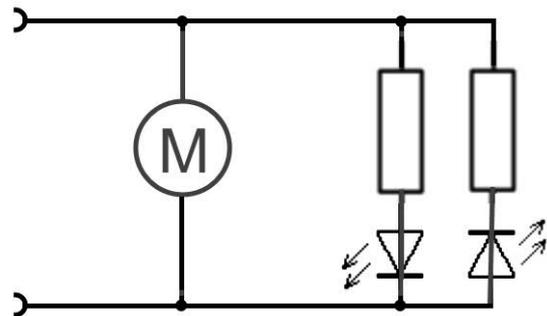


Abb. 3: Motor mit antiparallelen LEDs

Schalten wir zwei LEDs antiparallel an den Motor, leuchtet eine LED bei Rechtsdrehung, die andere bei Linksdrehung. Oder bei einem Fahrzeug eine in Vorwärtsrichtung und die andere bei Rückwärtsfahrt.

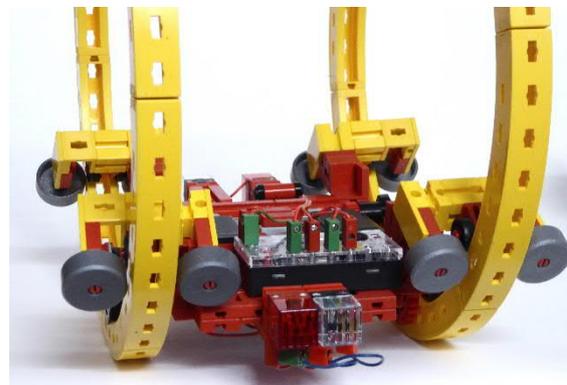


Abb. 4: Der kleine Hagelfeuer-Droide

In Abb. 4 habe ich den kleinen Hagelfeuer-Droiden [2] mit vier LED-Lampen ausgerüstet, zwei vorne und zwei hinten. In Fahrtrichtung leuchtet die weiße und gleichzeitig auf der Rückseite die rote LED.

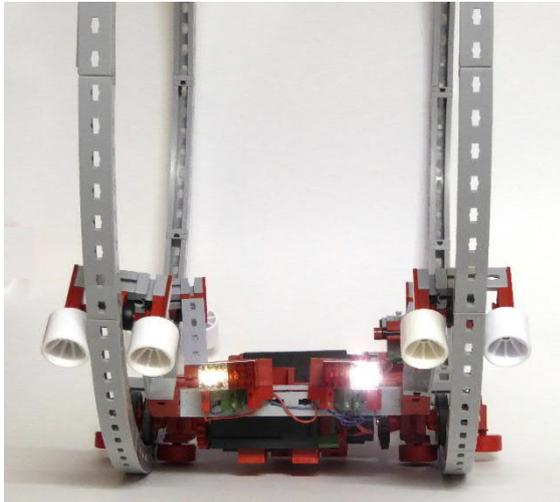


Abb. 5: Der große Hagelfeuer-Droide

Der große Hagelfeuer-Droide braucht natürlich Doppelbeleuchtung vorne und hinten.

### Hinweis

Alle beschriebenen Versuche habe ich mit LEDs von fischerfriendsman [1] vorgenommen; die Übertragbarkeit auf Original-fischertechnik-LEDs habe ich nicht überprüft, dies muss jeder selbst verantworten.

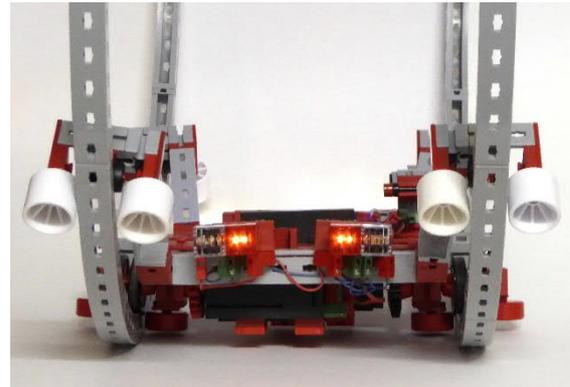


Abb. 6: Die Rücklichter

Noch ein Hinweis von fischerfriendsman: Die LEDs halten auch eine Zeitlang den Wechselstrom des alten grauen Trafos aus (30173). Aber nach einigen Stunden werden sie sichtbar schwächer. Die hohen Spannungsspitzen des Trafos in der negativen Welle sind einfach zu viel. Hier empfiehlt es sich, zum Schutz noch eine Standarddiode (z. B. 1N4001) in Flussrichtung vorzuschalten.

### Quellen

- [1] Webshop von fischerfriendsman.
- [2] Rüdiger Riedel: [Hagelfeuer-Droide](#). ftc-Bilderpool.

Getriebe

## Planetengetriebe (2)

Dirk Fox

*Unter den zahlreichen Getriebearten gibt es eine Kategorie mit ganz besonderer „Magie“ – die der Planetengetriebe, einer Sonderform der Umlaufgetriebe. Planetengetriebe sind heute aus keinem Fahrzeugantrieb mehr wegzudenken und erleben in Fahrrad-Nabenschaltungen einen zweiten Frühling. Im fischertechnik-System hielten sie vor 50 Jahren mit der Einführung des Innenzahnrad Z30 ([35694](#)) Einzug: Ein perfekter Anlass für eine Ergänzung des Beitrags von Thomas Püttmann aus der ft:pedia 2/2016 [1].*

### Hintergrund

Getriebe sind Maschinenelemente, die zwischen einem Antrieb und einem Abtrieb eine Bewegungsänderung vornehmen. Das können die Bewegungsart (z. B. Dreh- in Schubbewegung), die Bewegungsrichtung oder die Bewegungsgeschwindigkeit sein.

Zahnradgetriebe sind nur eine mögliche Getriebeform. Sie werden vor allem zur Änderung der Geschwindigkeit einer Drehbewegung eingesetzt – und damit zugleich zur Änderung des auf eine (Abtriebs-) Welle wirkenden Drehmoments (auf Kosten der Drehzahl). Sie sind damit Kraftverstärker. In Fahrzeugen spielen sie eine wichtige Rolle: Über ein *Schalt-* oder *Wechselgetriebe* bilden sie den (begrenzten) Drehzahlbereich des (Verbrennungs-) Motors, in dem dieser einen möglichst guten Wirkungsgrad aufweist, auf den Bereich der gewünschten Fahrgeschwindigkeit ab. Auch in Elektrofahrzeugen werden Getriebe benötigt, wenn auch nicht als Schaltgetriebe, da der Drehzahlbereich mit hohem Wirkungsgrad beim Elektromotor groß genug ist, sondern für eine Übersetzung ins Langsame.

Für Fahrzeuggetriebe wünscht man sich mehrere Eigenschaften, die Planetengetriebe gegenüber einfachen Stirnradgetrieben

auszeichnen: kompakte Bauweise, geringes Gewicht, koaxiale Lage von An- und Abtriebsachse und einen geräuscharmen Lauf.

### Planetengetriebe

Planetengetriebe sind Umlaufgetriebe, in denen sich Zahnräder umeinander drehen. Üblicherweise bestehen sie aus einem „Sonnenrad“ in der Mitte, um das mehrere (meist drei) „Planetenträger“ angeordnet sind, die wiederum in ein Innenzahnrad, das „Hohlrad“ greifen. Die Wellen (Achsen) der Planetenträger sind über einen Steg miteinander verbunden (Abb. 1) [3].

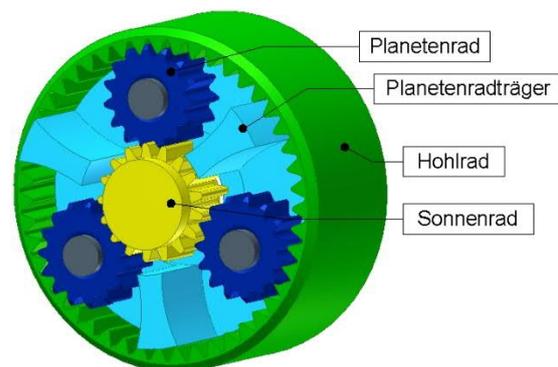


Abb. 1: Aufbau eines Planetengetriebes  
(Wikipedia, [CC BY SA 3.0](#))

Dabei können das Sonnenrad, der Steg (also die Drehung der Wellen der Planetenräder) oder das Hohlrad jeweils die Rolle des An- oder Abtriebs des Getriebes übernehmen.

Mit fischertechnik lassen sich Planetengetriebe seit der Einführung des Innenzahnrad Z30 (35694) in den Baukästen ut-2 und hobby2 im Jahr 1971 mit einem Z10 als Sonnenrad und zwei weiteren als Planetenräder konstruieren. Abb. 2 zeigt ein fischertechnik-Planetengetriebe mit der Welle des Sonnenrads als An- und der des Hohlrads als Abtrieb: Die Welle ist nicht durchgehend, sondern hinter dem Sonnenrad geteilt.



Abb. 2: Planetengetriebe (-1:3)

Wie man an der folgenden einfachen Umkonstruktion gut erkennt, ersetzt das Planetengetriebe aus Abb. 2 das Stirnradgetriebe in Abb. 3 durch eine kompakte koaxiale Übersetzung, bei der An- und Abtriebswelle in einer Flucht liegen.

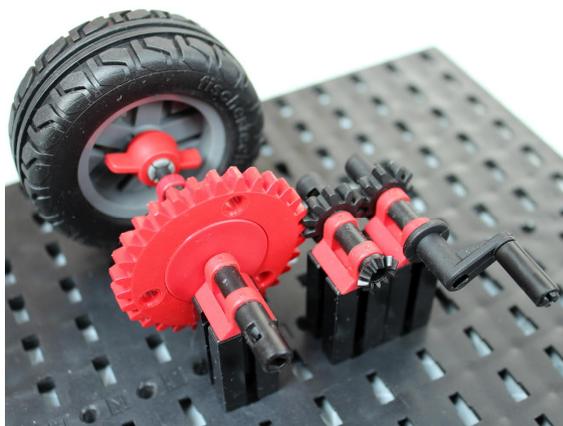


Abb. 3: Stirnradgetriebe (1:3)

Beide Getriebe übersetzen im Verhältnis 1:3 ins Langsame. Durch das Innenzahnrad kehrt sich beim Planetengetriebe allerdings die Drehrichtung der Abtriebswelle um.

Nun können wir abhängig von der Nutzung des Planetenrad-Stegs, des Sonnen- oder des Hohlrads als An- oder Abtrieb mit dieser Grundkonstruktion unterschiedliche Übersetzungen realisieren.

Im Hobby-Band 2-6 von fischertechnik aus dem Jahr 1977 findet man eine Übersicht der mit fischertechnik konstruierbaren Hohlrad-Planetengetriebe mit den zugehörigen Formeln zur Bestimmung des jeweiligen Übersetzungsverhältnisses (Tab. 1) [4].

Getriebeschema	fest Antrieb	Abtrieb	Übersetzungsverhältnis, allgemein.	Übersetzungsverhältnis, ft – Stirnrad-Planetengetriebe.	Anwendung zur
1.	3	1 P	$i = \frac{n_1}{n_3} = 1 + \frac{z_1}{z_3}$	$i = \frac{n_1}{n_3} = 4$	Drehzahlminderung
2.	3	P 1	$i = \frac{n_1}{n_3} = \frac{z_1}{z_1 + z_3}$	$i = \frac{n_1}{n_3} = \frac{1}{4} = 0,25$	Drehzahlsteigerung
3.	1	P 3	$i = \frac{n_1}{n_3} = \frac{z_1}{z_1 + z_3}$	$i = \frac{n_1}{n_3} = \frac{3}{4} = 0,75$	Drehzahlsteigerung
4.	1	3 P	$i = \frac{n_1}{n_3} = 1 + \frac{z_1}{z_3}$	$i = \frac{n_1}{n_3} = \frac{4}{3} = 1,33$	Drehzahlminderung
5.	P	1 3	$i = \frac{n_1}{n_3} = -\frac{z_1}{z_3}$	$i = \frac{n_1}{n_3} = -3$	Drehzahlminderung mit Drehrichtungs-umkehr.
6.	P	3 1	$i = \frac{n_1}{n_3} = -\frac{z_1}{z_3}$	$i = \frac{n_1}{n_3} = -\frac{1}{3} = -0,33$	Drehzahlsteigerung mit Drehrichtungs-umkehr.

Tab. 1: fischertechnik-Planetengetriebe mit Hohlrad (aus hobby 2, Band 6) [4]

Aus drei Rast-Z10 und einem Innenzahnrad Z30 konstruierte fischertechnik-Planetengetriebe sind überraschend leichtgängig. Abb. 4 zeigt die Variante mit festem Hohlrad, dem Sonnenrad als An- und dem Planetenrad-Träger als Abtrieb (Getriebeschema 1 in Tab. 1) – eine Übersetzung 1:4 ins Langsame.

Wählt man – unter Verzicht auf die koaxiale Anordnung der Wellen – statt des Stegs eine der Planetenrad-Wellen als Abtrieb, erhält man ein exzentrisches Rührgetriebe (Abb. 5).

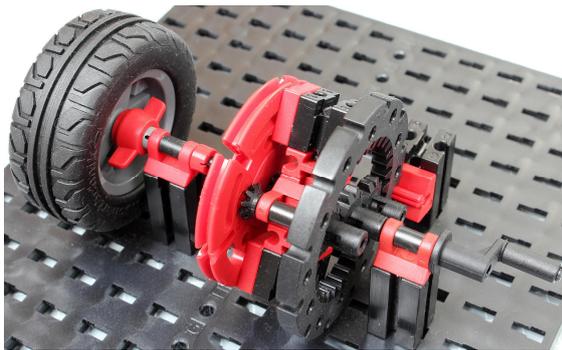


Abb. 4: Planetengetriebe (1:4)

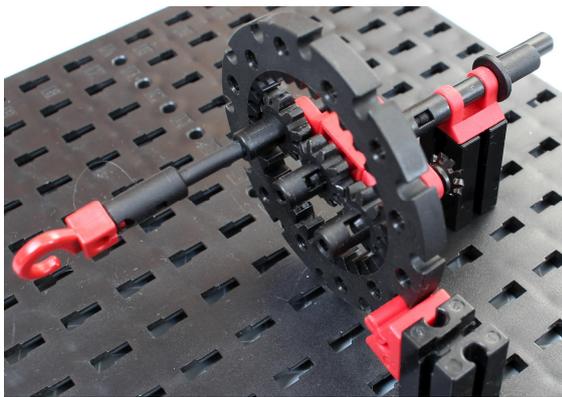


Abb. 5: Rührgetriebe

Schließlich können wir noch das Sonnenrad festhalten und den Planetenrad-Träger bzw. das Hohlräder als An- bzw. Abtrieb verwenden (Getriebeschema 3 in Tab. 1). Solche Planetengetriebe mit festem Sonnenrad kommen beispielsweise in Fahrradnabenschaltungen zum Einsatz. Abb. 6 zeigt die entsprechende Getriebekonstruktion. Man denke sich die Fahrradspitzen in die Nuten der rechten Drehscheibe 60 gesteckt.

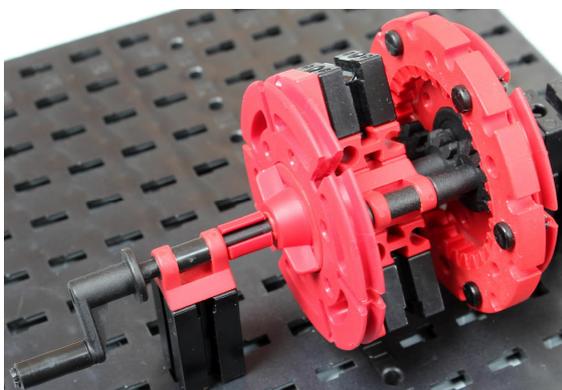


Abb. 6: Planetengetriebe (1:1,333)

Bei Planetengetriebebeschaltungen werden mehrere Planetengetriebe gekoppelt und beim Schalten die Funktion von Planetenradträger und Hohlräder als An- oder Abtrieb gewechselt. In der Regel benötigt man dafür ineinander laufende Hohlachsen. Die erste Zweigang-Nabenschaltung wurde 1895 von *Seard Thomas Johnson* entwickelt.

Unter Nutzung einer weiteren Variante des Planetengetriebes, nämlich einer festen Verbindung von An- und Abtriebswelle (Übersetzung 1:1), können wir auch mit fischertechnik das Funktionsmodell eines Zweigang-Schaltgetriebes konstruieren.

Ein solches Getriebe mit festem Hohlräder und einer Übersetzung von 1:1 bzw. 1:4 ins Langsame hat *Thomas Püttmann* 2016 vorgestellt [1]. Abb. 7 zeigt ein weiteres Zweigang-Schaltgetriebe. Es hält (wie bei Fahrradnabenschaltungen üblich) das Sonnenrad fest; das Hohlräder bildet den Abtrieb. Geschaltet wird, indem man die Welle so verschiebt, dass sich das Hohlräder von den Planetenrädern löst und über einen Mitnehmer (Verbindungsstopfen [32316](#)) zusammen mit dem Planetenrad-Steg dreht. Damit kann zwischen einer 1:1 und einer 1:1,333 Übersetzung umgeschaltet werden.

## Quellen

- [1] *Thomas Püttmann*: [Planetengetriebe](#). ft:pedia 2/2016, S. 38-43.
- [2] *Dirk Fox, Thomas Püttmann*: [Technikgeschichte mit fischertechnik](#). dpunkt-Verlag, 2015.
- [3] *Artur Fischer*: *fischer Elemente der Technik*. Heft 3, Bewegungsänderung. Fischer-Werke 1984.
- [4] *Artur Fischer*: *Planetengetriebe*. In: fischertechnik Hobby, Experimente und Modelle, hobby 2, Band 6, Fischer-Werke 1977, S. 30-33.

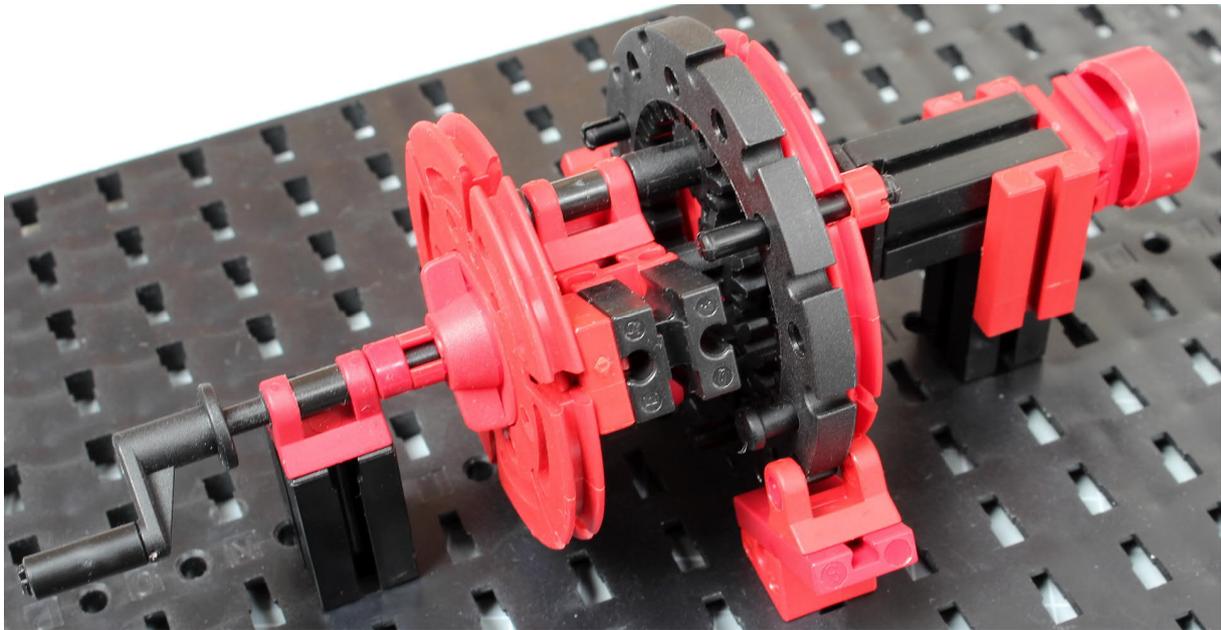


Abb. 7a: Zweigang-Planetenschaltgetriebe – Gang 1, Übersetzung 1:1,333

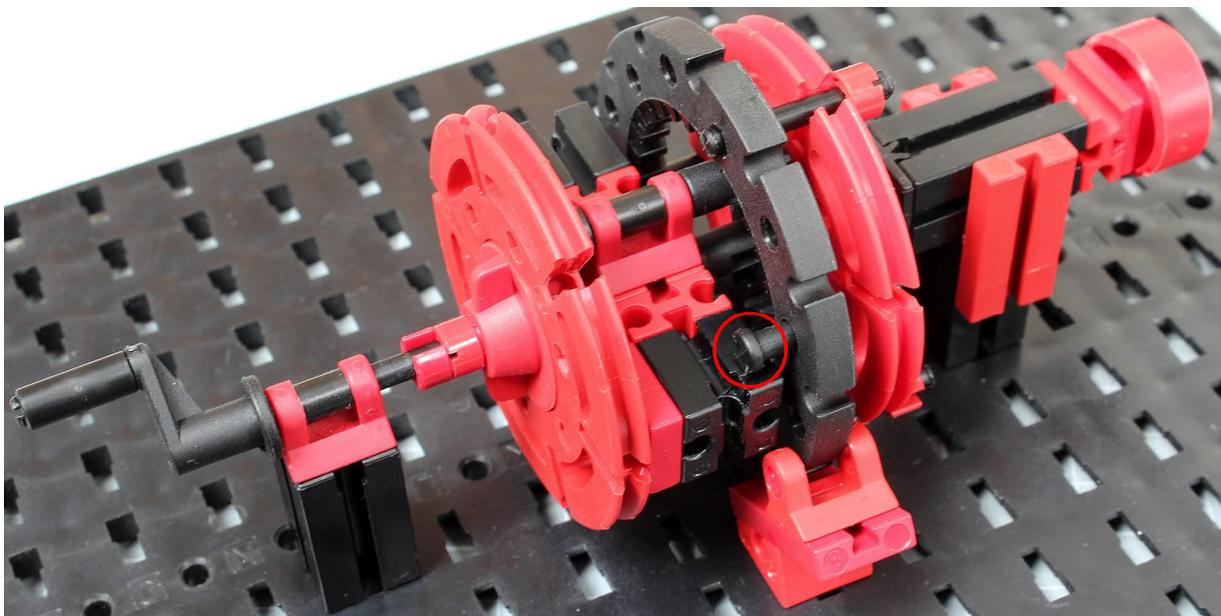


Abb. 7b: Zweigang-Planetenschaltgetriebe – Gang 2, Übersetzung 1:1 (roter Kreis: Mitnehmer)

Getriebe

## Uhr mit Planetengetriebe

Harald Steinhaus

*Die Beispiele im Bilderpool und in der ft:pedia zeigen es: Uhrenbau mit fischertechnik ist des Schweißes der Edlen wert.*

Nachdem ich mich nun schon recht lange mit Planetengetrieben herumschla... ääh, beschäftige, lag der Gedanke nahe, Uhrenbau und Getriebe zu verbinden. So soll nun hier eine Uhr mit Planetengetriebe entstehen. Die ausgewählte Bauart verlangt ‚exotische‘ Zahnräder in einer Vielfalt, die man von einem fischertechnik-Baukasten nicht erwarten kann. Aber für so etwas gibt es ja 3D-Drucker, darunter nicht zuletzt denjenigen von fischertechnik. Das hier zu Grunde liegende Paradigma ist anders als sonst üblich: nicht „wie kriege ich das mit den vorgegebenen Zahnrädern Z10, Z15, Z20, ... hin?“, sondern: „so so, da brauche ich ein Z57? Na gut, dann mache ich eins!“

Das Besondere an der Mechanik dieser Uhr ist der koaxiale Aufbau: Alle Elemente, vom Motor durch alle Stufen und Sätze hindurch, rotieren um eine zentrale Achse. Von außen gesehen hat die Uhr die Form eines Zylinders aus acht Scheiben, die abwechselnd drehfest fixiert sind oder mit nach vorn abnehmenden Geschwindigkeiten rotieren und diesem Fall die Zeiger tragen. Außerhalb des zylindrischen Körpers gibt es nur die Lagerung auf dem Sockel und eine Fixierung derjenigen Scheiben, die eben fixiert sein müssen.

### Aufbau: Stufen und Sätze

Den Antrieb soll einmal ein 50-Hz-Synchronmotor besorgen, wie er z. B. im Bilderpool gezeigt wird [1]. Die eigentliche Uhr kommt komplett aus dem 3D-Drucker.

Es gibt vier Untersetzungsstufen, die aus jeweils zwei Planetensätzen bestehen. Zwei Stufen werden gebraucht, um von der Motordrehzahl auf den Sekundenzeiger zu untersetzen. Von dort folgt eine Stufe auf den Minutenzeiger, und von da eine vierte auf den Stundenzeiger. Mein 3D-Drucker mit 0,35 mm Düse schafft auch brauchbare Zahnräder im Modul  $m=1$ , was Material und Bauzeit einsparen würde. Mit Blick auf den fischertechnik-3D-Drucker mit 0,4-mm-Düse wird hier aber der fischertechnik-Standard mit Modul  $m=1,5$  verwendet.

Planetengetriebe (allgemeiner: Umlaufgetriebe) führen bei fischertechnik eher ein Schattendasein, mit begrenzten Möglichkeiten um das Z30-Innenzahnrad und die verschiedenen Differenziale herum. Die Grundlagen sind aber in im fischertechnik-Band hobby 2 [2] und den ft:pedia-Artikeln von Thomas Püttmann [3, 4] beschrieben. Für große Übersetzungsverhältnisse werden einfache Planetengetriebe recht sperrig. Wenn aber zwei Planetensätze miteinander gekoppelt werden, eröffnen sich viele Möglichkeiten. Eine bestimmte Form der Koppelung macht aus je zwei Planetensätzen eine Wolfram-Stufe.

### Startpunkt: Wolfram-Planetenstufen

Die englische Wikipedia zeigt den Verstellmechanismus eines PKW-Außenspiegels mit zwei, sagen wir mal, „unvollständigen“ Planetensätzen übereinander (Abb. 1). Das

weiße Sonnenrad wird angetrieben, das weiße Hohlrad ist mittels der Noppen am Umfang an seinem Platz fixiert. Das schwarze Hohlrad gehört zum zweiten Planetensatz. Die Planetenräder sind immer zu zweit ‚huckepack‘ zu einem einzigen Bauteil verbunden und koppeln beide Planetensätze: Die untere Seite rollt auf dem weißen Hohlrad ab und nimmt die obere Seite mit. Diese treibt das schwarze Hohlrad an. Es bildet den Getriebeausgang, der eine Gewindestange umgreift und damit den Spiegel in einer Achse verstellt.

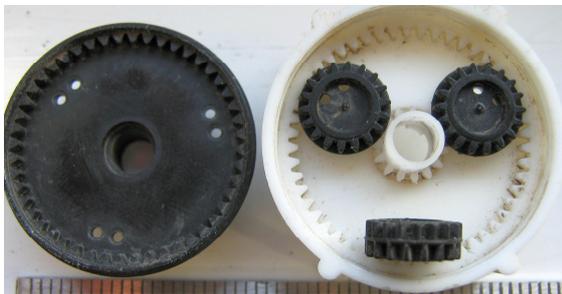


Abb. 1: Außenspiegelverstellung eines PKW  
(Wikipedia, [CC BY SA 3.0](https://creativecommons.org/licenses/by-sa/3.0/))

Die Fachliteratur kennt diesen Aufbau als Wolfrom-Getriebe, Zweisteggetriebe, oder reduziertes Planetengetriebe. Diese Bauart ermöglicht große Untersetzungsverhältnisse in kleinem Bauraum. Das ‚reduzierte‘ daran heißt, dass Teile weggelassen wurden, ganz in Anwendung des Grundsatzes „Perfektion ist nicht dann erreicht, wenn man nichts mehr hinzufügen, sondern nichts mehr weglassen kann“ (Antoine de Saint-Exupéry), oder mit Artur Fischer: „so einfach wie möglich“:

- Der Ausgang jeder Stufe erfolgt über das Hohlrad des zweiten Planetensatzes; dieser braucht kein Sonnenrad.
- Die Planeten beider Sätze sind als gemeinsame Körper (Päckchen) ausgeführt. Sie müssen nicht auf eigenen Planetenträgern gelagert werden.
- Die Planetenträger kann man sich als zunächst vereint (daher ‚Zweisteggetriebe‘) vorstellen, bis sie dann komplett

weggelassen wurden. Die Planetenpäckchen werden nunmehr nur zwischen den Hohlradern geführt.

- Es gibt im gezeigten Beispiel auch keine Lager. Das liegt aber nur am Anwendungsfall: ein Außenspiegel wird über die Lebensdauer des zugehörigen Autos keine nennenswerte Zahl von Betriebsstunden erreichen, und sollte er einfrieren und zerbrechen, freut sich der Ersatzteihändler...

## Nomenklatur

Mangels Normierung (böse Zungen behaupten, das Gute an Standards sei, dass es so viele davon gibt) verwende ich die Bezeichnungen, die in meiner Literaturstichprobe am „üblichsten“ sind. Es sind:

- A, ZA: das Sonnenrad und die Zahl seiner Zähne (außen verzahnt)
- B, ZB: das Hohlrad und die Zahl seiner Zähne (innen verzahnt, in der Literatur manchmal als negative Zahl verwendet, hier immer positiv)
- C, ZC: das Planetenrad und die Zahl seiner Zähne (außen verzahnt). „C“ kann man sich ableiten aus dem englischen „Carrier“, das den Planetenträger oder Steg bezeichnet.

Im Weiteren werden die Bauteile bezeichnet mit einer Zeichenfolge aus Nummer des Planetensatzes + Art des Zahnrads + Anzahl der Zähne.

Zum Beispiel:

- 1A16 ist das Sonnenrad (A) von Planetensatz 1, mit ZA = 16 Zähnen.
- 3C15 ist ein Planet (C) mit ZC = 15 Zähnen aus dem Satz 3.
- 4-2C14 ist ein Planet C14 aus dem Satz 2, der baugleich im Satz 4 verwendet wird.

Bei Letzterem liegt der tiefere Sinn darin, dass baugleiche Teile schon im CAD-

Programm als weitere Instanzen des gleichen Elements (in FreeCAD als ‚Links‘ bezeichnet) verwaltet werden können. Man kann sich selbst hereinlegen, wenn man nur solch einen Ableger ändern wollte, aber unerkant und ungewollt die verbundenen Teile mit ändert.

Bei üblichen Zahnrädern stehen die Zähne um ‚m‘ mm (bei Modul m=1,5 also 1,5 mm) über den Wälzkreis, und der Zahngrund liegt um ‚m‘ plus etwas ‚Luft‘ darunter. Am Wälzkreis entlang sind Zahn und Zahn um eine Bogenlänge von  $m \cdot \pi$  mm voneinander entfernt. Der Wälzkreis ist an einem gegebenen, fertigen Exemplar nicht sichtbar und i. d. R. nicht sauber messbar. Es ist aber hilfreich, den Wälzkreis beim Entwurf und beim Drucken mitzuführen: die Anzahl der Zähne ist proportional zum Durchmesser eines Zahnrads, also ist es egal, ob man in Zähnen oder in Wälzkreisdurchmessern rechnet.

Bei jeder Zahnpaarung müssen sich die Wälzkreise der beteiligten Zahnräder exakt berühren. Beim 3D-Drucken mit Haftgrund („Brim“, verhindert das Abheben des Teils durch thermischen Verzug) definiert man geschickter Weise im CAD-Programm einen flachen Ring, der zum Zahnrad gehört, etwas größer ist und von außen bis zum Wälzkreis reicht. Den Haftgrund braucht man sowieso; er soll nicht in den Zahngrund hinein reichen (das nachher zu entfernen ist eine üble Popelei), und im CAD-Programm sieht man sofort, wenn etwas nicht stimmt. Man liegt z. B. um einen Faktor 2 daneben, wenn man in Durchmessern rechnet und das CAD-Programm mit Radien gefüttert werden will.

**Wie viele Zählerl hätten’s denn gern?**

Für jeden Planetensatz gelten die Bedingungen:

$$ZA + 2 \cdot ZC = ZB \quad (1)$$

Lies: die Zahnräder müssen ineinander passen. Anschaulich: ein Schnitt durch den

Planetensatz (B) kann zerlegt werden in den Sonnendurchmesser (A) und zweimal den Planetendurchmesser ( $2 \cdot C$ ). Das gilt für die Zähne in gleicher Weise.

$$(ZA + ZB) \bmod NC = 0 \quad (2)$$

Das ist die erste sogenannte „Einbaubedingung“: Die Summe von ZA und ZB ist ganzzahlig teilbar durch die Anzahl der Planeten; die Planeten können rotations-symmetrisch um die Sonne herum montiert werden. Regel (2) ist aber belanglos, wenn man nur einen Planeten pro Satz verwendet und den unrunder Lauf in Kauf nimmt.

Die zweite Einbaubedingung fordert, dass sich die Planeten eines Satzes nicht untereinander in die Quere kommen dürfen.

	A	B	C	BW	BX	BY	BZ	CA	CB	CC	CD	CE	CF	CG	CH
1		m=1		ZB				ZB				ZB			
2				52				53				54			
3	ZC	Wälz-D		ZA	md4	md3	A+C	ZA	md4	md3	A+C	ZA	md4	md3	A+C
10	14	14		24	0	1	38	25	2	0	39	26	0	2	40
11	15	15		22	2	2	37	23	0	1	38	24	2	0	39
12	16	16		20	0	0	36	21	2	2	37	22	0	1	38
13	17	17		18	2	1	35	19	0	0	36	20	2	2	37
14	18	18		16	0	2	34	17	2	1	35	18	0	0	36
15	19	19		14	2	0	33	15	0	2	34	16	2	1	35
16	20	20		12	0	1	32	13	2	0	33	14	0	2	34
17	21	21		10	2	2	31	11	0	1	32	12	2	0	33
18	22	22		8	0	0	30	9	2	2	31	10	0	1	32
19	23	23		6	2	1	29	7	0	0	30	8	2	2	31
20	24	24		4	0	2	28	5	2	1	29	6	0	0	30
21	25	25		2	2	0	27	3	0	2	28	4	2	1	29

Abb. 2: grüne Felder zeigen an, dass ein Planetensatz aus ZA, ZB, ZC mit  $NC = 4$  oder  $NC = 3$  möglich ist.

Das Tabellenblatt in Abb. 2 errechnet die nach Bedingung 1 möglichen Werte von ZA und zeigt für die Kombination aus ZC (blau, Spalte A), ZB (orange/helle Balken, vier Spalten je Wert) und ZA (unter dem gewählten ZB, gelb/grau), ob die erste Einbaubedingung für ein NC von 3 oder 4 erfüllt ist. Dann zeigt die Spalte „modulo“ (md3 oder md4) eine Null auf grünem Hintergrund. Die markierte Zelle zeigt, dass ein Planetensatz mit ZA = 19, ZB = 53 und ZC = 17 die erste Einbaubedingung für vier Planeten erfüllt.

Die zweite Einbaubedingung wurde ‚freihändig‘ überprüft. Die hier verwendeten Planetensätze sind alle für  $NC = 4$  ausgelegt.

Die Bedingungen (1) und (2) sind grundlegend für alle Planetengetriebe und stehen in allen Büchern. Für Wolfram-Getriebe kommt für alle Planetensätze aus den Stufen  $i$  und  $j$  hinzu:

$$iNC = jNC \quad (3)$$

Die Planeten der Sätze einer Stufe werden huckepack aufeinander montiert. Bei Wegfall des Steges geht das nur mit gleicher Anzahl und Position der Planeten.

Aus praktischen Erwägungen wurden noch weitere Bedingungen gesetzt:

$$B \leq 65 \quad (4)$$

für alle Sätze. Bei Modul  $m=1,5$  erreicht der Wälzkreis einen Durchmesser von 90 mm, dazu kommen noch Zahngrund und etwas „Fleisch“ für die Außenhülle. Ursprünglich hatte ich mit Modul  $m=1$  die 60 mm Wälzkreisdurchmesser passend zur fischertechnik-Drehscheibe angepeilt.

$$A \geq 10 \quad (5)$$

für alle Sätze, die überhaupt ein Sonnenrad haben. Mit weniger als 10 Zähnen auf einer fischertechnik-Achse mit 4 mm Durchmesser wird man mit PLA-Filament nicht glücklich werden.

Vorgesehen ist ein 50-Hz-Synchronmotor mit 12 Magneten. Er braucht 120 ms für eine Umdrehung. Bisher habe ich aber noch keinen dieser Motoren zuverlässig zum Laufen bekommen. Derzeit ist ein Power-Motor verbaut, der ohne echten Zeitbezug oder Regelung drauflos dreht. Er ließe sich synchronisieren, wenn man noch einen Controller oder Arduino und weitere Teile spendiert [5, 6].

Von hier ausgehend, braucht die Uhr:

- eine Übersetzung 1:500 (120 ms / 60.000 ms), um den Sekundenzeiger anzutreiben. Der darin steckende Faktor 5

ist bereits Gegenstand vieler Diskussionen und mit fischertechnik-Mitteln nicht „einfach mal so“ zu bekommen (eine Lösung mit einem Umlaufgetriebe findet sich in [3]). Wolfram-Getriebe mit maximal  $B = 60$  erreichen knapp unter 1:300. Also wird diese Übersetzung geteilt in zwei Stufen von 1:25 und 1:20 (Stufen 1 und 2, Sätze 1+2 und 3+4).

- darauf folgend eine Übersetzung 1:60 für den Minutenzeiger. Das erledigt eine dritte Wolframstufe mit den Planetensätzen 5 und 6.
- danach eine Übersetzung 1:12 für den Stundenzeiger, aus der Stufe 4 mit den Sätzen 7 und 8.

## Übersetzungen mit Wolfram-Stufen

Die Übersetzung einer Wolfram-Stufe ergibt sich nach Blick in die Literatur und wenigen elementaren Umformungen (Entschuldigung – das war seinerzeit der ‚running gag‘ in vielen Vorlesungen) zu:<sup>1</sup>

$$\dot{U} = \frac{1 + \frac{B1}{A1}}{1 - \frac{B1 \cdot C2}{C1 \cdot B2}}$$

Diese Formel ist hergeleitet nach [7] und so umgeformt, dass nur noch  $A1$ ,  $C1$ ,  $C2$  frei wählbar sind. Weitere Details bedeuten einen Ausflug in die Tiefen des höheren Maschinenbaus. Interessierte Leser seien dazu auf die Literatur [7, 8, 9] verwiesen.

Bei der Wahl der Zahnräder müssen natürlich die oben aufgeführten Bedingungen allesamt erfüllt werden. Die möglichen Übersetzungen haben überwiegend „krumme“ Werte, und unter den ganzzahligen Ergebnissen sind die hier gesuchten Werte spärlich, manchmal aber auch mehrfach zu finden.

gehen und die Zellen  $A1$  bis  $C2$  als Eingabefelder verwenden.

<sup>1</sup> Hier habe ich die Nomenklatur umgedreht und die Buchstaben nach vorn gestellt. Damit kann man direkt in ein Tabellenkalkulationsblatt

A1	C1	C2	B1	A2	B2	Ü
12	24	9	60	27	45	12
15	20	10	55	25	45	12
21	21	12	63	30	54	12
25	15	10	55	30	50	12
9	24	12	57	21	45	20
10	25	13	60	22	48	20
12	24	14	60	22	50	20
15	21	14	57	22	50	20
27	15	12	57	30	54	20
10	25	15	60	20	50	25
14	21	15	56	20	50	25
18	18	14	54	22	50	25
18	15	12	48	21	45	25
26	13	11	52	28	50	25
9	25	20	59	14	54	60
11	24	20	59	15	55	60
19	20	18	59	21	57	60
29	15	14	59	30	58	60
10	25	24	60	11	59	295
9	27	26	63	10	62	372
8	24	17	56	15	49	42
21	20	22	61	19	63	-60

Abb. 3: ausgewählte Übersetzungsverhältnisse mit Wolfram-Sätzen, grüne Farbtöne verdeutlichen die Maße der Hohlräder

Abb. 3 zeigt eine Liste von Kandidaten und anderen interessanten Werten. Die höchsten Übersetzungsverhältnisse werden erreicht, wenn sich C1 und C2 nur um 1 unterscheiden. Bei C1 = C2 blockiert das Getriebe und die Formel läuft auf eine Division durch Null. Wenn C2 größer ist als C1, wird die Übersetzung negativ (d. h. die Drehrichtung am Ausgang kehrt sich um).

Die Wahl fiel auf Planetensätze mit den Werten:

- Stufe 1: 1A18, 1B54, 1C18; 2B50, 2C14; Sätze 1, 2, 3, 4 ergeben  $\ddot{U} = 25 \cdot 20 = 500$
- Stufe 2: 3A15, 3B57, 3C21; 4-2B50, 4-2C14;  $\ddot{U} = 20 \rightarrow$  Sekundenzeiger
- Stufe 3: 5A29, 5B59, 5C15; 6B58, 6-2C14;  $\ddot{U} = 60 \rightarrow$  Minutenzeiger
- Stufe 4: 7A25, 7B55, 7C15; 8-2B50, 8C10;  $\ddot{U} = 12 \rightarrow$  Stundenzeiger

7C15 könnte zu 7-5C15 und damit baugleich mit 5C15 gemacht werden, wenn man einen Adapter für die Aufnahme des viel kleineren Schafts seines Partner-Planetensatzes 8C10 erstellt.

## Zusammenbau

Neben fischertechnik-Standardteilen werden vier Kugellager 4x12x4 benötigt, wie sie auch in die fischertechnik-Schneckenmutter m=1,5 hinein passen. Der Synchronmotor hätte 12 Power-Magnete, wird aber hier von einem Power-Motor vertreten.

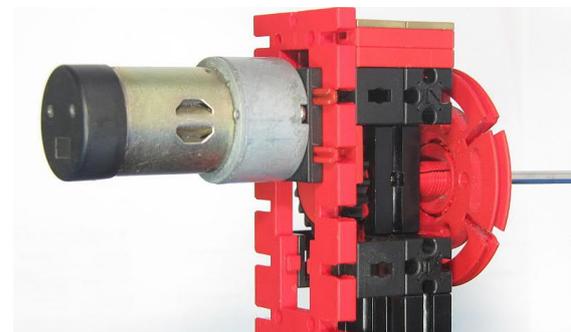


Abb. 4: Eine Prinz gewordene Kaulquappe: BSB-Grundplatte ([31557](#))

### 1. Zeiger montieren

Die Zeiger dieser Uhr sind nur als kleine Stummel (bei „12 Uhr“) und als Schwalbenschwanz-aufnahme bei „6 Uhr“ der ‚geraden‘ Hohlräder ausgebildet. Dort kann man je nach Geschmack weiter verschönerte Gebilde anbringen, etwa bunt eingefärbte Zahnstocher. Die Zeiger müssen flach genug sein, damit sie zwischen den Halteklauen für die ‚ungeraden‘ Hohlräder hindurch passen, und kurz genug, damit sie nicht in die Rastvorrichtungen geraten.

### 2. Lager in die Taschen der Sonnenräder einbauen

Kugellager haben deutlich engere Toleranzen als fischertechnik-Stahlachsen, d. h. man muss eine geeignete Stahlachse aussuchen. fischerfriendsman.de hat beides im Angebot. Die Hohlräder werden über ihre runde Mittelöffnung direkt auf den Sonnenrädern gelagert. Die Sonnenräder schließen

in Richtung der Mittelachse bündig aneinander an.

### 3. Sonnen- und Hohlräder der Wolfromstufen montieren

Man kann alle Zahnräder „einfach so“ hineinstecken bzw. -legen. Das liegt an der Vereinfachung auf nur einen Planeten pro Satz ( $NC = 1$ ). Der allgemeinere Fall ( $NC > 1$ ) führt zu Komplikationen, auf die weiter unten eingegangen wird. Hier aber ist es egal, wo man die Planeten platziert; den Rest erledigt sowieso der Stellmechanismus.

Das Sonnenrad 1A18 sitzt ganz vorne und wird von der Stahlachse über eine fischertechnik-Klemmbuchse angetrieben. Für die anderen Sonnenräder 3A15, 5A29, 7A25

gilt: Das Rad geht mit seinem Schaft zuerst durch das runde Loch in ‚seinem‘ Hohlrad und wird vom Sechskant im weiter ‚vorn‘ liegenden Hohlrad angetrieben. Man steckt also zuerst 3A15 durch 3B57 hindurch und klemmt dann 2B50 darauf. Ebenso geht 5A29 durch 5B59 hindurch und klemmt in 4-2B50. Ein drittes Paket entsteht aus 7A25, 7B55 und 6B58.

### 4. Hohlrad 1B54 montieren

Das Hohlrad 1B54 wird mit einer Klemmbuchse versehen und mit Achsen 30 an der fischertechnik-Drehscheibe befestigt.

### 5. Planetenpacks zusammenbauen

1C18 mit 2C14, 3C21 mit 4-2C14, 5C15 mit 6-2C14, 7C15 mit 8C10.

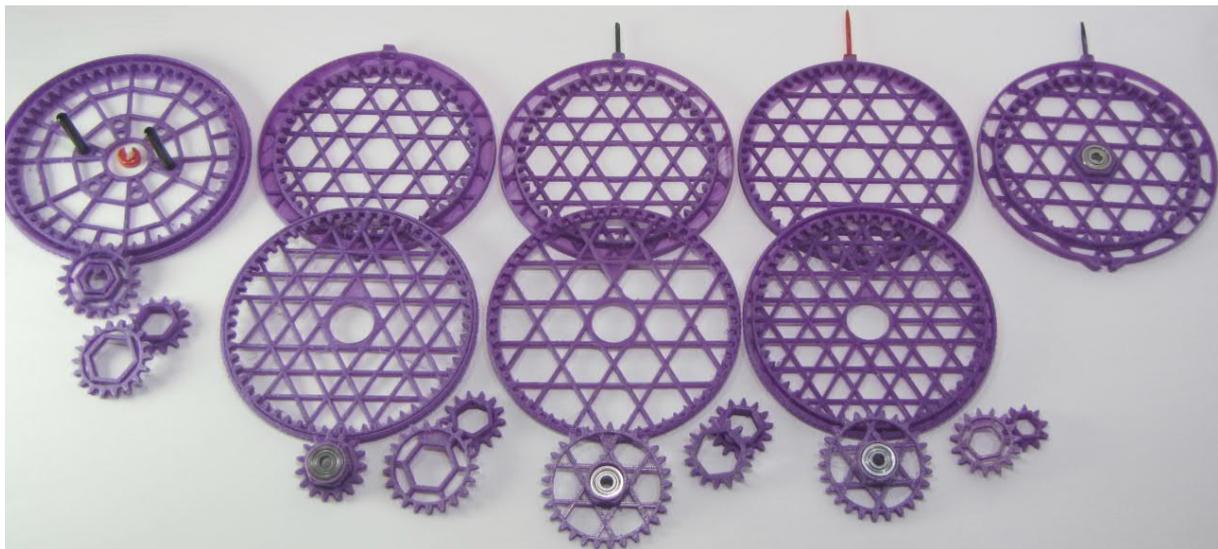


Abb. 5 Die gedruckten Teile und Baustufen: Die Sonnenräder mit Lager steckt man durch die Hohlräder mit runden Löchern (untere Reihe) und dann von der Rückseite in diejenigen mit eckigen Löchern.



Abb. 6: Das eingebaute Hohlrad 1B54 mit 1A18

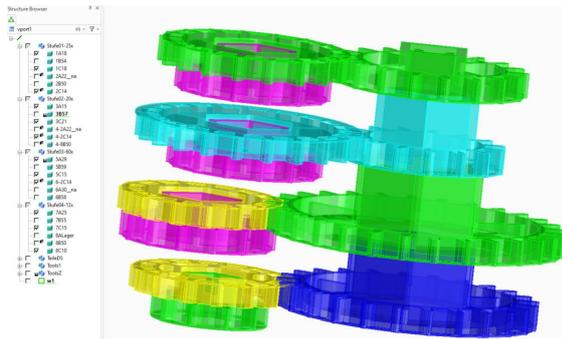


Abb. 7: Die Innereien der Uhr. Rechts ergeben die Sonnenräder einen Stapel, mit 1A18 oben. Links die Planetenpäckchen, bei denen immer das kleinere zum Ausgang hin liegt. 1A18 ist in älterer Bauform gezeigt; 8B50 ist mittlerweile baugleich mit 2B50 und wurde zu 8-2B50. Die ‚geraden‘ Sonnenräder wurden nur zur Kontrolle konstruiert.

## 6. Endmontage

Jedes Päckchen kommt mit dem größeren Planeten nach unten in dessen Hohlrad, und der somit vervollständigte Planetensatz wird auf die Achse aufgeschoben. Zuletzt kommt 8-2B50 mit seiner Lagerbuchse. Eine fischertechnik-Flachnabe obenauf hält den Stapel zusammen.

## Stellen der Uhr

Zum Stellen der Uhr kann jeder Zeiger einzeln verdreht werden: man löst die Raste (Kardanhülse) am ‚ungeraden‘ Hohlrad,

bringt das ‚gerade‘ Hohlrad mit dem Zeiger auf Wunschposition und lässt die Raste wieder eingreifen. Das Hohlrad 1B54 hat zwar auch Rastungen, wird aber hier mittels zweier Achsen 30 fest mit einer fischertechnik-Drehscheibe verbunden.

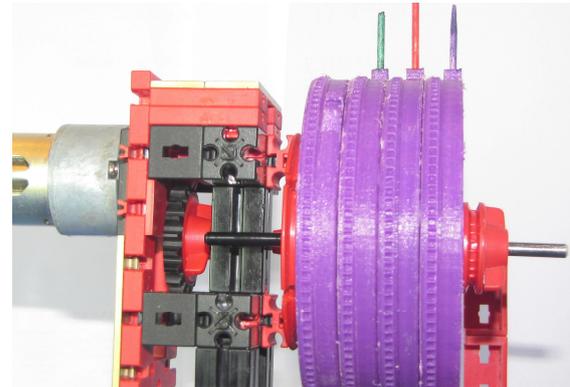


Abb. 8: die BSB Lok Grundplatte 90x45 trägt den P-Motor

Die Uhr kann im laufenden Betrieb gestellt werden. Planetengetriebe sind mechanische Addierwerke für Drehwinkel und -geschwindigkeiten. Für den Lauf der Uhr ist es gleichgültig, ob die Hohlräder stehen oder sich drehen.

Ein stehendes Hohlrad addiert einen konstanten Winkel (multipliziert mit dem Übersetzungsverhältnis des Planetensatzes für den Fall ‚Antrieb über Hohlrad‘) und eine Geschwindigkeit von 0°/s.

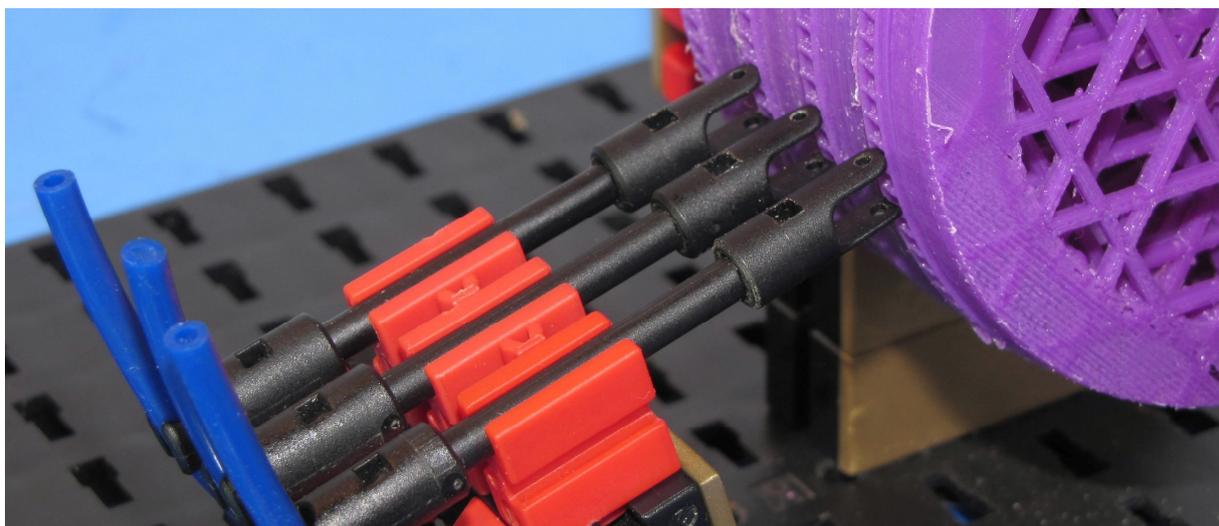


Abb. 9: die Rastungen werden mittels P-Schläuchen angedrückt. Auf dem Umfang sind Mulden so verteilt, dass die Kardanhälften hinein passen. Dieses Maß ergibt „krumme“ 112 Mulden.

Gleichermaßen wird auch die Umstellung auf Sommer/Winterzeit durch einen festen Versatz am Hohlrad von Satz 8 erledigt. Genauso könnte man eine Gangkorrektur und Funktionen wie Zeitlupe und Zeitraffer durch Antriebe an den ‚ungeraden‘ Hohlrädern hinzufügen.

### **Disclaimer: Druckgenauigkeit**

Mein 3D-Drucker hat eine Düse von 0,35 mm. Der fischertechnik-Drucker und die meisten übrigen Drucker haben 0,4 mm. Wandstärken und andere Maße in den STL-Dateien sind ‚über den Daumen‘ auf den fischertechnik-Drucker angepasst, d. h. Wandstärken sind 0,8 oder 1,2 mm entsprechend zwei oder drei Druckbahnen. Bei den Radien für die Lagersitze habe ich einen Aufschlag von 0,25 mm verwendet, mit dem mein Drucker Lagersitze produziert, die nach Entgraten mit Sandpapier direkt verwendbar sind.

### **Was ist mit NC > 1?**

Die Uhr wurde ursprünglich für  $NC = 4$  Planeten pro Satz konzipiert. Daraus ergaben sich weitere Definitionen und Konventionen:

1. Bei Uhrzeit 0:0:0:0 haben alle Planetensätze einen ihrer Planeten mit Zähindex 0 (etwa 3C21.0) im ‚Norden‘ („12 Uhr“-Position), mit einem Zahn ebendort. Gezählt wird im Uhrzeigersinn bei Draufsicht auf die fertige Uhr. Beispiel: Hohlrad 2C18.3 ist der vierte Planet (Index 3) von Satz 2. Im CAD-Programm tut man sich leichter, wenn man alle Zahnräder im Koordinatenursprung belässt. Aber beim Drucken und beim Einbau liegen die Sonnenräder und die ‚geraden‘ Hohlräder in der Gegenrichtung.
2. Alle Hohlräder haben eine Zahnücke im Norden (folgt aus 1.).
3. Die Planeten sind rotationssymmetrisch um die Sonne herum angeordnet. Bei unsymmetrischer Anordnung der Planeten

müssten alle Abweichungen von der Symmetrie in gleicher Weise für beide Sätze einer Wolfram-Stufe eingehalten werden -- das muss man nicht haben. Die Planetensätze einer Stufe müssen gleich viele Planeten haben.

4. Die Sonnenräder stehen so, wie es der Planet Cxx.0 des jeweiligen Satzes bestimmt. Ist NC gerade, dann hat der Planet mit Index 0 einen Zahn im Süden und die Sonne eine Zahnücke im Norden.

5. Mit  $NC = 4$  werden weitere Planeten um jeweils  $90^\circ$  versetzt eingebaut. Dort können Sonne und Hohlrad Zahn oder Lücke zeigen oder irgendwo dazwischen stehen.

6. Die Einbauregeln stellen sicher, dass diese weiteren Planeten pro Satz lagerichtig eingebaut werden können. Es kann aber für jedes Päckchen unterschiedlicher Versatz (Drehwinkel) zwischen den beiden Planeten nötig sein, d. h. die vier Planetenpäckchen einer Stufe sind nicht notwendiger Weise baugleich. Sie sind nur für ihren Platz geeignet; sie müssen entweder jedes für sich konstruiert und gedruckt werden, oder sie werden erst beim Einbau lagerichtig zueinander fixiert (sprich: geklebt) und gekennzeichnet. Das eröffnet reichlich Fehlermöglichkeiten und geht nicht mit den hier verwendeten sechseckigen Kupplungsstücken.

Mit dieser Betrachtung schließt sich der Kreis zum obigen Beispiel aus Wikipedia: Bei genauer Untersuchung erkennt man auf Abb. 1 jeweils zwei Löcher in den Hohlradkörpern und in den Planetenpacks. Diese Löcher sind Markierungen für den lagerichtigen Einbau: sie müssen deckungsgleich liegen (d. h. Licht scheint hindurch), dann sind die Planeten nach Drehwinkel und nach Position im Hohlrad richtig angeordnet. Das abgebildete Getriebe ist nicht richtig montiert.

## Wer die Herausforderung ...

... sucht, untersucht die Machbarkeit einer ‚complication‘ wie etwa des Modells ‚UR-202‘ der Marke ‚Urwerk‘ mit seinen Umlaufgetrieben [10]. Manchmal ist es nur eine Frage des Maßstabs, ob ein Gerät ein Uhrwerk oder eher ein Kirmes-Fahrgeschäft ist.

## Links und Referenzen

- [1] Bilderpool der ftCommunity: [Uhren](#).
- [2] fischerwerke: *Planetengeriebe*. In: hobby2 Band 6, S. 30 ff
- [3] Püttmann, Thomas: [Planetengeriebe](#). ft:pedia 2/2016, S. 38-43.
- [4] Püttmann, Thomas: [Das Differentialgetriebe](#). ft:pedia 4/2014, S. 12-19.
- [5] Falk, Stefan: [Mikro-Uhr mit kontinuierlich geregeltem Gleichstrommotor](#). Bilderpool der ftCommunity.
- [6] Birkenstock, Gerhard: [Power-Synchronmotoren](#). ft:pedia 3/2017, S. 42-45.
- [7] Looman, Johannes: *Zahnradgetriebe*. Springer Verlag, 2. Auflage 1988.
- [8] Mulzer, Florian: [Systematik hochübersetzender Koaxialgetriebe](#). Dissertation, TUM 2010.
- [9] Volmer, Johannes: *Umlaufträgergetriebe*. VEB Verlag Technik Berlin, 4. Auflage 2001.
- [10] Urwerk: [UR-202](#).

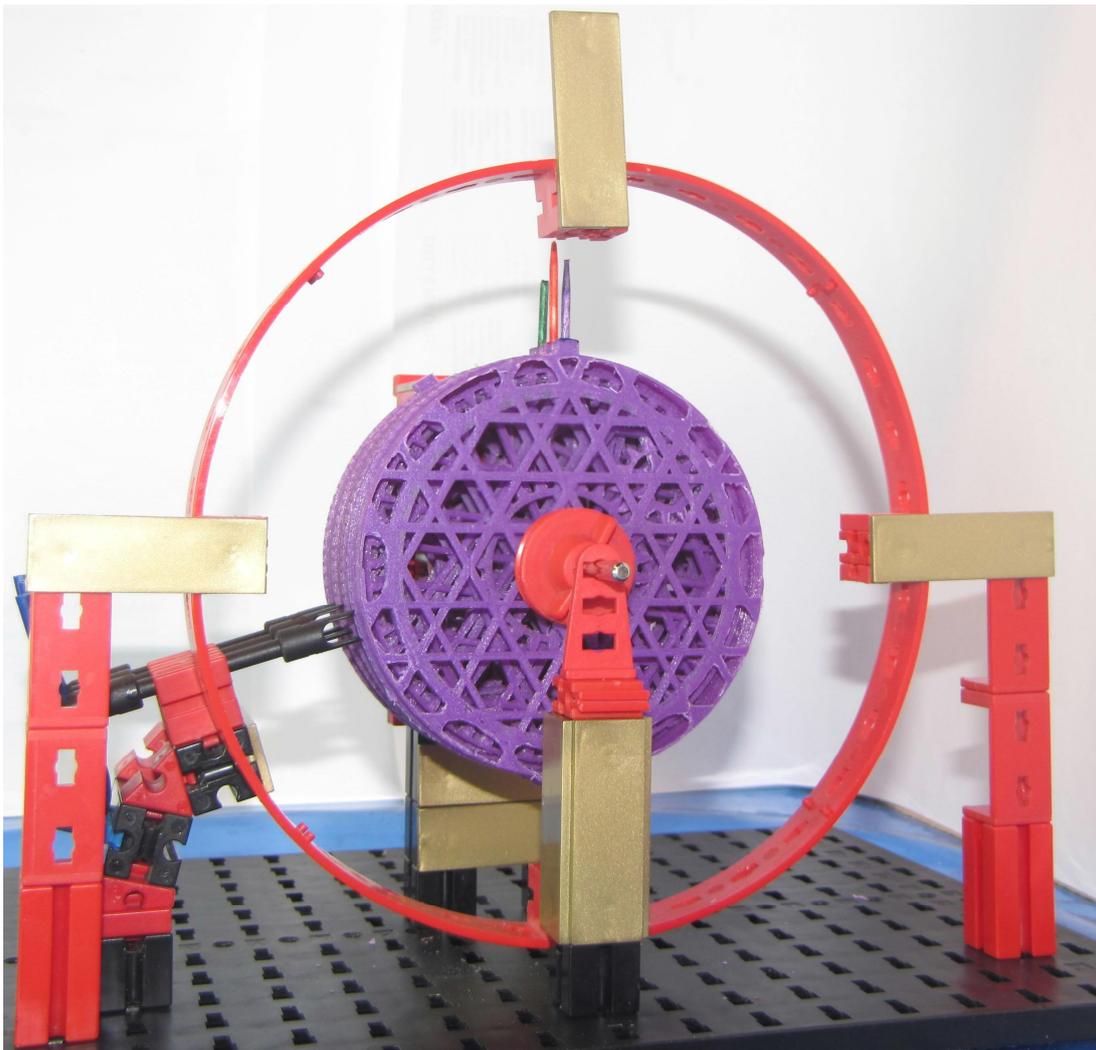


Abb. 10: Gesamtansicht – die goldenen Bauplatten stammen vom fischertechnik-Tag in Tumlingen

Tipps &amp; Tricks

## Eingänge vermehren

Kurt Mexner

Die acht Eingänge des TXT sind schnell belegt. Wenn Endlagenschalter, Steuerungsschalter, Fototransistor oder weitere Module angeschlossen werden müssen, gelangt der Controller schnell an seine Grenzen. Ein weiterer TXT oder eine Erweiterung über den EXT-Anschluss geht ins Geld. Hier kommt eine einfache und kostengünstige Eigenbaulösung, wie ein Eingang als Mehrfach-Eingang benutzt werden kann.

Die Lösung liegt in den Eingängen selbst verborgen. Es sind Analog-Eingänge. Das bedeutet: die Eingänge erkennen nicht nur „EIN“ und „AUS“, sondern auch Zwischenwerte. Je nach Beschaltung werden diese in Ohm oder Volt ausgegeben. Wir nehmen nun vier Taster in Reihenschaltung. Drei davon sind mit einem Widerstand von 1 k $\Omega$  (= 1000  $\Omega$ ) miteinander verbunden. Das folgende Schaltbild zeigt diese einfache Schaltung (Abb. 1).

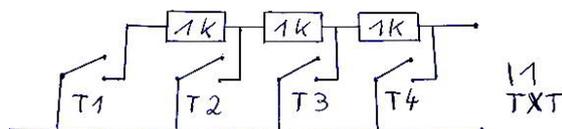


Abb. 1: Die Schaltung

Sie kann auch von Ungeübten auf einem Steckboard aufgebaut werden. Mit geringen Lötkenntnissen ist sie auch auf einer Streifenrasterplatine schnell umgesetzt. Widerstände, Steckboard oder Streifenrasterplatine sind bei den Bauteilehändlern (Pollin, Reichelt, Conrad, Völkner usw.) leicht erhältlich.

Die Widerstände sind hintereinandergeschaltet. Wenn sie mit den Tastern T1 bis T4 nacheinander angesteuert werden, liefern sie theoretisch einen analogen Wert von 3000  $\Omega$ , 2000  $\Omega$ , 1000  $\Omega$  und 0  $\Omega$ . Bei meiner Schaltung maß ich die Werte 3295  $\Omega$ , 2162  $\Omega$ , 1320  $\Omega$  und 280  $\Omega$ .

Handelsübliche Widerstände haben Toleranzen von ca. 5-10 %. Hinzu kommt der Innenwiderstand des TXT. Das erklärt die Unterschiede.

### Widerstände messen

Der TXT hat ab Werk ein Widerstandsmessgerät eingebaut, das aber nur Werte bis 15 k $\Omega$  misst. Wenn der Interfacetest angeklickt wird, öffnet sich ein Fenster, bei dem die Eingänge und Ausgänge angezeigt werden. Wenn man bei einem Eingang, z. B. I1, auf das Auswahlmenü klickt und dann *Analog 5 k $\Omega$  NTC* wählt, kann man am Ausgang I1 Widerstände messen. Man muss aber 100  $\Omega$  abziehen, da der TXT bereits intern mit einem 100  $\Omega$ -Schutzwiderstand abgesichert ist.

### Programm in ROBO Pro

Die Auswertung der Werte übernimmt ROBO Pro für uns. Das Programm steht unter [1] zum Download bereit.

Das Verzweigungselement I1 ist im Menü *Sensortyp* auf *NTC-Widerstand* eingestellt. I1 wird mehrmals abgefragt und entsprechend den Werten verzweigt.

Die Schaltstufe wurde in das Unterprogramm *Schalten* verlegt. Das dortige Befehlselement erzeugt die Befehle „EIN“ und „AUS“ jeweils im Wechsel. Bei Rückkehr aus dem Unterprogramm wird der Befehl

umgesetzt, entweder in einem Anzeigeelement auf dem Monitor oder über die Ansteuerung einer fischertechnik-LED an O1-O4 in einem fischertechnik-Modell.

## Bedienung

Bei kurzen Druck auf einen der Taster (diese sind als Schließer geschaltet) leuchtet das jeweilige Anzeigeelement und/oder eine der LEDs (O1-O4) auf. Wenn man zu lange drückt, blinkt die entsprechende Anzeige/LED. Will man ausschalten, einfach nochmals drücken.

Je nach Wert der benutzten Widerstände sind Anpassungen bei den Werten der Verzweigung I1 zu machen. Die „größer“- bzw. „kleiner“-Werte müssen evtl. geändert werden.

## Ausbaumöglichkeiten

Es ist möglich, sechs oder acht Widerstände mit entsprechenden Schaltern zu kombinieren, um so aus einem Eingang sechs bis acht Eingänge zu erzeugen.

## Beschränkungen

Es gibt eine kleine Zeitverzögerung, die durch das Element *Wartezeit* verursacht wird. Ich habe dies auf 0,5 s eingestellt. In ROBO Pro ist diese Stelle durch *Time* gekennzeichnet. Wenn man diese Zeit verkürzt, gibt es manchmal Probleme mit Lampe 1/O1. Sie schaltet, wenn T2 oder T3 gedrückt wird. Ich habe lange gesucht, aber keine Erklärung für dieses Problem gefunden. Bei der Einstellung von 0,5 s gibt es aber keine Probleme.

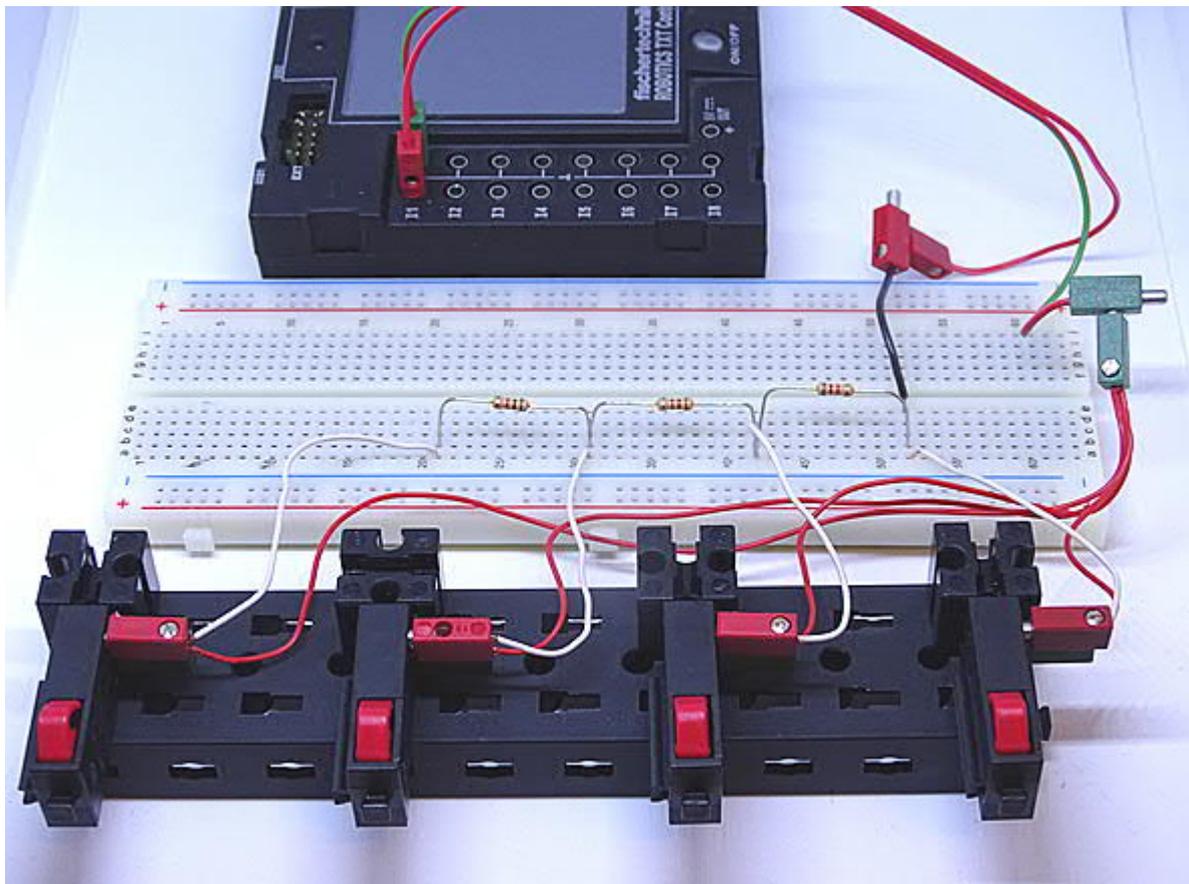


Abb. 2: Aufbau auf einem Steckboard

Das gleichzeitige Drücken von zwei oder mehr Tastern ist im Programm nicht vorgesehen. Das Programm eignet sich also nur für einfache EIN/AUS-Aufgaben. Zeitkritische oder komplexere Aufgaben scheiden aus. Aber für viele einfache Schaltaufgaben ist es eine gute und kostengünstige Lösung, falls weitere Eingänge benötigt werden.

## Quellen

- [1] Kurt Mexner: [FT 4fach Eingang.rpp](#). Im Downloadbereich der ft Community, 2020.

Computing

## fischertechnik-Roboter mit Arduino (Teil 2): 2,4-GHz-Fernbedienung mit dem PS2-Gamepad

Arnoud van Delden

*In dem Buch „fischertechnik-Roboter mit Arduino“ [1] werden fünf Roboter vorgestellt und mit zahlreichen Funktionen und Arduino-Steuerungen ausgestattet. Einige Leser haben in den vergangenen Monaten weitere Modellideen beigetragen – in loser Folge stellen sie in dieser Serie auch einzelne ihrer Modellvarianten vor. Diesmal geht es um die Fernsteuerung der Arduino-Roboter mit einem Wireless Gamepad für die Sony PlayStation 2.*

### Eine kabellose Steuerkonsole

Die erste Generation von Spielekonsolen wie die Sony PlayStation 2 verfügte noch nicht über eine kabellose Steuerung. Heutzutage werden kabellose Spielekonsolen in verschiedenen Robotergeschäften und Online-Geschäften verkauft, von denen das Empfängermodul an der Vorderseite der PlayStation (klassische Version 2) angeschlossen werden kann. Für rund 20 € erhält man eine Funk-Fernsteuerung mit zwei Joysticks und fünfzehn zusätzlichen Funktionstasten. Wer ein paar Wochen warten möchte, kann den Controller auch für weniger als die Hälfte dieses Preises direkt aus China beziehen.



Abb. 1: Sony PlayStation 2 (PS2) Controller

Das Gamepad überträgt die Daten im 2,4-GHz-HF-Frequenzband, und die gesende-

ten Signale können digital vom mitgelieferten Empfängermodul (gemäß Spezifikation) aus einer Entfernung von bis zu 10 Metern empfangen werden. Dies macht es natürlich attraktiv zu untersuchen, ob ein solches Gamepad als billige, hochfunktionale Fernbedienung für fischertechnik-Modelle wie die Arduino-Roboter aus dem Buch dienen könnte!

Das Besondere ist, dass die Schnittstelle bidirektional ist. Sowohl die Schnittstelle als auch das Gamepad sind daher sowohl Sender als auch Empfänger. Es gibt also auch einen Datenfluss vom Modell zurück zum Gamepad, um zwei kleine Motoren im anzutreiben, die durch Vibration eine Rückmeldung geben können. Sehr interessant, um beispielsweise die Greif- oder Wendegrenzen von Robotern, mögliche Hindernisse auf dem Weg eines lenkbaren Fahrzeugs oder, wie ich es jetzt für den „Flitzer“ als Experiment genutzt habe, konkrete Rückmeldungen beim Bremsen anzuzeigen.

Der Gamepad fühlt sich im Vergleich mit den eigentlichen Sony PlayStation-Controllern eher wie ein Spielzeug an. Aber das ist zu diesem Preis natürlich auch zu erwarten. Mein Modell enthielt kein Handbuch, ich konnte aber ein (wahrscheinlich aus dem Chinesischen übersetztes) Handbuch online



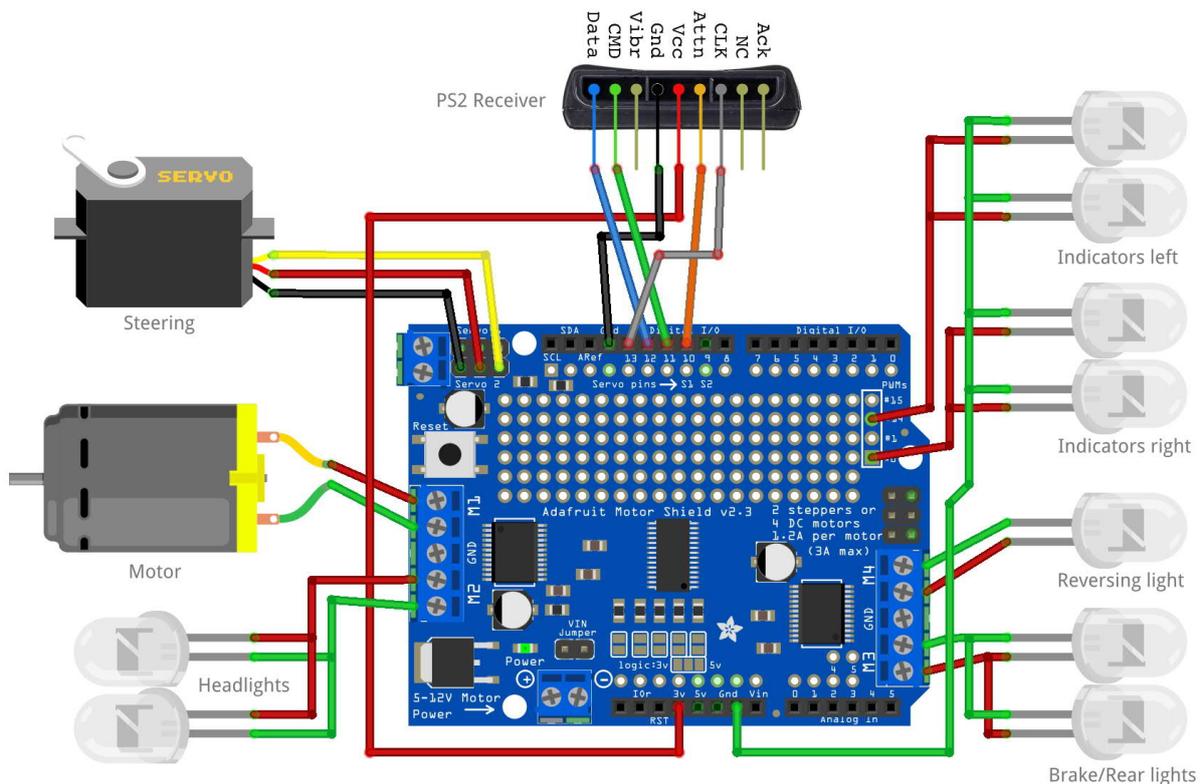


Abb. 4: Anschluss des PS2-Empfängermoduls an das Adafruit Motor Shield

## Verbindung mit dem Arduino

Zur Steuerung der fischertechnik-Roboter aus dem Buch müssen wir das Empfängermodul mit dem Motor Shield des Arduino verbinden. Wir schließen es an die digitalen Pins D10 bis D13 an (Abb. 4). Eine Versorgungsspannung von 5V verursacht keine Probleme, aber da eine klassische PlayStation 2 nur 3,4V verwendet, bin ich auf der sicheren Seite geblieben und habe den 3,3V-Ausgang des Motor Shield verwendet.

Für diejenigen, die den „Flitzer“ bereits gebaut haben, sind die Verbindungen nichts Neues. Das einzige, was ich geändert habe, ist, dass ich die roten Rücklichter von einem Motorausgang aus steuere, um verschiedene Lichtintensitäten zu erzielen. Dies ermöglicht es, während der Fahrt normale Rücklichter zu verwenden, diese jedoch beim Bremsen (Verzögerung) heller leuchten zu lassen. Der von mir erstellte Arduino-Sketch unterstützt diese Funktion jetzt [6].

## Rückkopplung durch Vibration

Ich musste mein Gamepad aufschrauben, da der Ein-/Ausrichter manchmal nicht richtig funktionierte. Nachdem die Anschlüsse um das Batteriefach und den Schalter sauber verlötet waren, funktionierte alles einwandfrei. Da ich das Gamepad geöffnet hatte, konnte ich einen Blick ins Innere werfen (Abb. 5).

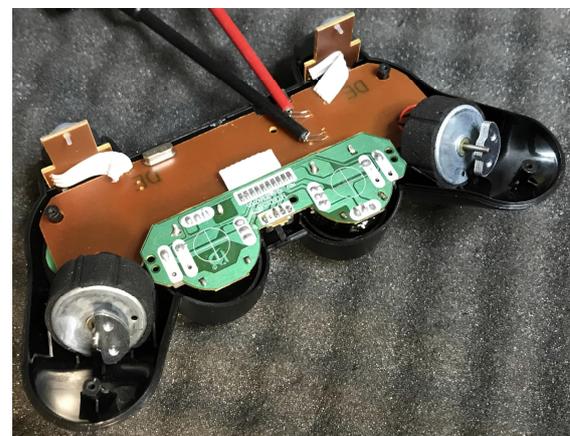


Abb. 5: Innenansicht des PS2-Gamepad

In den Griffen sind die beiden Vibrationsmotoren deutlich zu erkennen. Der kleine Motor rechts kann ein- oder ausgeschaltet werden; der Motor links mit etwas größerem Gegengewicht ist stufenlos einstellbar. Für einen zusätzlichen Effekt können auch beide Motoren gestartet werden. Mit einem Test-Sketch konnte ich deren Steuerung und Funktionsweise untersuchen [5].

Die Drucktasten können je nach Modus digital oder analog ausgelesen werden. Sie können einen von der ausgeübten Druckkraft abhängigen analogen Wert zurückgeben, auch wenn der nicht sehr genau ist. Die Steuerung verfügt laut Beschreibung über zwei Modi: Es ist auch möglich, nur Binärwerte auszulesen. Vielleicht ist dies ein Überbleibsel aus den ‚frühen Tagen‘ der PlayStation 2, als die Spiele weniger fortgeschritten waren. Bei Verwendung der oben genannten PS2-Bibliothek ist standardmäßig der A/D-Modus aktiviert, in dem ein Wert zwischen 0 und 255 die Lage der Joysticks zurückliefert und die zwölf Funktionstasten auf Wunsch druckempfindlich sind.

Die Verbindung zwischen dem Gamepad und dem Empfänger ließ sich schnell und zuverlässig herstellen. Es macht keinen Unterschied, ob das Gamepad zuerst eingeschaltet wird oder ob man den „Flitzer“ zuerst startet.

Nach einiger Zeit der Inaktivität schaltet sich das Gamepad selbstständig aus. Nach dem gefundenen Handbuch sollte dies nach einer Minute geschehen; mein Controller wurde jedoch erst nach fünf Minuten ausgeschaltet. Danach funktionierte das automatische Wiederverbinden des Controllers mit der Benutzeroberfläche durch Drücken der „Start“-Taste auf dem Gamepad nicht.

Dies scheint mir in der Praxis jedoch kein großes Problem zu sein, da diese Zeitüberschreitung beim normalen Betrieb eines Modells nicht so bald erreicht wird.

Sollte dies erforderlich sein, kann man jedoch eine Lösung entwickeln, bei der die Empfängerschnittstelle (die die erneute Verbindung initiiert) von der Software aus neu gestartet wird, wenn keine Kommunikation erfolgt. Die aktuelle Arduino-Bibliothek bietet jedoch kein solches Zurücksetzen oder erneutes Verbinden via Software.

Eine einfache Hardware-Umgehung könnte eine Unterbrechung der 3,3V-Versorgungsspannung des Empfängers sein. Zu diesem Zweck könnte erwogen werden, den Empfänger über einen der freien Motorausgänge des Motor Shield zu versorgen. Es erscheint dann sinnvoll, die Vcc-Eingangsspannung (z. B. mit einer Zener-Diode) auf 3,3V zu begrenzen. Aber auch weil es für mich kein Problem war habe ich nicht weiter damit experimentiert.

Das PS2-Empfängermodul wird mit selbstklebendem Klettverschluss auf den Batteriehalter des „Flitzer“ befestigt (Abb. 6).

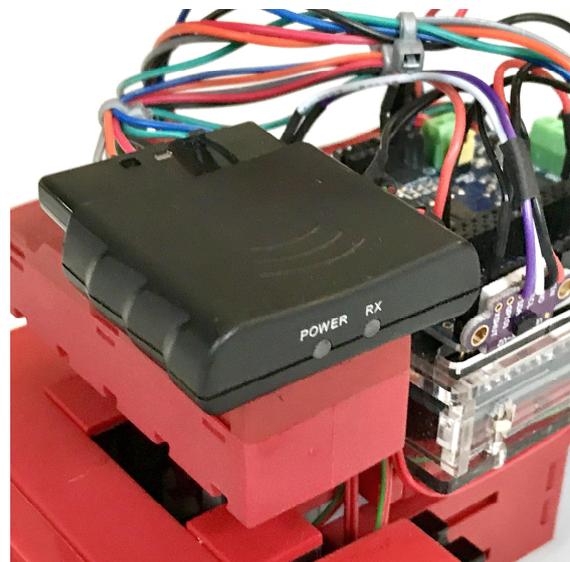


Abb. 6: Montage des PS2-Empfängers

## Die Software

Glücklicherweise haben schon einige Leute mit einem PS2-Gamepad experimentiert [2, 3] und es gibt auch bereits eine Bibliothek für den Arduino [4]. Sie ist nicht in der Bibliotheksverwaltung der Arduino IDE enthalten, aber das Hinzufügen ist einfach.

In dem auf der [Webseite der ft:pedia herunterladbaren Sketch](#) habe ich die oben erwähnte Erweiterung für hellere Bremslichter bereits vorgenommen [6].

Ich habe auch mit der Steuerung der Vibrationsmotoren experimentiert, um der Konsole eine taktile Rückmeldung an die Fahrzeugsteuerung zu übermitteln, während der „Flitzer“ verlangsamt (gebremst wird). Dieses Verhalten kann mit der Taste mit dem rosa Quadrat ein- und ausgeschaltet werden. Die Beleuchtung (vorne und hinten) wird mit der Taste mit dem roten Kreis eingeschaltet, und die Taste mit dem grünen Dreieck schaltet den Warnblinker ein. Die Geschwindigkeit und Fahrtrichtung des „Flitzers“ wird mit dem linken Joystick gesteuert. Beim Verlangsamen (Bremsen)

leuchten die Bremslichter eine Sekunde lang. Der rechte Joystick sorgt für die Lenkung mit dem Servo.

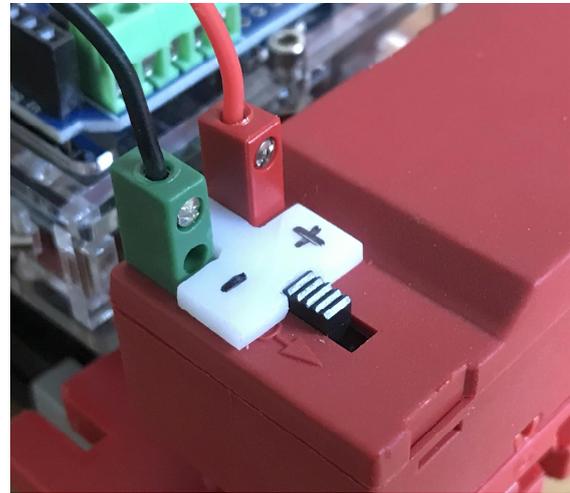


Abb. 7: Polaritätsschutz am Batteriehalter



Abb. 8: Flitzer mit PS2-Fernsteuerung

Wie bereits erwähnt kann der PS2-Gamepad mit zwei wieder aufladbaren AAA-Akkus betrieben werden. Ich verwende einen 9V-Blockakku, um mein Modell mit Strom zu versorgen.

Diese Lösung ist klein und leicht. Da die Polarität beim Einschalten wichtig ist, habe ich eine kleine Anpassung am Batteriehalter ([135719](#)) vorgenommen: Die Stromversorgung kann wegen eines kleinen Plexiglas-Scheibchens nur in einer Richtung eingeschaltet werden (Abb. 7).

## Fazit

Ich habe ein Video von meinem „Flitzer“ mit diesem kabellosen Gamepad veröffentlicht [6]. Ich kann jedem, der eine zuverlässige und kostengünstige Fernbedienung mit vielen Erweiterungsoptionen sucht, ein Experiment mit diesem PS2-Gamepad empfehlen. Mit zwei Joysticks und nicht weniger als zwölf Funktionstasten für zuweisbare Funktionen bietet es viele Steuerungsmöglichkeiten für jedes Arduino-Modell. Das Vibrationsfeedback im Gamepad selbst ist ein schöner Bonus, der für einige Modelle tatsächlich funktional genutzt werden kann.

## Quellen

Sucht einfach im Internet nach „Wireless PS2 Controller“, um verschiedene aktuelle Angebote für diesen „2.4G Wireless Dual Shock PS2 Game Controller“ zu finden. Das Gehäuse des Gamepad ist in verschiedenen Farben erhältlich. Es wird wahrscheinlich wenig elektronische Unterschiede geben.

- [1] Dirk Fox, Thomas Püttmann: [fischertechnik-Roboter mit Arduino](#). dpunkt-Verlag, 2020.
- [2] Scott Driscoll: [Interfacing a PS2 \(PlayStation 2\) Controller](#). Curious Inventor.
- [3] Bill Porter: [PlayStation 2 Controller Arduino Library v1.0](#). billporter.info, 05.06.2010.
- [4] Bill Porter: [PS2-Controller Library für den Arduino](#). github.com.
- [5] Arnoud van Delden: [Arduino-Sketches zum Testen der Vibrationsmotoren und Hauptprogramm für den „Flitzer“](#).
- [6] Arnoud van Delden: [Wireless PS2 Controller with Arduino](#). YouTube, whizzbizz.com

Computing

## Bluetooth-Modul für den ftDuino

Till Harbaum

*Bluetooth hat inzwischen überall in unserem Leben Einzug gehalten – PCs, Tablets und Smartphones verbinden wir darüber mit Kopfhörern, Druckern und Musikanlagen. In der fischertechnik-Welt führt Bluetooth hingegen noch ein Randdasein. Sehr zu Unrecht: Denn in Verbindung mit einer Smartphone-App lassen sich elegante, flexible und komfortable Fernsteuerungen realisieren. Der ftDuino macht es vor.*

### Hintergrund

Dass Bluetooth ein „heißes“ Thema auch für den ftDuino ist, hat die Community in einem entsprechenden Thread im Forum deutlich gemacht [1]. Daraus hat sich schnell ein weiterer Thread ergeben, in dem ein paar mögliche Lösungen diskutiert und auch umgesetzt wurden. Diese Lösungen erfüllten ihren Zweck, zogen aber aufgrund ihres etwas unkonventionellen Aufbaus schnell Spott auf sich (Abb. 1) [2].

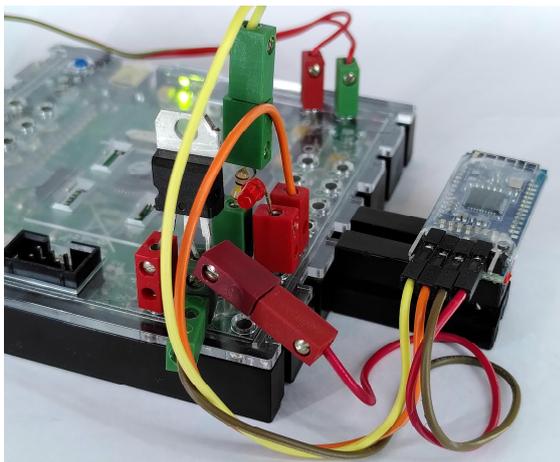


Abb. 1: Ein erster, als „Stromturm“ belächelter fliegender Aufbau

Eines bewiesen diese Lösungen aber auf jeden Fall: Es gab und gibt Interesse an einer Bluetooth-Lösung für den ftDuino und technisch stellt die Anbindung kein Problem dar. Parallel weckte der Thread zum Buch „fischertechnik-Roboter mit

Arduino“ zusätzliches Interesse an einer Bluetooth-Lösung im fischertechnik- und Arduino-Umfeld [3, 4]. Und schließlich war Bluetooth in dieser Form auch bereits Thema eines Artikels in der vorherigen Ausgabe der ft:pedia [5].

Damit war das Ziel gesteckt: Eine Bluetooth-Lösung für den ftDuino musste her. Mit vollem Funktionsumfang, handlich klein, robust, preisgünstig und als Fertiggerät erhältlich, damit auch Anfänger es sofort nutzen können.

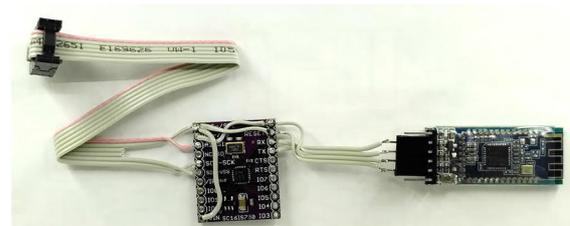


Abb. 2: HM-10-Bluetooth-Modul an SC16IS750-I<sup>2</sup>C-UART-Bridge

In den Forums-Threads hatte sich bereits herauskristallisiert, dass eine I<sup>2</sup>C-Lösung eigentlich am elegantesten wäre, da sie sich nahtlos in das Konzept des ftDuino integriert. Die nötigen Bauteile waren aber zusammen recht unhandlich und auch nicht ganz billig. Im Wesentlichen bestand die bereits im Thread veröffentlichte Lösung aus einem handelsüblichen HM-10-Bluetooth-Modul und einem I<sup>2</sup>C-UART-Brückenbaustein SC16IS750 (Abb. 2).

Der Brückenbaustein übersetzt zwischen dem I<sup>2</sup>C-Anschluss des ftDuino und dem mit der klassischen seriellen PC-Schnittstelle nah verwandtem UART-Anschluss des HM-10. Eine dritte Funktion erledigen ein paar Bauteile auf dem HM-10 und der SC16IS750 eher nebenbei: Die Umsetzung diverser Signale zwischen den 5 Volt des ftDuino und den 3,3 Volt, die das HM-10-Bluetooth-Modul erwartet.

Eine elegante Lösung der drei Aufgaben „Bluetooth-Modul“, „I<sup>2</sup>C-Anbindung“ und „Spannungsumsetzung“ war also das Ziel.

Die erste Aufgabe „Bluetooth-Modul“ war schon nicht ganz unproblematisch. Was man unter der Bezeichnung HM-10 kauft ist nämlich in der Regel mitnichten ein HM-10 von Huamao [6]. Stattdessen gibt es eine Unzahl kompatibler und inkompatibler Plagiate, Nachbauten und Nachfolgermodelle in unterschiedlichen Bauformen und mit unterschiedlichen elektrischen und funktionalen Eigenschaften (Abb. 3). Letztlich fiel die Wahl auch nach Rücksprache mit dem Originalhersteller auf das HM-11, die mechanisch kleinere Variante des verbreiteten HM-10. Das HM-11 eignet sich wie das HM-10 vor allem zur Verbindung mit Smartdevices, allen voran auf Apple-IOS- und Android-Basis. Zur Verbindung mit dem (Windows-)PC eignet sich das HM-11 durch seine Bluetooth-LE-Implementierung eher weniger.



Abb. 3: Lauter grundverschiedene Bluetooth-Module

Einen Spannungsregler für die 3,3 Volt hätten wir uns gern gespart; die ersten Versionen der Nachfolgemodule HM-17 und HM-19 wurden in älteren Datenblättern auch noch als 5 Volt-verträglich gelistet. In der Praxis stellte sich das als falsch heraus; in aktuellen Datenblättern des HM-17 und HM-19 ist das korrigiert. An einem Spannungsregler führte also kein Weg vorbei.

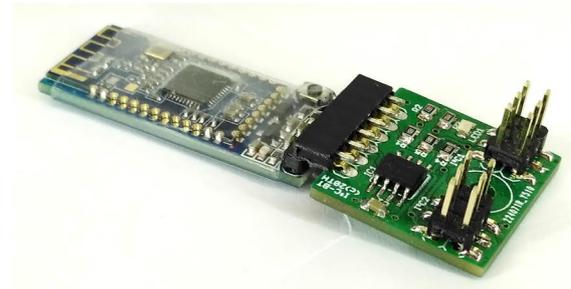


Abb. 4: Zwischenschritt: HM-10 an eigenem Adapter

Schließlich stellt die I<sup>2</sup>C-Anbindung die letzte Hürde dar. Der SC16IS750 ist leider recht groß und teuer, so dass sich die Frage stellte, ob man hier nicht mit einem kleinen passend programmierten Mikrocontroller platzsparender und günstiger zum Ziel kommt. Auch das gestaltete sich etwas schwieriger als gedacht, da die wenigsten Controller in dieser Größenordnung zuverlässig alle benötigten UART-Datenraten zwischen 1200 und 230400 bit/s erzeugen können und dabei gleichzeitig eine I<sup>2</sup>C-Schnittstelle bedienen. Aber auch hier wurden wir schließlich fündig, so dass alle Komponenten für einen kleinen und günstigen Adapter beisammen waren.

Außerdem erlaubt der Einsatz eines eigens programmierten Controllers als Brückenbaustein auch kleine Hilfsfunktionen. Wer etwas Erfahrung mit dem HM-10 hat, vermisst beim ftDuino-Adapter ggf. den kleinen Drucktaster, mit dem ein völlig fehlkonfiguriertes Modul wieder in den Ausgangszustand versetzt werden kann. Diese Funktion übernimmt im ftDuino-Bluetooth-Adapter ebenfalls der Mikrocontroller, sodass der mitgelieferte Sketch („blue-

tooth\_config“) jedes noch so sperrig verstellte Modul ganz automatisch wieder zum Leben erweckt.



Abb. 5: Die Platine des ftduino-Bluetooth-Adapters

Das Ergebnis ist eine kleine Platine, die sich auf einem 30x45-Fußabdruck entweder auf eine fischertechnik-Bauplatte kleben oder gleich mit einem passenden Gehäuse versehen lässt.

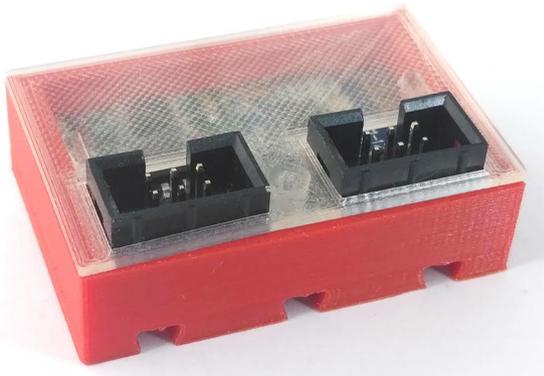


Abb. 6: ftduino-Bluetooth-Adapter in 3D-gedrucktem Gehäuse

Die aktuelle Version der ftduino-Integration in die Arduino-IDE bringt bereits einige passende Beispiele mit. Mit dabei ist auch ein ArduinoBlue-Beispiel, das dem ftduino die in der letzten ft:pedia beschriebene App zugänglich macht [5, 7]. Weiterhin sind Beispielsketches für 1Sheeld [8] und ftduinoBlue [9] dabei.

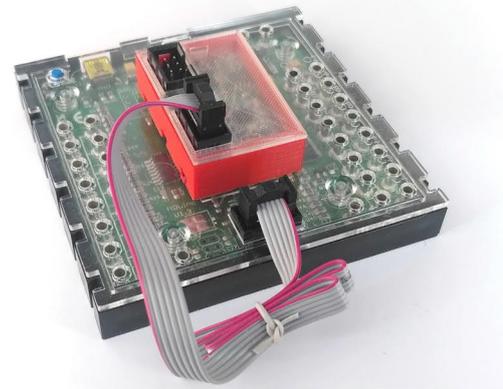


Abb. 7: ftduino-Bluetooth-Adapter mit zwei Federnocken auf ftduino befestigt

ftduinoBlue wurde eigens für den ftduino-Bluetooth-Adapter entwickelt und versucht sich an einer etwas anderen Variante der App-Steuerung. Hier gibt nicht die App die Steuerelemente vor, sondern der Arduino-Sketch [9]. Auf diese Weise lassen sich komfortable App-Bedienoberflächen für fischertechnik-Modelle ganz ohne App-Programmierung erzeugen. Natürlich ist ftduinoBlue nicht auf den ftduino und seinen Adapter beschränkt. Es lässt sich auch mit jedem Arduino verwenden und ist im Android-App-Store zu finden.



Abb. 8: Der 3-Achs-Roboter aus dem TXT-Automation-Robots mit ftduino und Adapter

Natürlich haben wir auch vom ftduino-Bluetooth-Adapter wieder eine Kleinserie aufgelegt. Der fertig aufgebaute Adapter ist daher beim Erscheinen dieser ft:pedia-Ausgabe exklusiv bei MINtronics erhältlich [10]. Wer sich genauerer über den ftduino-Bluetooth-Adapter informieren will, findet die Anleitung zu unserem kleinen Gerät unter [11].

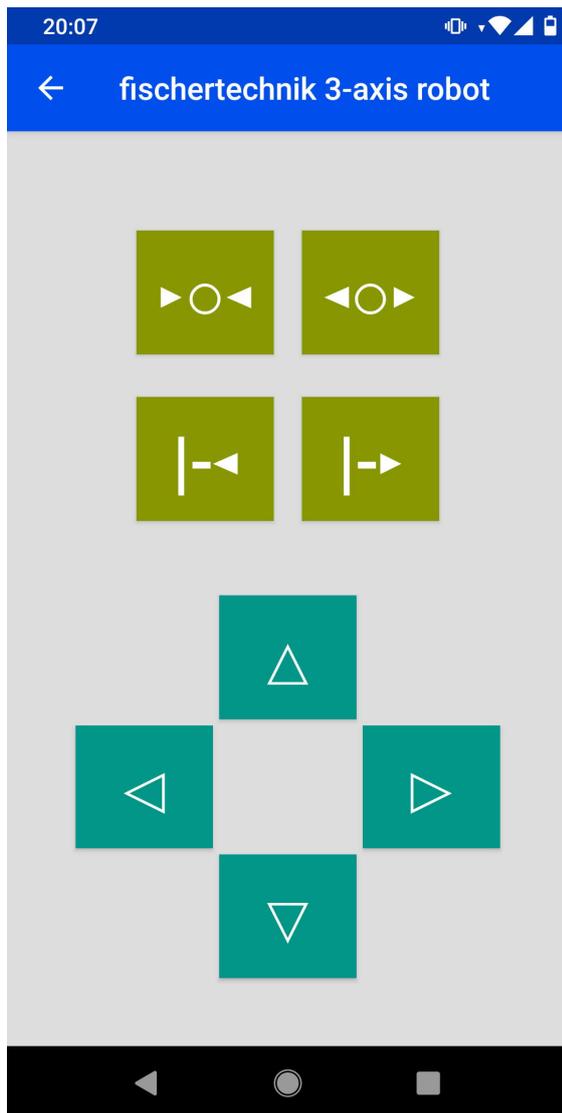


Abb. 9: ftDuinoBlue-Oberfläche zum 3-Achs-Roboter aus TXT-Automation-Robots

## Referenzen

- [1] [Thread „HC-05 an ftduino“](#), ftcommunity Forum
- [2] [Thread „Bluetooth am ftduino“](#), ftcommunity-Forum
- [3] Dirk Fox, Thomas Püttmann: [fischertechnik-Roboter mit Arduino](#). dpunkt-Verlag, 2020.
- [4] [Thread „Modelle aus Baukasten und Buch Fischertechnik-Roboter mit Arduino“](#), ftcommunity-Forum
- [5] Dirk Fox: [fischertechnik-Roboter mit Arduino \(Teil 1\): Smartphone-Steuerung über BLE](#), ft:pedia 3/2020, S. 93-100.
- [6] Jinan Huamao Technology Co.: [BLE Modules](#).
- [7] Anurag Purwar, Jaehyeon An: [ArduinoBlue](#). Stony Brook University.
- [8] Google Play Store: [ISheeld](#).
- [9] Till Harbaum: [ftduinoBlue – Control Your Arduino from Android](#).
- [10] Peter Habermehl: [MINtronics](#).
- [11] Till Harbaum: [Anleitung zum ftduino-Bluetooth-Adapter](#), 16.12.2020.

Elektronik

## ft-Soccer – IR-Sensor für den RoboCup-Junior-Wettbewerb „Soccer“

Björn Gundermann, Christian Lauff, Stefan Fuss

Seit dem Jahr 2000 gibt es bei dem weltweit ausgetragenen Robotik-Wettbewerb RoboCup einen Junior-Wettbewerb. Eine der Disziplinen ist „Soccer“ – ein Fußballwettkampf. Andreas Kempf hat 2015 darüber in der ft:pedia berichtet [1]. Für die Lokalisierung des Fußballs, der in hoher Frequenz IR-Signale aussendet, benötigt man einen Sensor, der die aktuelle Position und die Bewegungsrichtung des Balls feststellen kann. Das gelingt mit dem in diesem Beitrag vorgestellten Sensormodul „ft-Soccer“.

### Hintergrund

Vor einiger Zeit erhielt ich (Björn) eine Anfrage, ob ich nicht einen Sensor für den RoboCup-Junior-Wettbewerb ‚Soccer‘ entwickeln könnte. Einige Monate zuvor hatte ich schon einmal eine ähnliche Anfrage erhalten. Die Zeit passte besser und meine Neugierde war geweckt. Für die Prüfung erhielt ich die Beschreibung des vorhandenen Sensors und einen funktionierenden sowie einen defekten Sensor. Hierbei handelte es sich um einen Tris10 Findball, der jedoch nicht mehr vertrieben wird (Abb. 1).

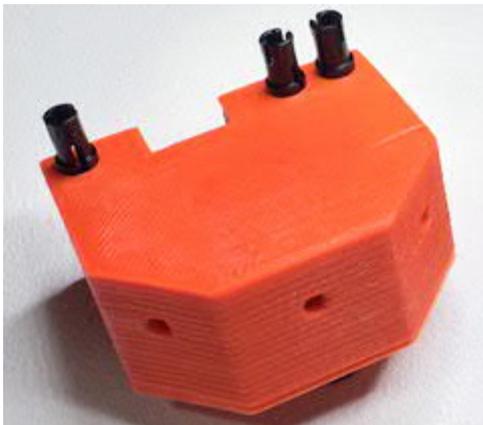


Abb. 1: Tris10 Findball

Während der Analyse war also zu klären, auf was der Sensor überhaupt reagieren soll.

### IR-Signal des Fußballs

Erster Punkt war also die Spezifikation des verwendeten Fußballs. In der RoboCup-Beschreibung steht [2]: *In der Lightweight League sendet der Ball Infrarot-Licht aus, sodass er mit den Infrarot-Sensoren des Roboters erkannt werden kann.* Die eigentlichen Soccer-Regeln spezifizieren den Ball dann wie folgt [3]:

*The ball emits infrared (IR) light of wavelengths in the range 920nm - 960nm, pulsed at a square-wave carrier frequency of 40 kHz. The 40 kHz carrier output of the ball shall be modulated with a trapezoidal (stepped) waveform of frequency 1.2 kHz. Each 833-microsecond cycle of the modulation waveform shall comprise 8 carrier pulses at full intensity, followed (in turn) by 4 carrier pulses at 1/4 of full intensity, four pulses at 1/16 of full intensity and four pulses at 1/64 of full intensity, followed by a space (i.e. zero intensity) of about 346 microseconds.*

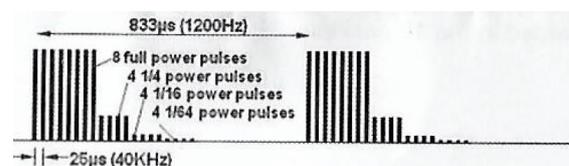


Abb. 2: IR-Signale des Fußballs

Bildlich sieht das dann so wie in Abb. 2 aus.

### Der IR-Sensor

Nachdem die Art des Sendesignals bekannt war, galt es als Nächstes dazu passende Sensoren zu finden. Die Soccer-Regeln erwähnten TSOP-Sensoren, daher fiel meine Wahl auf einen TSP31240. Dieser verfügt über drei Pins. Zwei sind die Spannungsversorgung und ein Pin ist der Ausgang. Im nächsten Schritt galt es herauszufinden, wie sich der TSP31240 verhält. Was passiert bei Empfang des Signals und was bei den verschiedenen Empfangsstärken?

Abb. 3 zeigt einen Auszug aus der Spezifikation des TSOP31240 [2].

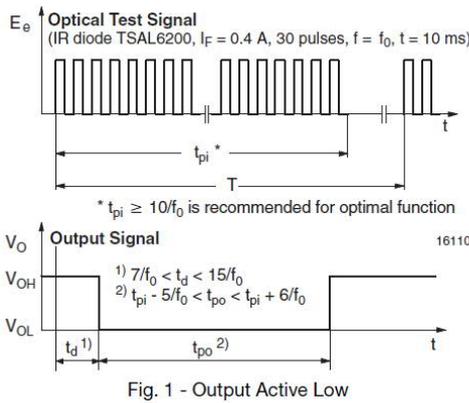


Fig. 1 - Output Active Low

Abb. 3: Verhalten des Sensors TSOP31240 [2]

Jetzt galt es noch zu prüfen, wie sich die Low-Phase in Verbindung mit dem Fußball verhält. Ein erstes Bild vom Oszilloskop zeigte dann Folgendes:

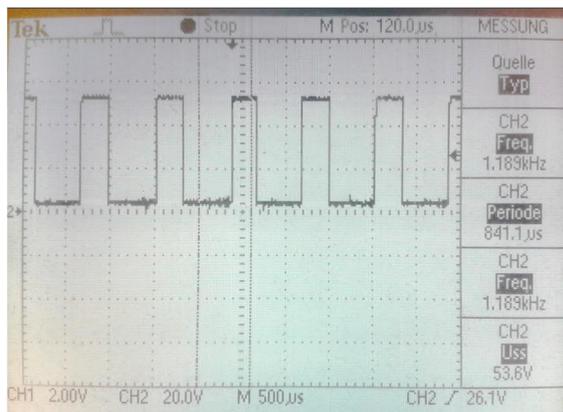


Abb. 4: Messung mit dem Oszilloskop

Der TSOP hält den Ausgangspin also, wie in der Spezifikation beschrieben, während der Pulsausgabe auf Low. Erkennt der TSOP kein Signal, dann gibt er High aus. Je kürzer die Low-Phase, desto schlechter ist also der Empfang des TSOP. Auf diese Art kann eine Auswertung über die Empfangsqualität erfolgen.

Darauf folgten weitere Untersuchungen mit verschiedener Empfangsqualität. Für diese Untersuchungen wechselte ich zu einem anderen Oszilloskop. Abb. 5 zeigt eine Messung bei guter Empfangsqualität.

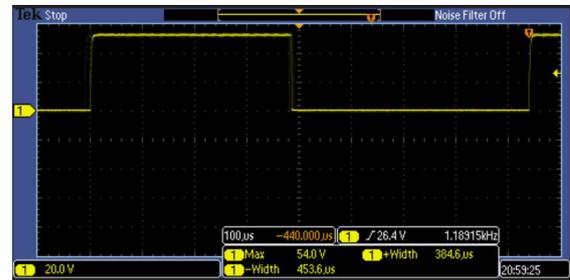


Abb. 5: Messung bei guter Empfangsqualität

Diese ließ sich einfach durch Abdecken der Sensoren mit den Händen verringern. Abb. 6 und Abb. 7 zeigen das Signalverhalten bei immer schlechterem Empfang.



Abb. 6: Schlechterer Empfang

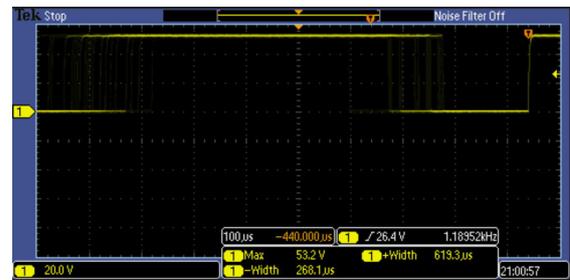


Abb. 7: Noch schlechterer Empfang

Wenn der Ball direkt neben dem Sensor liegt, also beste Empfangsverhältnisse herrschen, dann messen wir High  $320\mu\text{s}$  und Low  $520\mu\text{s}$ . Jetzt legen wir die Hand auf den Sensor, um den Empfang zu verschlechtern. Zu erwarten ist eine verkürzte Low-Phase. Dies konnte so reproduziert werden: Gesehen wurden High  $672\mu\text{s}$  und Low  $172\mu\text{s}$ . Das Minimum lag bei ungefähr  $164\mu\text{s}$ . Dann verliert der Sensor die Verbindung und wir sind bei einem konstanten High-Signal.

## Die Schaltung

Nachdem das Verhalten des Sensors geklärt war ging es zusammen mit Stefan an einen Schaltplan. Der ft-Soccer verfügt über fünf IR-Sensoren, um die Richtung des Balles besser identifizieren zu können (Abb. 8). Dies folgt dem Design des Tris10.

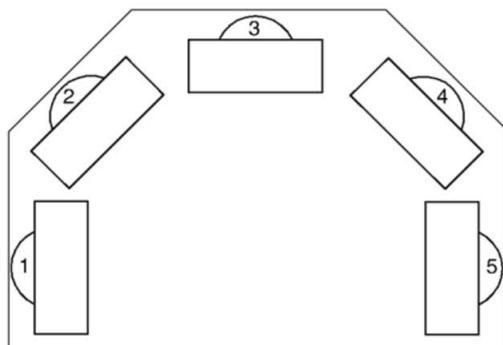


Abb. 8: Aufbau des ft-Soccer

Um kompatibel zu bleiben wurde die gleiche Registerbelegung verwendet. Genutzt wurde ein Arduino ATMEGA328P-AU. Als externe Anschlüsse wurden zwei Möglichkeiten vorgesehen: Einmal unsere Standard 6-polige Pfostenwanne mit der I<sup>2</sup>C-Anschlussbelegung des TX/ftDuino und zusätzlich eine RJ12-Buchse mit der Belegung für ein Lego-System. Auf der Platine befinden sich nur noch die notwendigen Kondensatoren und Widerstände.

Die Bestückung wurde auf unserem Bestückungsautomaten vorgenommen. Einen bestückten Nutzen und einen bestückten Sensor zeigen Abb. 9 und 10.



Abb. 9: Bestückung

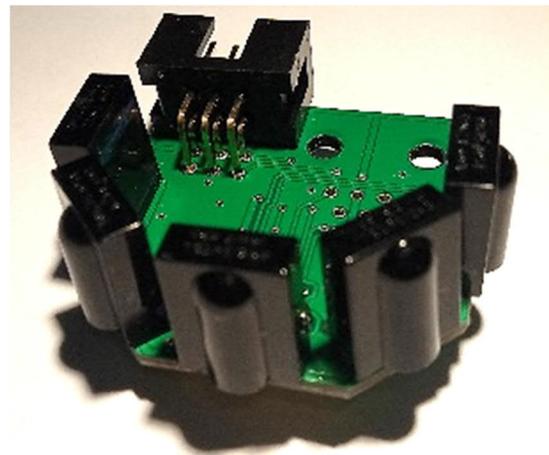


Abb. 10: Bestückter Sensor (ohne Gehäuse)

## Anschluss an den Controller

Die Stromversorgung erfolgt über den I<sup>2</sup>C-Bus. Sowohl die TSOP-Sensoren als auch der Arduino können mit 3,3V oder auch 5V betrieben werden. Die Versorgungsspannung der TSOP-Sensoren kann über den Arduino ein- und ausgeschaltet werden.

Die Buchsen an den Flachbandkabeln lassen sich auch ohne Spezialwerkzeug crimpen. An der Buchse ist Pin 1 mit einem Pfeil markiert. Für ein 1:1-Kabel wird hier die markierte Ader des Flachbandkabels angelegt. Ob das Kabel links oder rechts aus der Buchse geführt wird ist völlig egal, die Pinbelegung am Kabel bleibt gleich.

Danach wird entweder mit einer Wasserpumpenzange oder mit dem Schraubstock der Sicherungsclip auf die Buchse gedrückt. Dabei wird das Flachbandkabel automatisch kontaktiert. Auf jeden Fall ist dabei

darauf zu achten, dass dies gefühlvoll passiert. So kann leicht das benötigte Verbindungskabel zum Sensor erstellt werden.

Der TX/ftDuino ist – wie bereits erwähnt – direkt mit einer 1:1 Belegung anschließbar. Für den TXT bietet sich die Nutzung des ftExtender an [5].

## Ansteuerung

Um kompatibel mit dem Tris10 zu sein wurde dessen Registerbelegung übernommen. Die I<sup>2</sup>C-Adresse wurde jedoch geändert, da der Tris10 mit der Adresse 0 eine reservierte Adresse belegt hat. Der ft-Soccer reagiert auf die I<sup>2</sup>C-Adresse 0x21.

Die Registerbelegung ist in zwei Gruppen unterteilt. Die erste Gruppe beschäftigt sich mit den wichtigen Informationen des Gerätes.

Adresse	Beschreibung	Länge	Standard Wert
0x00	Version der Firmware	4	V1.0
0x08	Herstellername	6	Gunder
0x10	Sensortyp	2	IR
0x42	Nummer des Sensors mit bestem Empfang	1	
0x43	Intensität bester Sensor	1	
0x44	Intensität Sensor 1	1	
0x45	Intensität Sensor 2	1	
0x46	Intensität Sensor 3	1	
0x47	Intensität Sensor 4	1	
0x48	Intensität Sensor 5	1	
0x53	Intensität bester Sensor in Prozent	1	
0x54	Intensität Sensor 1 in Prozent	1	
0x55	Intensität Sensor 2 in Prozent	1	
0x56	Intensität Sensor 3 in Prozent	1	
0x57	Intensität Sensor 4 in Prozent	1	
0x58	Intensität Sensor 5 in Prozent	1	
0x80	Neues Scanintervall	1	Definiert das Scanintervall.

Tab. 1: Registerbelegung (Teil 1)

Die zweite Gruppe liefert rein statische Informationen, um mit LEGO-NXT/EV3-Geräten kompatibel zu sein.

Adresse	Beschreibung	Länge	Standard Wert
0x01	Kompatibilität mit Lego	1	0x03
0x12	Kompatibilität mit Lego	1	0x01
0x13	Kompatibilität mit Lego	1	0x01
0x14	Kompatibilität mit Lego	6	10E-2m
0x40	Kompatibilität mit Lego	1	0x01
0x51	Kompatibilität mit Lego	1	0x01
0x52	Kompatibilität mit Lego	1	0x01

Tab. 2: Registerbelegung (Teil 2)

Die Registerbelegung bis 0x48 entspricht der des Tris10. Die nachfolgenden Register aus der ersten Gruppe wurden von mir ergänzt.

## Firmware

Mit den analysierten Daten galt es jetzt einen Algorithmus zu erstellen. Der verwendete Arduino hat nicht genügend Interrupts, um für jeden Sensor einen eigenen nutzen zu können. Mit Interrupts wäre aber auch keine gemeinsame Zeitbasis vorhanden gewesen. Die Auswertung erfolgt daher über einen Timer, der alle fünf Sensoren gleichzeitig abfragt. Die Sensoren sind alle an einem Port angeschlossen, sodass der einzelne Sensor durch Maskieren des kompletten Ports abgefragt werden kann. Die Auswertung innerhalb des Timer Interrupts muss auch sehr schnell erfolgen, da dieser alle 20µs aufgerufen wird. So kommen wir auf ein Intervall von 840µs, das geringfügig über der Telegrammlänge von 833µs liegt.

Die Sensoren werden nach jedem Auswertintervall für 5ms ausgeschaltet und dann wieder eingeschaltet. Nach Einschalten der Versorgungsspannung liefern die Sensoren jedoch direkt noch keine gültigen Werte. Daher wird hier eine Verzögerung von 800µs eingefügt. Ohne diese Verzögerung können keine stabilen Werte aus dem Register PIND gelesen werden. Die Sensoren sind alle am Port D angeschlossen.

Das Auswertintervall besteht jetzt aus der Summe der gelesenen Telegramme, deren Anzahl konfiguriert wurde. Wie bereits erwähnt gilt: je höher der Low Anteil, desto besser der Empfang. Die Periode unseres Telegramms hat 840µs, im optimalen Fall 320µs High und 520µs Low. Diese maximale Low Zeit gilt dann auch als 100% und ist die Basis für die Register mit der Sensorintensität in %.

Je größer das Auswertintervall desto genauer das Ergebnis bei statischer Betrachtung. Das Ergebnis selbst ist aber auch eher statisch und eine Bewegung des Balles wird somit nicht sehr gut erkannt. Hier gilt es jetzt den Mittelweg zwischen stabiler Auswertung pro Sensor und der gewünschten Agilität zu erhalten. Dafür dient Register

0x80. Mit diesem kann das Auswertintervall konfiguriert werden. Der Initialwert von 100 führt erst einmal zu stabilen Werten, um sich mit dem Sensor vertraut zu machen. Eine Reduzierung führt zu einem agileren Verhalten des Sensors.

Die Registerwerte für die Sensoren sind vom Typ Byte. Bei einem größeren Auswertintervall wird die Anzahl der Low-Phasen den Wertebereich überschreiten. Daher werden diese erst einmal vor der finalen Bereitstellung angepasst. Dazu werden alle Werte halbiert, sofern ein Sensor den Wertebereich verlassen hat. Diese Prüfung wird für alle Sensoren durchgeführt. Die Berechnung der %-Werte erfolgt jedoch mit den Originalwerten ohne diese Anpassung.

## Suchalgorithmus

Die einfachste Möglichkeit ist, den Sensor zyklisch abzufragen und in Abhängigkeit vom gelieferten Wert für den besten Sensor den Roboter zu steuern. In RoboPro könnte dies so wie in Abb. 11 gezeigt aussehen.

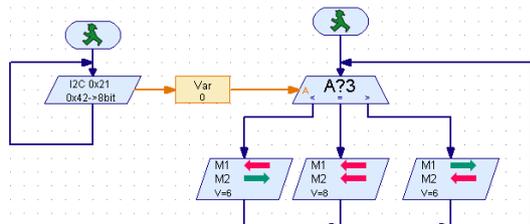
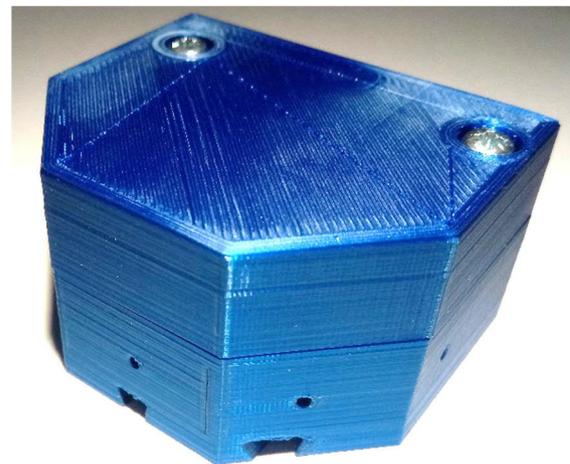


Abb. 11: ROBOPro-Programm

Hier wird jedoch immer nur hart zwischen zwei Positionen unterschieden. Wenn sich der Ball jedoch von einem Sensor zum anderen bewegt, dann wird die Qualität des einen Sensors sinken und die des anderen sich erhöhen. Auf diese Art kann die Bewegungsrichtung ermittelt werden und schon im Vorfeld eine Richtungskorrektur erfolgen. Der Algorithmus hierfür ist deutlich umfangreicher als die simple Auswertung rein über den besten Sensorwert. Der Sensor könnte bei fallenden Werten schon in die Richtung des Sensors gedreht werden, dessen Werte steigen.

## Gehäuse

Beim Gehäuse ist es wichtig, dass die internen Kammern der Sensoren untereinander abgeschirmt sind. Andernfalls führt der Einfall des Infrarot-Signals zu einer fehlerhaften Erkennung bei einem anderen Sensor. Direkt vor jedem Sensor ist ein Loch im Gehäuse. Je größer der Durchmesser, desto besser ist der Empfang. Allerdings verbessert sich damit auch der Empfang des Nachbarsensors. Hier muss also ein Kompromiss zwischen der Lochgröße, dem Empfang des eigenen und dem des benachbarten Sensors gefunden werden.



Derzeit befindet sich der Sensor im Praxistest; er kann im [Webshop](#) bezogen werden.

## Quellen

- [1] Andreas Kempf: [RoboCup German Open](#). ft:pedia 2/2015, S. 4-7.
- [2] RoboCup German Open: [RoboCup Junior Soccer](#).
- [3] RoboCup: [RoboCup Junior Soccer Rules 2020](#).
- [4] Vishay Semiconductors: [IR Receiver Modules for Remote Control](#). Data-sheet Rev. 1.7, 24.09.2018.
- [5] Stefan Fuss, Christian Bergschneider, Björn Gundermann, Christian Lauff, Rudenz Schulz: [Plug & Play am I<sup>2</sup>C-Bus mit dem ftExtender](#). ft:pedia 2/2018, S. 55-59.

Elektronik

## Rollende Steine – die ftcSoundBar

Christian Bergschneider, Stefan Fuss, Oliver Schmiel

*„Der Takt setzte sich fort, sprang von Bewusstsein zu Bewusstsein, schnippte mit den Fingern und kräuselte die Lippen. Lebendige Musik. Sich wild und ungehemmt entfaltende Musik mit Steinen drin... Endlich frei! Die Klänge rasten von Kopf zu Kopf, drangen durch die Ohren ein und setzten ihren Weg zum Rautenhirn fort.“ [1]*

In Terry Pratchetts Klassiker „Rollende Steine“ ist Musik der Antrieb der ganzen Scheibenwelt und Auslöser des Urknalls. Selbst der Tod legt Wert auf gute, wilde Musik. Er würde sich deshalb vehement dagegen wehren, den im TXT verbauten Piezo-Beeper mit dem Begriff „Sound-Wiedergabe“ in Verbindung zu bringen. Auch in der fischertechnik-Community wird immer wieder diskutiert, die vorgegebenen Sounds in ROBOPRO zu pimpen. Doch ein Beeper bleibt ein Beeper und kann nur Geräusche.

### Die ftcSoundBar

Das ftcSoundBar-Projekt setzt deshalb auf eine externe Soundkarte mit echten Boxen und erreicht so ungeahnte Klangwelten (Abb. 1).



Abb. 1: Die ftcSoundBar mit beiden Boxentypen

Wie der Thread im Forum um den ft-Swarm zeigt, haben bereits einige Mitglieder der Community die ESP32-Prozessoren für sich

entdeckt. Sie sind leistungsfähiger als die Atmel-Prozessoren auf den Arduino-Boards und glänzen zusätzlich mit Dual Core Bluetooth und integriertem Wifi. Da der Prozessor „alles drin“ hat, wird er gerne für SmartHome- und IoT-Anwendungen verwendet.

Der Hersteller espressif hat verschiedene Development Boards aufgelegt, u. a. eine Serie von Audiobords inklusive eines umfangreichen Frameworks. Die Hardware der ftcSoundBar ist ein solches Development Board – das LyraT (Abb. 2). Es enthält bereits alle Bauteile für eine gute Soundkarte. Für den kristallklaren Sound ist ein eigener Audio Codec zuständig; der Prozessor muss nur schnell genug den Stream liefern.

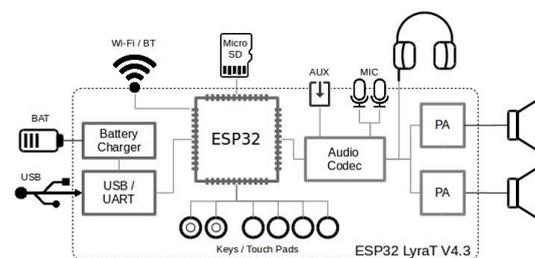


Abb. 2: Blockschaltbild der LyraT-Hardware [2]

Der Audio Codec kann auch als Eingang genutzt werden. Eine Spracheingabe ist mit dem Board technisch möglich, aber bisher nicht implementiert. Mit einem 3D-ge-

druckten Gehäuse und Boxen mit fischertechnik-Nuten wird aus dem LyraT die ftcSoundBar.

Die Boxen gibt es in zwei Ausführungen (Abb. 3): Die linke Variante ist nur einen Baustein tief, die rechte Variante hat durch das größere Gehäuse und einen anderen Speaker einen Tick bessere Bässe.



Abb. 3: Die beiden Boxen-Typen für die ftcSoundBar

Eine SD-Karte speichert die Sounddateien. Sie können entweder über den auf der ftcSoundBar integrierten Webserver via Browser oder direkt mit ROBOPRO-Kommandos abgespielt werden. Für die Fans der cfw gibt es eine Python Library. Unterstützt werden die gängigen Audioformate wie z. B. mp3, ogg und wav.

### Spannungsversorgung

Am einfachsten wird die ftcSoundBar über ein USB-Netzteil versorgt. Aufgrund des hohen Strombedarfs für die 2 x 3W Endstufe sollte mindestens ein 15 W-Netzteil verwendet werden. Alternativ kann ein flacher 3,7 V-LiPo-Akku mit zweipoligem Stecker im Gehäuse untergebracht werden. Eine Ladeelektronik ist auf dem Board vorhanden. Für akzeptable Laufzeiten wird

ein Akku mit mindestens 500 mAh benötigt.

### Anschluss

Die ftcSoundBar wird über Wifi angesprochen. Sie muss dafür ins heimische WLAN integriert werden. Der Standalone-Betrieb am TXT im AP-Modus ist auch möglich, sodass auf Conventions und Ausstellungen keine weitere Hardware benötigt wird. Außerdem kann die ftcSoundBar ohne Kabelsalat an einer beliebigen Stelle im Modell eingebaut werden.<sup>2</sup>

Ursprünglich sollte die ftcSoundBar über den I<sup>2</sup>C-Bus an den TXT angeschlossen werden. Die vorbereitete Schnittstelle – ein 10-poliger Stecker – ist am Gehäuse auch schon vorgesehen. Leider verhindert ein Bug im espressif-Framework bislang den Betrieb als I<sup>2</sup>C-Slave. Beim TXT ersetzt die Wifi-Integration die I<sup>2</sup>C-Schnittstelle; TX und ftDuino können leider noch nicht mit der ftcSoundBar kommunizieren.

### Integrierter Webserver

Der Zugriff auf den integrierten Webserver ist einfach: Im Browser <http://ftcSoundBar> aufrufen.

Die Webseite zeigt die erkannten Musiktitel bzw. Tracks (Abb. 4). Die Tracks werden durch Anklicken abgespielt; über die Icons können verschiedene Funktionen wie z. B. die Lautstärkeregelung ausgewählt werden.

Mit dem Button „Einstellungen“ gelangt man auf die Setup-Seite (Abb. 5). Hier können die Wifi-Parameter eingestellt werden. Für ein Firmwareupdate wird die Datei ftcSoundBar.bin von github auf die SD-

<sup>2</sup> Auch moderne Fertigungsanlagen verzichten immer mehr auf komplexe Datenbusse und nutzen kabellose Übertragungsprotokolle. An bewegten Elementen entfällt die „Schleifringübertragung“ der Steuersignale; es muss nur noch die Spannungsversorgung sichergestellt werden. Wifi oder der neue Mobilfunkstandard 5G ersetzen die Steuerleitung. Die Wifi-Übertragung aus der fertigen Industrie ist natürlich

auch bei großen fischertechnik-Modellen eine gute Möglichkeit, Komplexität und damit auch Fehleranfälligkeit zu reduzieren. Man baut nur noch einzelne Module mit einer kleinen, autarken Steuereinheit und lässt diese untereinander per Wifi kommunizieren. Et voilà: Die Idee des ftSwarm ist geboren.

Karte geladen. Über den Knopf „update firmware“ speichert man die neue Firmware im Flash-Speicher der ftcSoundBar.

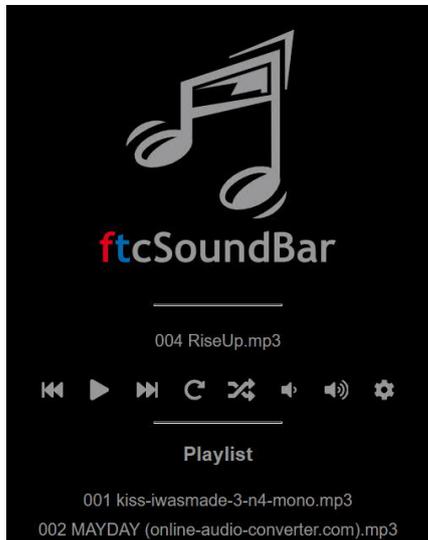


Abb. 4: Bedienung über die Weboberfläche

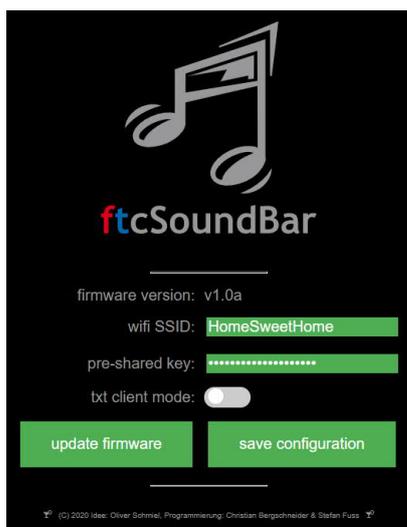


Abb. 5: Setup-Seite

## REST-API

Der TXT steuert die ftcSoundBar über Wifi und eine REST-API. Das REST-Protokoll

<sup>3</sup> Das REST-API-Protokoll steckt heute hinter jeder modernen Webseite in Angular oder React und ist Basis für alle Smartphone-Apps. Als Steuerprotokoll ist es – wie bei der ftcSoundBar – eine gute Wahl, wenn sich immer nur zwei Systeme in einem Master-Slave-Betrieb austauschen. In einer komplexen Umgebung wird normalerweise das MQTT-Protokoll als virtueller Datenbus verwendet. Wir haben MQTT nicht

basiert auf dem HTTP-Protokoll und tauscht Daten nicht in HTML, sondern stattdessen in Textform zwischen Sender und Empfänger aus.<sup>3</sup>

Das HTTP-Protokoll ist mehr als die reine HTML-Syntax: Bei einer Anfrage an einen Server gibt das Protokoll auch eine Verarbeitungsmethode mit. Dabei bedeutet GET das Abrufen von Informationen. So beantwortet beispielsweise der Webserver [www.ftcommunity.de](http://www.ftcommunity.de) eine GET-Anfrage an die URL „/ftpedia“ mit dem HTML-Text der ftpedia-Webseite.

Mit PUT werden Daten gesendet, z. B. nach dem Ausfüllen eines Formulars im Browser. Dabei werden die Daten aus dem Formular als Parameter an den Webserver übertragen.

Die Abfrage der aktuellen Lautstärke an der ftcSoundBar ist daher nur eine HTTP-GET-Anfrage. Der TXT sendet an den Server ftcSoundBar die Anfrage

```
GET /api/v1/volume
```

und erhält als Antwort die aktuelle Lautstärke:<sup>4</sup>

```
{ "volume": 50 }
```

Verwendet man anstelle des GETs ein PUT und gibt den Parameter

```
{ "volume": 70 }
```

mit, so setzt man umgekehrt die Lautstärke auf den Wert 70.

Auch die Funktionen der ftcSoundBar-Webseite lösen wieder nur REST-API-Aufrufe an sich selbst aus.

eingesetzt, da damit auch die technischen Randbedingungen komplexer werden.

<sup>4</sup> Die Daten werden im JSON-Format ausgetauscht. Damit lassen sich sehr komplexe Datenstrukturen aufbauen. Durch die Repräsentation in Textform ist JSON unabhängig von Programmiersprache oder Betriebssystem.

## Nachbau der ftcSoundBar

Für den Nachbau werden nur einfache Lötkenntnisse und 3D-gedruckte Gehäuseteile benötigt. Wer dies nicht selbst machen möchte, kann das fertige Gerät bei Björn im Webshop bestellen [5]. Für die Maker folgt nun die Beschreibung des Nachbaus des Gerätes. Wer ein fertiges Gerät nutzt, liest einfach ab dem Abschnitt „Wifi konfigurieren“ weiter.

### Material

Die Platine wird als „esp32 LyraT“ [2] im Internet über verschiedene Shops angeboten. Achtung: Das LyraT mini hat eine andere Form, sodass das Gehäuse nicht passt. Außerdem verfügt das Mini nur über einen Monoausgang.

Den silbernen Speaker gibt es bei Digi-Key [6], der Schwarze ist von gikfun (und z. B. bei Amazon erhältlich) [7]. Es kann grundsätzlich auch jeder andere 4-Ohm-Lautsprecher verwendet werden; die 3D-Druckdateien für die Boxen müssen dann gegebenenfalls angepasst werden.

Darüber hinaus werden noch acht Bundhülsen, ein 10-poliger Wannenstecker und 14 Blechschrauben der Größe 2,2 x 4,5 mm benötigt.

### Zusammenbau der SoundBar

Als erstes werden die STL-Dateien für das Gehäuse und die Boxen bei thingiverse [4] heruntergeladen und gedruckt.

Die vier Bundhülsen für das Gehäuse der ftcSoundBar werden dann an kurze Kabel gelötet. Der Kabelquerschnitt ist bei 3W-Speakern fast egal; normale fischertechnik-Kabel sind sehr gut geeignet.

Die beiden zweipoligen Lautsprecherstecker auf dem LyraT-Board (Position 1 in Abb. 7) werden ausgelötet und dafür die Lautsprecherkabel von unten eingelötet.



Abb. 7: Das geöffnete Gehäuse zeigt die Befestigung der Bundhülsen und des 10-poligen Steckers

Am 10-poligen I<sup>2</sup>C-Stecker werden die Pins 1, 5 und 6 mit einem Stück Flachbandkabel belegt. Das Lötten ist einfacher, wenn zunächst eine 10-polige Buchse auf den Stecker gesteckt wird. Auf der Unterseite des I<sup>2</sup>C-Steckers (Position 2 in Abb. 7) werden die Kabel angeschlossen (Abb. 8).

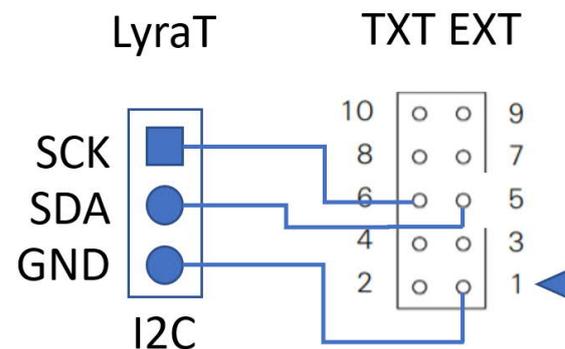


Abb. 8: Steckerbelegung des I<sup>2</sup>C-Bus

Die Stecker werden in die Auflagen des Gehäuses gesteckt und mit Repair- oder Sekundenkleber befestigt (Abb. 7).

### Zusammenbau der Boxen

Beim Zusammenbau der Boxen werden die Bundhülsen zunächst in das Halteplättchen gesteckt und die Kabel an Bundhülsen angelötet. Dabei ist darauf zu achten, dass der lange Teil der Bundhülse durch das Halteplättchen gesteckt wird.

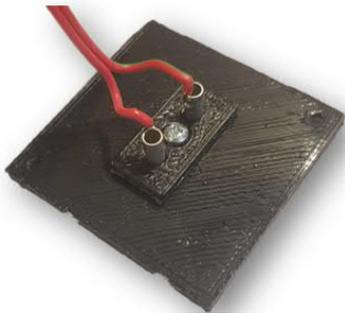


Abb. 9: Befestigung der Kabel am Gehäusedeckel der Box

Anschließend werden die Bundhülsen auf den Gehäusedeckel gesteckt und verschraubt (Abb. 9). Das andere Ende des Kabels wird nun an den Lautsprecher gelötet. Die schwarzen Lautsprecher werden mit Hilfe von zwei Haltblättchen an der Box fixiert, der silberne Lautsprecher wird nur geklemmt. Gehäusedeckel verschrauben und fertig ist die Box.

## Inbetriebnahme

### Firmware aufspielen

Um die Firmware selbst zu kompilieren und zu flashen sind einige Schritte nötig. Wer bereits ein geflashtes Gerät besitzt, kann die fertige Software als Release in [3] herunterladen und wie oben beschrieben über die SD-Karte flashen.

Die beiden espressif-Frameworks lassen sich am einfachsten über git installieren:<sup>5</sup>

```
> git clone -b v4.1
https://github.com/espressif/esp-idf.git
> cd esp-idf
> install.bat
> git clone
https://github.com/espressif/esp-adf.git
```

In der Datei esp-isf/export.bat muss anschließend noch die Variable ADF\_PATH mit dem Pfad auf das esp32-adf-Framework gesetzt werden. Die Firmware wird mit

```
> git clone
https://github.com/elektrofuzzis/ftcSoundBar.git
> esp-idf/export.bat
> cd ftcSoundBar/firmware
> idf.py build
geklont und kompiliert.
```

Die ftcSoundBar wird nun mit den beiden USB-Anschlüssen an den PC (UART) und am Netzteil (POWER) angeschlossen. Dann kann man den Power Switch auf der Platine einschalten und im Gerätemanager die emulierte serielle Schnittstelle bestimmen.

### Der Befehl

```
> idf.py -p <COM-Port> flash
```

flasht die Firmware. Über den RESET-Button wird die ftcSoundBar gebootet. Wenn nun die grüne LED am Prozessor blinkt, war das Flashen erfolgreich.

### Wifi konfigurieren

Um die ftcSoundBar zum ersten Mal ins Wifi-Netz zu bekommen, muss auf der SD-Karte die Konfigurationsdatei bearbeitet werden:

```
ftcSoundBar.conf
```

Dazu wird die SD-Karte mit fat32 formatiert, in die ftcSoundBar gesteckt und diese mit dem RESET-Button neu gestartet. Der Neustart erzeugt eine Beispieldatei, so dass diese nun am PC bearbeitet werden kann.

In der Datei müssen nur die Parameter WIFI\_SSID und WIFI\_PASSWORD gesetzt werden; die restlichen Parameter werden nicht verändert. Nun erneut die ftcSoundBar booten. Ist die grüne LED dauerhaft an, kann die ftcSoundBar über Wifi erreicht werden: <http://ftcSoundBar>. Zur Fehlersuche kann man die Konsole der ftcSoundBar einsehen mit

```
> idf.py -p <COM-Port> monitor
```

<sup>5</sup> Das esp-adf-Framework ist nur zu bestimmten esp-idf-Versionen kompatibel. Deshalb muss beim Klonen des esp-idf-Frameworks darauf

geachtet werden, dass die zum adf-Framework passende Version verwendet wird.

## Nutzung in ROBOPro

REST-APIs werden von ROBOPro nicht direkt unterstützt. Eine C-Library auf dem TXT stellt daher die notwendige Funktionalität zur Verfügung. Die Befehle in der ROBOPro-Bibliothek sind nur „Fassade“ – sie rufen die Kommandos der Library auf.

### Hochladen der ROBOPro-Library

Zur Programmierung unter ROBOPro muss dafür die Library libftcSoundBar.so über den Webserver des TXT in das Verzeichnis „libs“ geladen werden. Dazu muss der TXT in den Einstellungen als Wifi-Client konfiguriert werden. Anschließend wird die IP-Adresse des TXT im Browser eingegeben.<sup>6</sup>

Der TXT verlangt nun nach einer Anmeldung. Der Benutzername ist „TXT“, das Passwort die vierstellige Seriennummer des TXT. Nach dem Wechsel in das Verzeichnis „libs“ kann mit dem +-Icon die libftcSoundBar.so-Library hochgeladen werden.

Wird zum ersten Mal eine .so-Library hochgeladen, fehlen die Schreibrechte für dieses Verzeichnis. In diesem Fall muss man sich über ein SSH-Terminal (wie z. B. putty) mit dem TXT verbinden.<sup>7</sup> Das Kommando

```
chmod 744 /opt/knobloch/libs
```

setzt die richtigen Rechte; anschließend lässt sich die Library hochladen.

Die eigentliche ROBOPro-Bibliothek wird im eigenen Bibliotheksverzeichnis gespeichert.<sup>8</sup> Auf die Programmelemente kann nun unter „Elementgruppen“ im Bereich

„eigene Bibliotheken/ftcSoundBar“ zugegriffen werden.

### Steuerung der ftcSoundBar

Das Beispielprogramm in Abb. 6 spielt zunächst zehn Sekunden lang Track 1, danach wird für zehn Sekunden Track 5 abgespielt. Zum Schluss wird der Shuffle-Mode gesetzt, d. h. die ftcSoundBar spielt nun die Tracks in zufälliger Reihenfolge.

Auch wenn das ROBOPro-Programm an dieser Stelle endet, bleibt die ftcSoundBar als eigenständiges Device im Shufflemodus und spielt weiterhin Tracks ab.

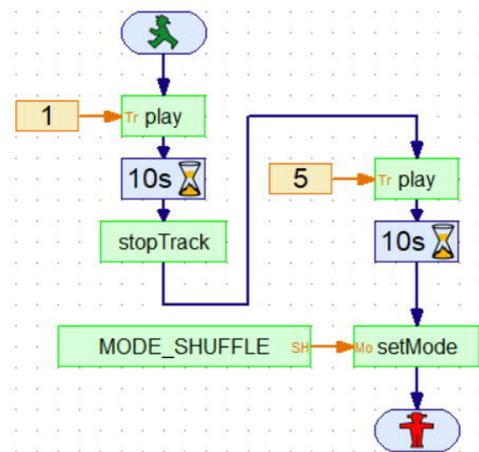


Abb. 6: Ein einfaches ROBOPro-Programm

Wie auch bei der Nutzung des I<sup>2</sup>C-Interfaces in ROBOPro tritt bei der Verwendung von externen Libraries der „Fünf-Sekunden-Bug“ auf. Liegen zwischen der Ausführung von zwei Library-Aufrufen fünf Sekunden oder mehr, so wird der zweite Aufruf von ROBOPro nicht durchgeführt.<sup>9</sup>

<sup>6</sup> Im Menü „Einstellungen“, „Netzwerk“, „WLAN“ wählen; die benötigte IP-Adresse steht im Bereich WLAN/Client.

<sup>7</sup> Username ROBOPro, Passwort ROBOPro.

<sup>8</sup> Das Bibliotheksverzeichnis kann in ROBOPro im Menü unter „Datei/Eigenes Bibliotheksverzeichnis“ eingestellt werden.

<sup>9</sup> In der ROBOPro-Bibliothek ftcSoundBar.rpp gibt es eine Funktion, die alle vier Sekunden die Library im Hintergrund nutzt und so den Bug umgeht.

## Schlussbemerkung

Das Wifi-Passwort wird auf der SD-Karte im Klartext gespeichert. Die Webseite der ftcSoundBar und die Ansteuerung via Wifi erfolgen unverschlüsselt und ohne Autorisierung über das HTTP-Protokoll. Im öffentlichen Internet wäre dies ein No-Go. Auf höhere Sicherheitsstandards wurde aber bewusst verzichtet. Die Firmware, der Aufbau und die Inbetriebnahme der ftcSoundBar werden so massiv vereinfacht.

Da die ftcSoundBar nur im eigenen Wifi-Netzwerk betrieben wird, besteht kein Risiko, dass das Wifi-Passwort unerwünscht in den Händen Dritter landet. Somit ist die ftcSoundBar in diesem Umfeld hinreichend sicher.

## Quellen

- [1] Terry Pratchett: *Rollende Steine*. Goldmann Verlag, 1996
- [2] espressif: [Dokumentation des LyraT Development Boards](#).
- [3] Christian Bergschneider, Stefan Fuss, Oliver Schmiel: [ftcSoundBar-Projekt](#). github.com
- [4] Christian Bergschneider, Stefan Fuss, Oliver Schmiel: [ftcSoundBar & loudspeaker boxes](#). thingiverse.com
- [5] [Webshop](#) von Björn Gundermann
- [6] [Silberner Lautsprecher](#) von Digi-Key
- [7] [Schwarzer Lautsprecher](#) bei Amazon



Abb. 10: „Pink Tom“ rocken die Hütte

Elektronik

## Elektronik-Module (Teil 5)

Hans-Christian Funke

Die Elektronikmodule können auch hervorragend zusammen mit einem programmgesteuerten Controller (ROBO, TX/TXT, Arduino) eingesetzt werden. In diesem Teil zu den Elektronikmodulen stelle ich einige Einsatzmöglichkeiten und Modelle mit einem vollständigen Programm vor.

Die Elektronikmodule arbeiten mit positiver Logik. In den Ausführungen und Beschreibungen werden folgende Bezeichnungen verwendet:

- logisch „0“ = low = minus (-) = Masse = GND
- logisch „1“ = high = plus (+) = Betriebsspannung =  $V_{dd}$
- FlipFlop = FF
- MonoFlop = MF

### Erweiterung

Aktuell gibt es drei Elektronikmodule, die mit dem Ziel konstruiert wurden, als Erweiterung oder Ergänzung für Controller

eingesetzt zu werden. Grundidee war es, mit einfachen Mitteln eine Erweiterung der Anzahl von Ein- und Ausgängen für die programmgesteuerten Controller zu schaffen.

Mit Hilfe des Meldemoduls (Abb. 2) werden acht zusätzliche mit Speichern hinterlegte digitale Eingänge zur Verfügung gestellt, und mit jedem Steuermodul (Abb. 1) stehen acht zusätzliche mit Speichern hinterlegte digitale Ausgänge bereit.

Das Multiplexmodul (Abb. 3) kann als 1:8-Multiplexer oder 8:1-Demultiplexer eingesetzt werden und somit wahlweise acht verschiedene Ereignisse oder Situationen

über nur einen Eingang vom Controller abfragen oder über nur einen Ausgang vom Controller bis zu acht verschiedene Verbraucher ein- bzw. ausschalten. Das Multiplexmodul kann auch als Relais mit einem 1:8-Umschalter eingesetzt werden, weil das Multiplexmodul sogar analoge Signale durchleiten kann, jedoch nur mit einem maximalen Stromfluss von 25 mA.



Abb. 1: 70016  
Steuermodul

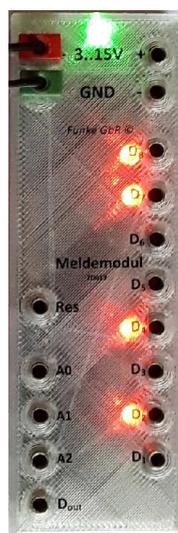


Abb. 2: 70017  
Meldemodul

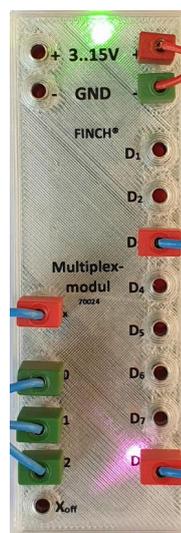


Abb. 3: 70024  
Multiplexmodul

### Das Steuermodul (70016)

Herzstück des Steuermoduls ist ein 8-bit Schieberegister (Abb. 4). Das Schieberegister verfügt über acht Daten-Ausgänge (D1-D8), die entweder den Zustand „1“ oder „0“ annehmen können (Magnet, Lampe, Kompressor „ein“ oder „aus“). Die Ausgänge können keine Stufeneinstellungen für Motoren übertragen.

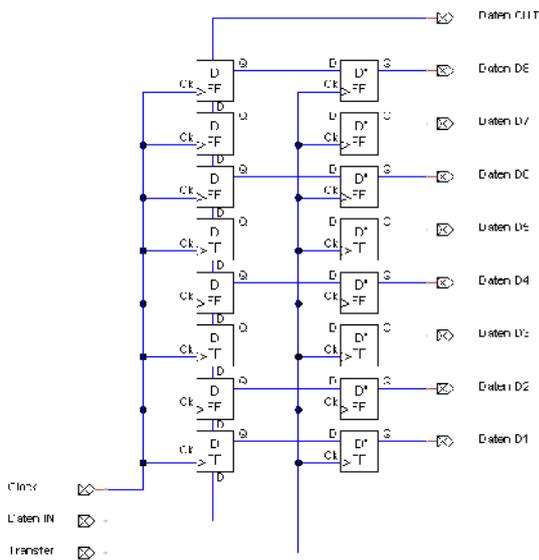


Abb. 4: Schematische Darstellung eines Schieberegisters

### Funktionsbeschreibung

Für die weitere Beschreibung empfehle ich, das Taktdiagramm (Abb. 5) zur Hand zu nehmen, um meinen Ausführungen besser folgen zu können.

Um den Zustand an den Ausgängen zu ändern, müssen die D'-FFs (linke Spalte) mit den Daten für die Ausgänge (D1-D8) versorgt werden. Dafür wird der gewünschte Zustand für den Ausgang D1 zuerst an den D<sub>IN</sub>-Eingang gelegt (low oder high). Durch einen Zustandswechsel von low auf high am Clock-Eingang übernimmt das erste D'-FF die Daten vom D<sub>IN</sub>-Eingang; alle weiteren D'-FFs übernehmen vom vorherigen D'-FF die Daten, so dass die Daten wandern bzw. geschoben werden (Schieberegister). Durch Wiederholung des Vorganges und nacheinander anlegen der weiteren Daten (D2-D8) an den D<sub>IN</sub>-Eingang, gefolgt von einem low-high Wechsel am Clock-Eingang, werden alle Daten in das Schieberegister geschoben.

Sind alle Daten eingeschoben, erscheinen diese noch nicht an den Ausgängen, sondern erst mit high am Transfer-Eingang werden die Daten von den D'-FFs (linke Spalte) gelesen und an die D-FFs (rechte

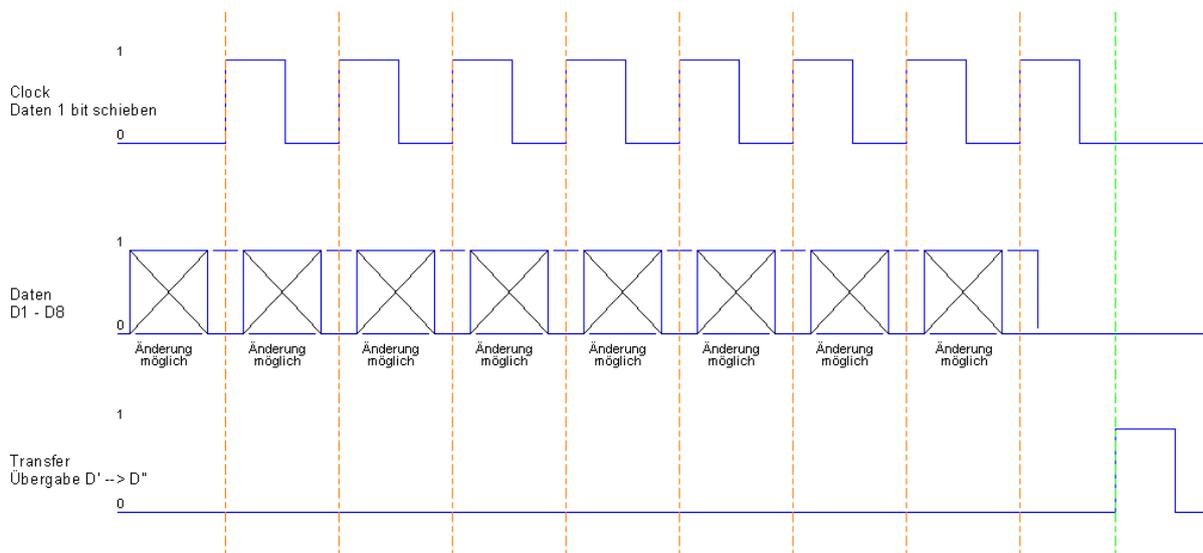


Abb. 5: Taktdiagramm zur Ansteuerung des Schieberegisters

Spalte) übergeben und gespeichert. Durch die separate Speicherung in den D“-FFs können so weitere Daten in das Schieberegister geschoben werden, ohne dass sich die Daten an den Ausgängen ändern.

Die orangefarbenen LEDs signalisieren, wenn ein Ausgang auf high liegt. Nach dem Einschalten nehmen die FFs einen undefinierten Zustand ein. Damit müssen nach dem Einschalten über den Eingang D<sub>IN</sub> erst einmal 8 Bit eingeschoben werden, um einen definierten Start-Zustand zu erhalten.

Soll der Clock-Eingang über einen mechanischen Taster angesteuert werden, dann empfiehlt es sich, den Jumper am Modul auf „A“ für analog zu stecken. Ein integrierter Tiefpass sorgt dann dafür, dass auftretende Prellimpulse abgefangen werden. Für alle anderen Fälle sollte der Jumper auf „D“ für digital stecken, in diesem Fall wird der Tiefpass überbrückt.

Wer mehr Ausgänge benötigt, kann den Ausgang D<sub>OUT</sub> mit einem D<sub>IN</sub>-Eingang eines weiteren Steuermoduls verbinden, sowie die Eingänge Clock und Transfer beider Steuermodule miteinander koppeln und erhält somit insgesamt 16 digitale Ausgänge.

### Das Meldemodul (70017)

Bei der Erweiterung der Eingänge schien mir die Lösung mit dem Schieberegister nicht sehr sinnvoll, denn es sollte möglich sein, die Eingangswerte schnell abfragen zu können. Aus diesem Grund habe ich mich entschieden, beim Meldemodul einen Multiplexer einzusetzen. So können die Eingänge zwar gezielt abgefragt werden, aber nicht gleichzeitig, und ankommende Signale können verloren gehen. Um dies zu vermeiden, brauchten die Eingänge einen Speicher, der selbst ein sehr kurzen Signal auffangen und registrieren kann. Die Speicher müssen auch wieder zurückgesetzt

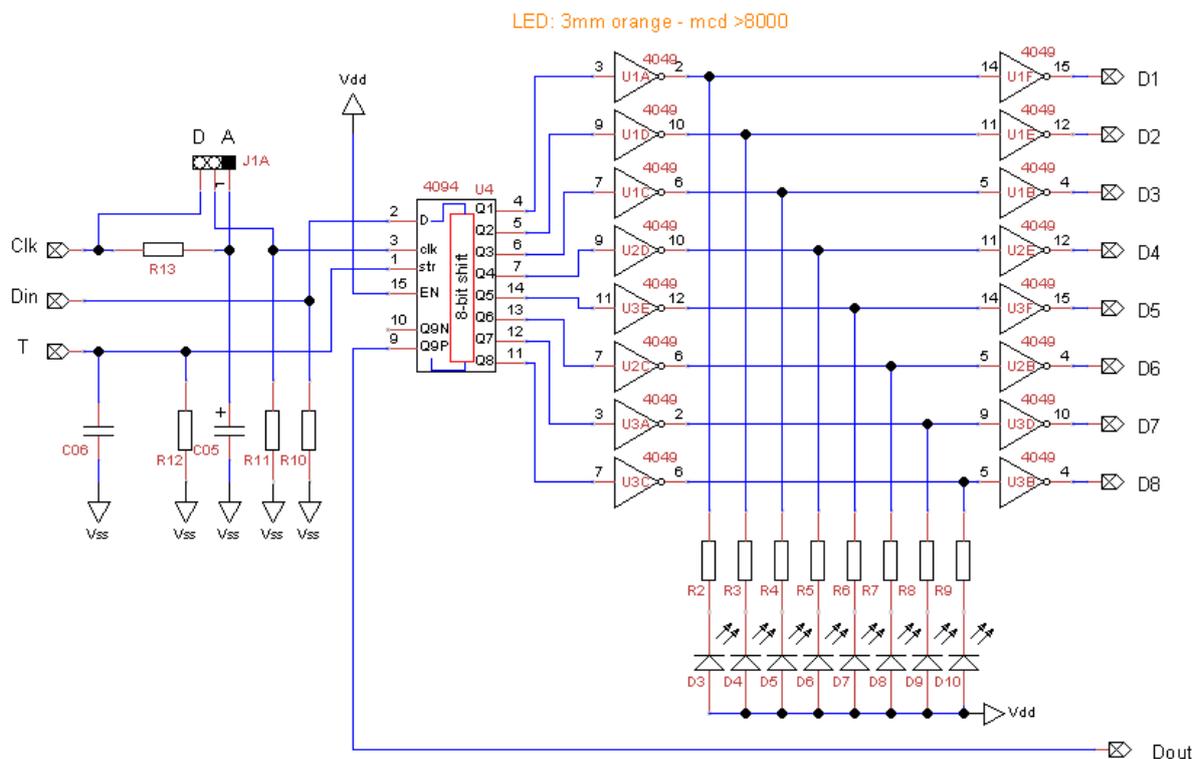


Abb. 6: Teilschaltplan des Steuermoduls

werden können, idealerweise jeder gezielt und separat für sich.

Um all diesen Anforderungen Rechnung zu tragen, hat das Meldemodul für jeden Eingang ein RS-FF bekommen, einen Multiplexer zum Abruf des Status eines jeden FFs und einen Multiplexer, um jedes FF separat zurücksetzen zu können – das Ergebnis war eine relativ komplexe Schaltung (Abb. 7).

**Funktionsbeschreibung**

Um die gesamte Ansteuerung des Meldemodules zu vereinfachen, wurden gleiche Funktionen zusammengelegt und damit die Anzahl der notwendigen Steuereingänge minimiert. Die Ansteuerung der acht Multiplexer-Eingänge erfolgt über drei Steuerleitungen A0-A2 mit Hilfe einer Binärcodierung (Tab. 1).

Eingänge				Ausgang
A <sub>0</sub>	A <sub>1</sub>	A <sub>2</sub>	Res	D <sub>out</sub>
0	0	0	0	D1
1	0	0	0	D2
0	1	0	0	D3
1	1	0	0	D4
0	0	1	0	D5
1	0	1	0	D6
0	1	1	0	D7
1	1	1	0	D8

Tab. 1: Binärcodierung

Beide Multiplexer – auslesen und zurücksetzen der FFs – werden über die gleichen Steuerleitungen A0-A2 angesprochen oder, wie man auch sagt, adressiert.

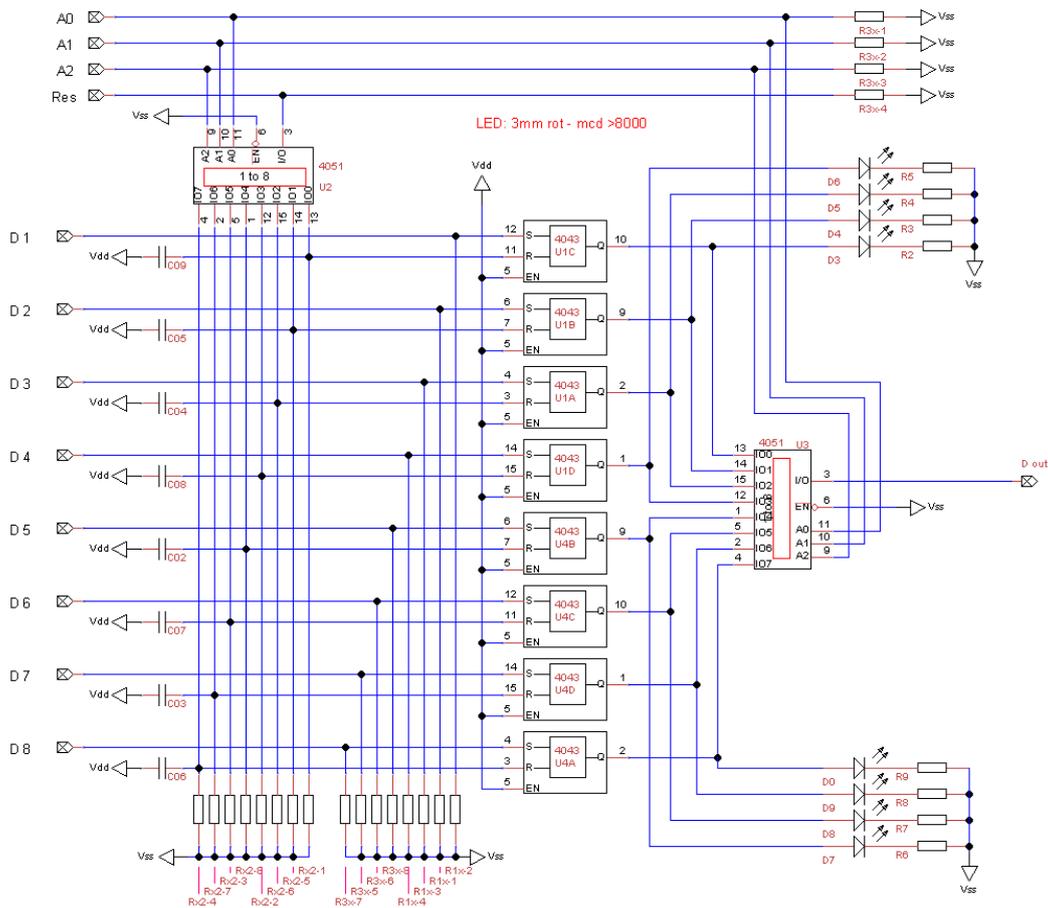


Abb. 7: Teilschaltplan des Meldemoduls

Werden Daten über den Multiplexer (U3) ausgelesen, kann der Zustand eines FFs über den Ausgang D<sub>OUT</sub> abgefragt werden; am Eingang Res muss währenddessen low anliegen.

Möchte man ein FF zurücksetzen, erfolgt dies über den Multiplexer (U2) und der Eingang Res muss auf high liegen. Je nach Adressierung der Steuerleitung wird das entsprechende FF zurückgesetzt.

Die FFs verfügen über einen Einschalt-Reset, so dass alle FFs nach dem Einschalten zurückgesetzt sind. Sobald high an einer der Eingänge erscheint, wird damit die Setz-Funktion des dahinterliegenden FFs ausgelöst und durch eine rote LED signalisiert.

## Das Multiplexmodul (70024)

Das Innenleben des Multiplexmoduls basiert auf einen bidirektionalen 1:8-Multiplexer, so dass dieser gleichzeitig auch als 8:1-Demultiplexer eingesetzt werden kann. Man kann sich den Multiplexer vergleichsweise auch als ein Relais mit acht Umschaltkontakten vorstellen, wobei die Richtung des Stromflusses keine Rolle spielt und die Spannung einen analogen Wert bis zur Betriebsspannung annehmen darf.

### Funktionsbeschreibung

Ebenso wie das Meldemodul verfügt das Multiplexmodul über drei Steuer- bzw. Adresseingänge (A0-A2), die mittels Binärcodierung (siehe Tab. 1) die Auswahl des jeweiligen Kontaktes bestimmen (D1-D8),

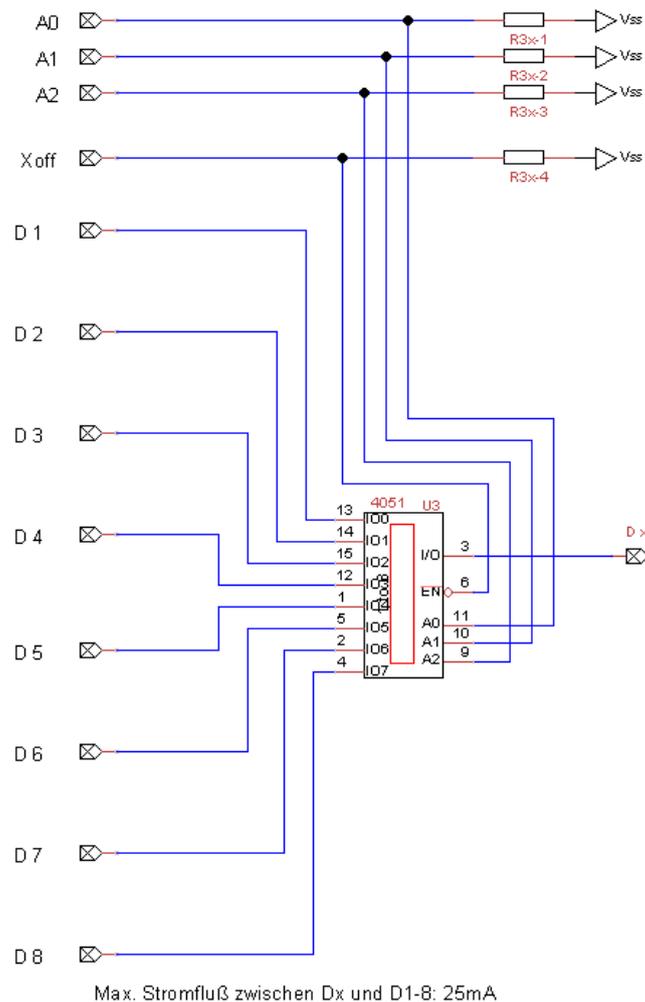
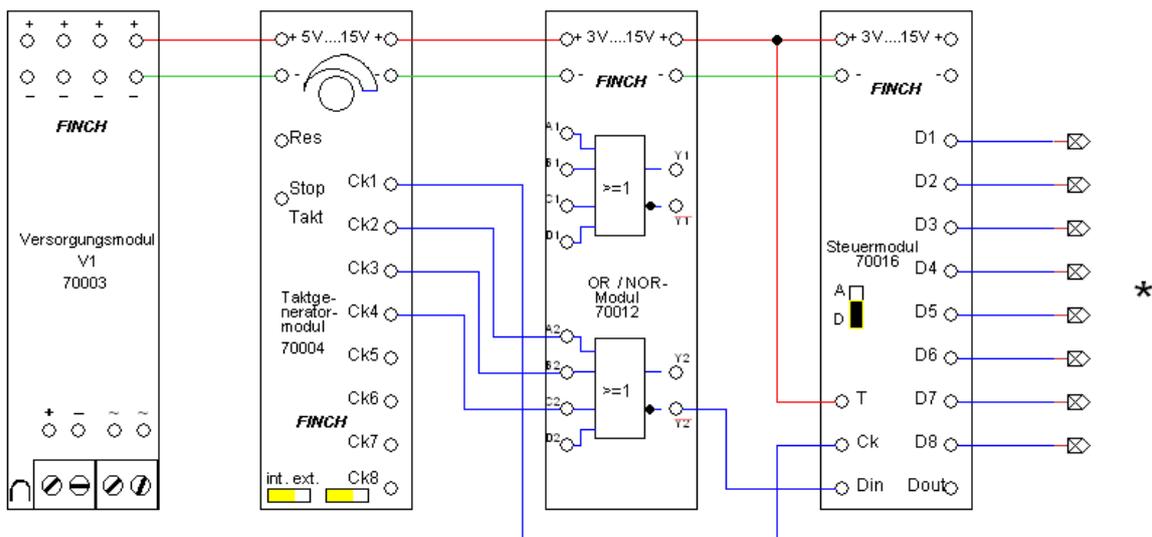


Abb. 8: Teilschaltplan des Multiplexmoduls

der mit dem Einzelkontakt Dx verbunden wird.

Bei einem Relais ist immer einer der ausgewählten Kontakte mit dem Einzelkontakt verbunden; beim Multiplexmodul kann durch Anlegen von high an Eingang X<sub>off</sub> der Einzelkontakt von allen anderen Kontakten getrennt werden.

Einzige Einschränkung gegenüber dem Relais ist, dass nur Spannungen zwischen 0 V und der Betriebsspannung V<sub>dd</sub> (max. 15 V) geschaltet und max. 25 mA Stromfluss geschaltet werden können.



- \* Die Ausgänge können direkt mit einer LED von max. 3 mA verbunden werden. Sollen andere Verbraucher z.B. eine Lampe angeschlossen werden, müssen die Ausgänge über ein Relaismodul 70018 oder Inverter-Plus-Modul 70019 an das Steuermodul angeschaltet werden.

Abb. 9: Schaltung für ein Lauflicht ohne Controller

### Achtung

Es ist unbedingt darauf zu achten, dass die Elektronikmodule immer zuerst mit Betriebsspannung versorgt werden, bevor die einzelnen Kontakte mit Spannung beschaltet werden. Dies gilt insbesondere für das Multiplexmodul.

### Ansteuerung ohne Controller

Aus den Funktionsbeschreibungen von Steuermodul, Meldemodul und Multiplexmodul ist ersichtlich, dass ohne ein programmgesteuertes Interface oder Controller diese kaum sinnvoll eingesetzt werden können.

Allerdings können die drei Elektronikmodule z. B. in Verbindung mit dem Taktgenerator (70004) zum Leben erweckt werden – ganz ohne einen Controller.

### Beispielsteuerung für das Steuermodul

Das Steuermodul wird hier in Kombination mit dem Taktgenerator eingesetzt, um ein

Lauflicht zu generieren (Abb. 9). Die Geschwindigkeit des Lauflichtes kann über den Drehregler am Taktgenerator eingestellt werden.

### Beispielsteuerung für das Meldemodul

Für das Meldemodul habe ich für die manuelle Anwendung kein Schaltungsbeispiel parat. Dieses Elektronikmodul kann z. B. in einem Pinball oder einer Fahrstraße eingesetzt werden. Die Signale nimmt das Meldemodul auf und können vom Controller abgerufen und weiterverarbeitet werden.

### Beispielsteuerung für das Multiplexmodul

Beim Multiplexmodul kann die Auswahl gut über Schalter realisiert werden, wobei mit einem Schalter zwei Schaltzustände (D1 oder D2), mit zwei Schaltern vier Schaltzustände (D1-D4) und mit drei Schaltern acht Schaltzustände (D1-D8) eingestellt werden können. Die eine Seite vom Schalter muss an Plus angeschlossen

werden und die andere Seite je an einen Eingang A0-A2.

Möchte man eine Automatik einbauen, kann auch hier der Taktgenerator helfen, indem man drei Ausgänge des Taktgenerators mit den Eingängen A0-A2 vom Multiplexmodul verbindet.

Als praktisches Beispiel habe ich einen Morsezeichensender entworfen. Hier sieht man auch gleich, dass bei entsprechender

Kombination der 1:8-Schalter auch auf 1:16 erweitert werden kann (Abb. 10). Das ODER-Gatter wird hier nur als Inverter eingesetzt – es kann auch ein UND-Gatter, EXODER-Gatter oder Inverter verwendet werden.

Mit dem „Start“-Taster werden die eingestellten Morsezeichen von der LED angezeigt. Folgende Einstellung ist zu beachten:

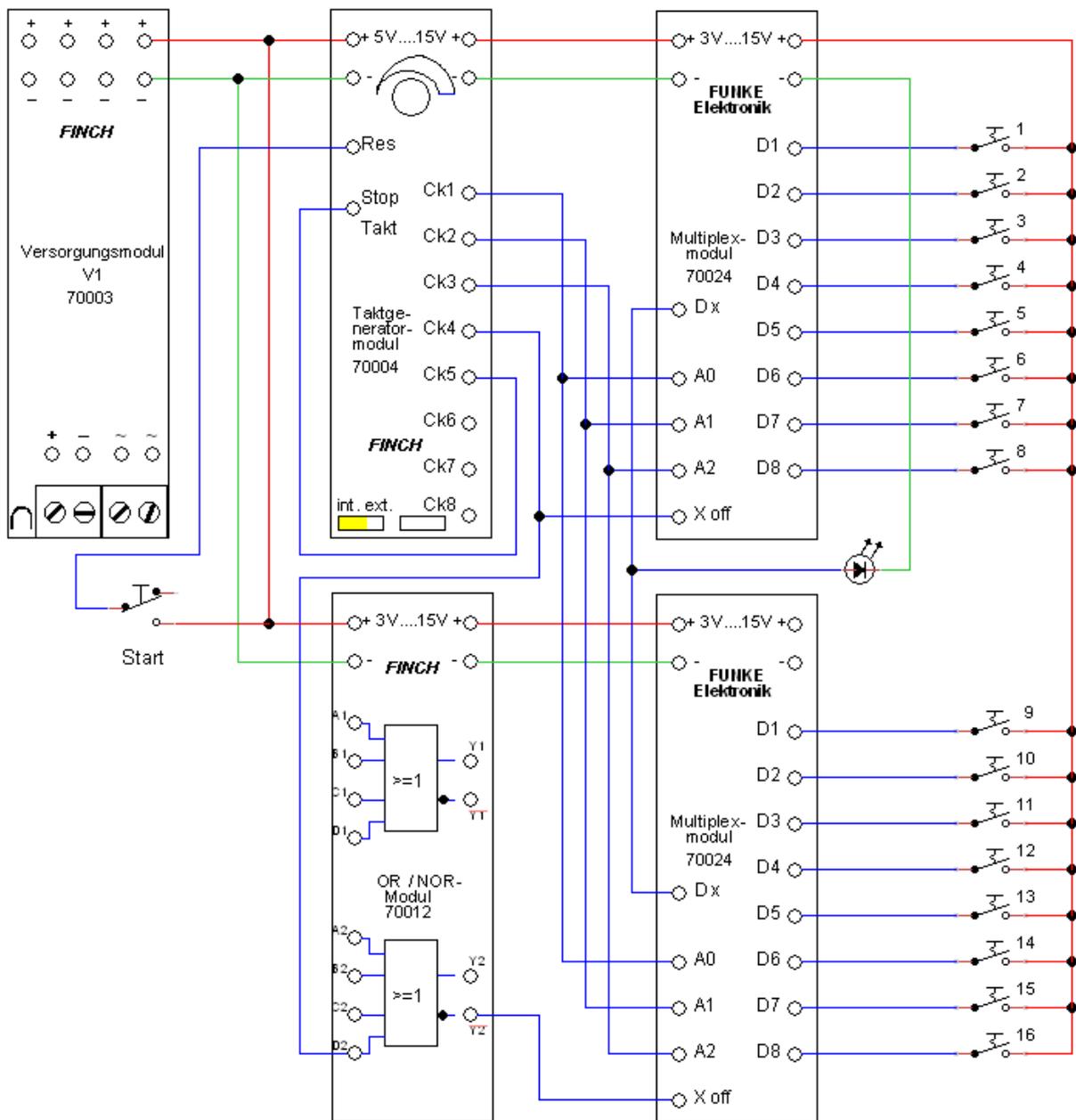


Abb. 10: Schaltung für einen Morsezeichensender ohne Controller

- Lang: Zwei aufeinander folgende Schalter auf ‚ein‘
- Kurz: Ein Schalter auf ‚ein‘
- Pause: Nach jedem ‚Lang‘ oder ‚Kurz‘ muss ein Schalter auf ‚aus‘ stehen

Hier zwei Beispiele:

Buchstabe J · - - - kurz lang lang lang

Schalter ‚ein‘ 1, 3+4, 6+7, 9+10

Buchstabe Y - · - - lang kurz lang lang

Schalter ‚ein‘ 1+2, 4, 6+7, 9+10

Weitere Morsezeichen sind in Tab. 2 aufgelistet.

### Programmierung und Ansteuerung der Elektronikmodule

In diesem Teil werde ich den Einsatz der Elektronikmodule unter Verwendung eines Controllers demonstrieren. Dafür habe ich drei Modelle, für die ich euch die Schaltung bzw. das Anschlussschema und auch ein dazugehöriges Programmbeispiel mit Erklärungen liefere. Für die Steuerung der Modelle kommt ein TXT-Controller zum

Einsatz; für das Modell „Demo“ verwende ich das ROBO-Interface.

Die Programmierung ist nicht mein Fachgebiet, daher sind die hier vorgestellten Programme zwar alle in der Praxis geprüft und funktionieren, möglicherweise können einige Stellen aber noch eleganter oder effizienter gelöst werden. Mein Hauptanliegen ist es, für die Ansteuerung der Elektronikmodule eine mögliche Programmversion vorzustellen, um damit eine Grundlage für die Umsetzung weiterer Ideen und Experimente zu liefern.

Bei den Modellen handelt es sich um das „Pianophone“ und die „Digitale Uhr“, die ich beide in der letzten Ausgabe vorgestellt habe. Als drittes zeige ich ein Demo-Modell, das ich auf den Conventions einsetze, um die Funktion von Steuer- und Meldemodul zu zeigen. Hier die Elektronikmodule, die durch das Programm angesteuert werden:

- Digitale Uhr: Ansteuerung von 70001 Zählermodulen.
- Jukebox (Pianophone): Ansteuerung eines 70024 Multiplexmodules.
  - Demo: Demonstration der Funktionsweise von 70016 und 70017.

#### Modell „Digitale Uhr“ mit Steuerung über einen Controller

Eine ausführliche Aufbauanleitung für das Modell ist in der ft:pedia-Ausgabe 03/2020 abgedruckt. Abb. 11 zeigt im Schaltplan, wie der Controller mit dem Modell zu verbinden ist, damit das vorgestellte Programm zusammen mit dem Modell funktioniert. Es sind nur minimale Änderungen in der Schaltung erforderlich, und die Verbindungen zum Controller herzustellen.

Das gesamte Programm (Hauptprogramm mit Unterprogrammen) findet ihr als PDF-Darstellung und als fertiges Programm (.RPP-Datei) auf der [www.ftcommunity.de](http://www.ftcommunity.de) im Abschnitt

T -	M - -	O - - -	CH - - - -
		G - - -	Ö - - - -
	N - ·	K - · -	Q - - - -
		D - · -	Z - - - -
	A · -	W · - -	Y - · - -
		R · - ·	C - - - -
E ·	I · ·	U · · -	X - · - -
		S · · ·	B - · · -
			J · - - -
		P · - - -	
		Ä - - - -	
		L · - - -	
		Ü · - - -	
		F · - - -	
		V · - - -	
		H · - - -	

Tab. 2: Morse-Alphabet [1]

ft:pedia unter der Rubrik „Ergänzende Downloads“.

Das Hauptprogramm besteht aus zwei Prozessen: Einer zur Taktung und einer zur Steuerung der Uhr. Die Steuerung, die das eigentliche Programm darstellt, hat einen einleitenden Teil, der die Zähler auf null zurücksetzt (Unterprogramm *Reset*) und eine Startzeit für die Anzeige festlegt (Unterprogramm *Setzwert*). Anschließend läuft das Programm in eine Endlosschleife, die ständig aus dem Prozess Taktung, der den sekundengenauen Takt liefert, die Variable *Sek.Zähler* abfragt.

Die aktuelle Uhrzeit auf der Anzeige läuft synchron mit den Variablen *Stunde*, *Minute* und *Sekunde*. Ist die Variable *Sek.Zähler* einen Takt weiter, dann wird die Uhrzeit auf allen Anzeigen/Variablen geprüft und nach Bedarf angepasst.

Der Ausgang O1 liefert eine Sekunden-taktung, die für das Modell nicht benötigt wird. Der Zeitwert von 0,485 Sekunden vor dem Einschalten des Ausgang O1 ist ein empirisch ermittelter Wert, um eine genaue Taktung von 1 Sekunde zu erhalten. Dieser Wert kann bei den unterschiedlichen Controller-Typen variieren.

Der Durchlauf durch die Programmschleife verbraucht auch Zeit und muss berücksichtigt werden; für diesen Controller beträgt der benötigte Zeitaufwand 0,015 Sekunden oder 15 ms.

Das Unterprogramm *Reset* dient zum zurücksetzen der Zählermodule. Die Programmierung ist so universell gestaltet, dass sowohl die Zähler einzeln als auch in jeder beliebigen Kombination zurück-gesetzt werden können: Die Zahle 0 bis 7 als Übergabewerte setzen bitweise kombinierbar zurück:

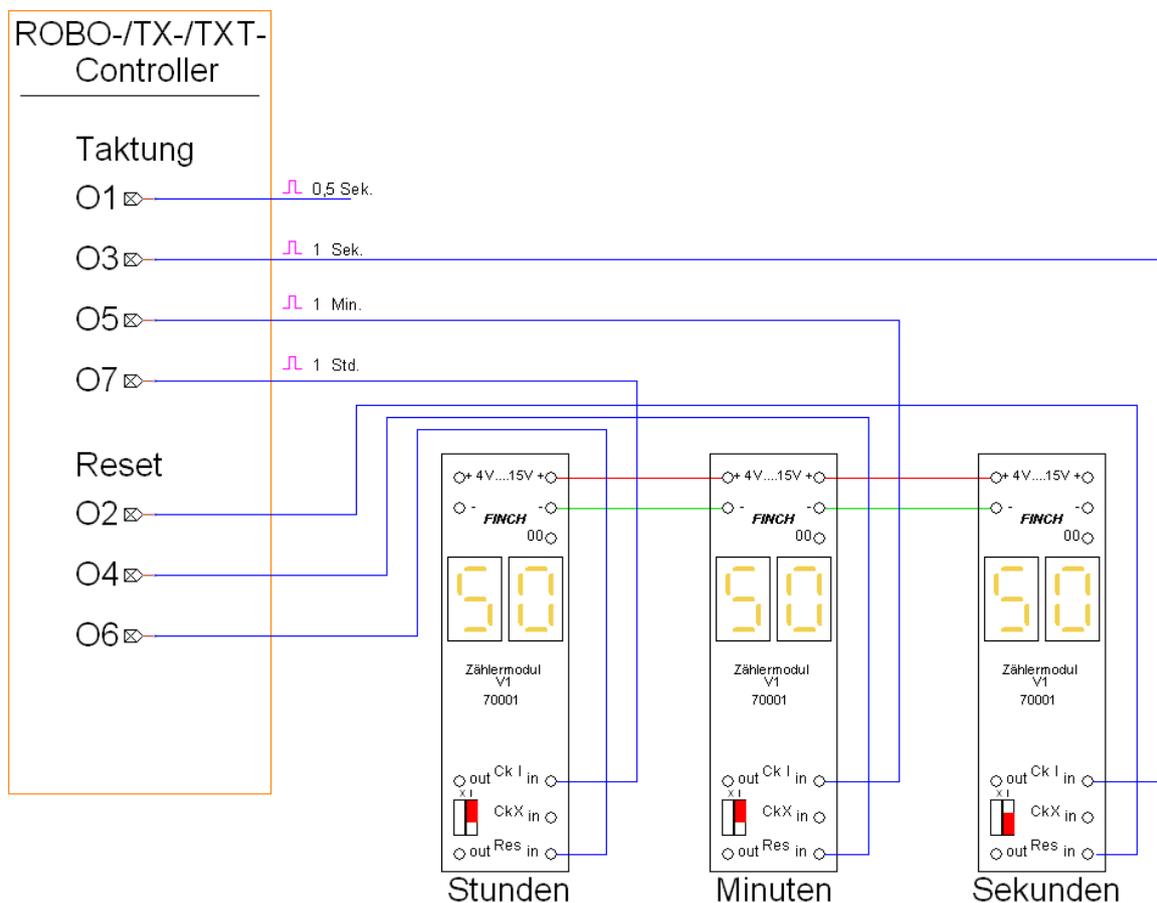


Abb. 11: Verbindung der Digital Uhr mit dem Controller

- 1 = Sekunden-Zähler
- 2 = Minuten-Zähler
- 4 = Stunden-Zähler

Durch Addition der Werte können auch mehrere Zähler mit einem Programmaufruf zurückgesetzt werden:

- 3 = Sekunden- und Minutenzähler
- 5 = Sekunden- und Stundenzähler
- 6 = Minuten- und Stundenzähler
- 7 = alle Zähler

Das Unterprogramm wird ohne Prüfung auf unzulässige Werte (kleiner als 0 oder größer als 7) wird ausgeführt.

Im Unterprogramm *Setzwert* wird eine vorgegebene Zeit auf der Uhr eingestellt. Die Zähler werden entsprechend den Werten der Variablen *Sekunde*, *Minute* und *Stunde* hochgezählt.

Bei Stunden und Minuten ist entsprechend der Bauanleitung der Jumper auf den Zählermodulen auf der Position „A“ (analog) einzustellen. In diesem Fall

werden die Zählimpulse über ein internes MF geleitet, dass eine Haltezeit von 300 ms hat (zum Abfangen von Prellimpulsen). Aus diesem Grund ist ein Zeitglied mit einer relativ langen Wartezeit von 0,3 Sekunden eingestellt, bevor das Programm weiterlaufen kann.

Nur beim Sekundenzähler steckt der Jumper auf „D“ und hier kann die minimale Verzögerung eingestellt werden. Wichtig ist auch die Zeitverzögerung nach dem Ausschalten des Signals, damit die Elektronik der Endstufe vom Controller genug Zeit hat, die Umschaltung auszuführen.

Die globalen Variablen *Stunde*, *Minute*, *Sekunde* werden auch ohne Prüfung auf unzulässige Werte (kleiner als 0 oder größer als 59 bzw. 23) verwendet.

### Modell „Jukebox“ mit Steuerung über einen Controller

Grundlage für die Jukebox bildet das Pianophone. Dessen Schaltung muss mit einem

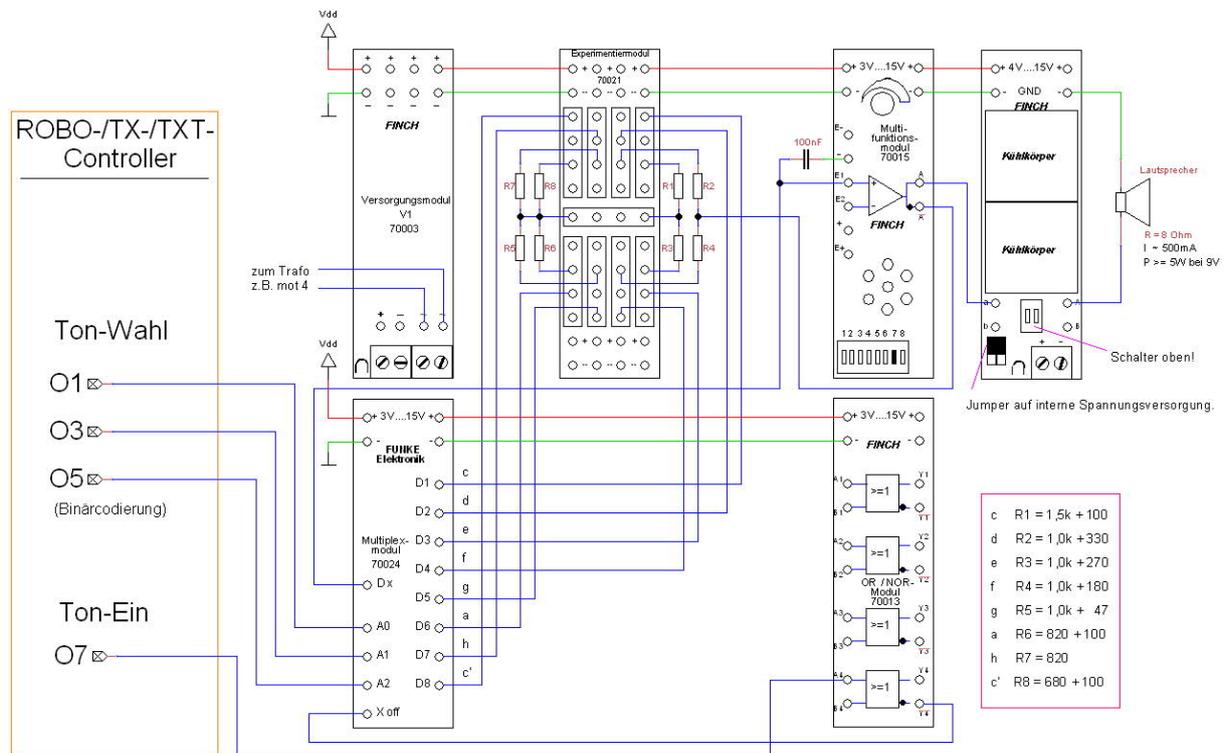


Abb. 12: Erweiterter Schaltplan vom Pianophone zur Jukebox durch Hinzufügen eines Multiplexmoduls 70024 und eines Inverters (Ersatz durch ein ODER-Gatter 70013).

Multiplexmodul 70024 (Abb. 12) erweitert werden, damit die Steuerung über einen Controller möglich wird. Dies ist auch ein sehr schönes Anwendungsbeispiel für den Einsatz eines Multiplexmodules.

Das gesamte Programm findet Ihr als PDF-Darstellung und als fertiges Programm (.RPP-Datei) auf der [www.ftcommunity.de](http://www.ftcommunity.de) im Abschnitt ft:pedia unter der Rubrik „Ergänzende Downloads“.

Mir persönlich fehlt der Bezug zur Musiklehre bzw. dem Spielen von Noten, so dass ich kein fertiges Musikbeispiel präsentieren kann. Zur Demonstration der Jukebox werde ich mich daher auf das Abspielen der Tonleiter beschränken müssen.

Für das Spielen einer Note sind die Tonhöhe und die Tondauer maßgebend. Auf einem Instrument können mehrere Töne gleichzeitig angestimmt werden, aber in der vorliegenden Modellversion von der Jukebox ist dies nicht möglich. Dazu müssten mehrere Tongeneratoren aufgebaut werden.

Als Vorbereitung für den Programmlauf muss eine Liste erstellt werden. Diese Listenvariable muss die Reihenfolge der abzuspielenden Noten enthalten (Tonhöhe, Tondauer).

Zahlenwert	Ton
0	c
1	d
2	e
3	f
4	g
5	a
6	h
7	c'

Tab. 3: Tonhöhen je nach Zahlenwert

Ein Ton dauert mindestens 10 ms (definiert durch den Zeitbaustein im Programm). Der Zahlenwert gibt die Anzahl der Durchläufe

in der Unterschleife an. Ist der Wert z. B. 3, beträgt die Tondauer 30 ms. Im Zeitbaustein der Unterschleife kann auch eine anderer Zeitparameter für die 10 ms eingesetzt werden.

Für die zu spielende Melodie wird dann eine Liste mit Tonhöhen- und Tondauer-Werten erstellt:

	A	B
1	Höhe	Dauer
2	7	1
3	6	2
4	5	2
5	4	3
6	3	3
7	2	4
8	1	4
9	-1	0
10		

Tab. 4: Listenvariable als CSV-Datei

Die Listenvariable kann mit Hilfe eines Tabellen-Kalkulationsprogramms erstellt und anschließend als Datei im CSV-Format gespeichert werden. Oder man erstellt eine Werteliste im Editor und legt die Datei mit der Endung .CSV an (Abb. 13).

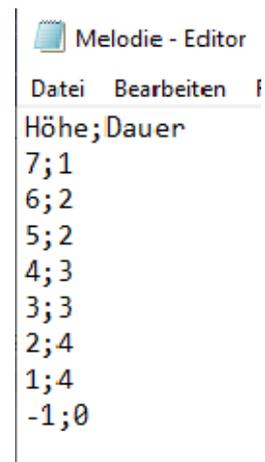


Abb. 13: Listenvariable im Editor erstellt

Bei einem Editor muss nur darauf geachtet werden, dass die Werte mit einem Trennzeichen (Komma oder Semikolon) separiert werden. In meinem Beispiel habe

ich die Wertetabelle für die Listenvariable *Melodie.CSV* benannt.

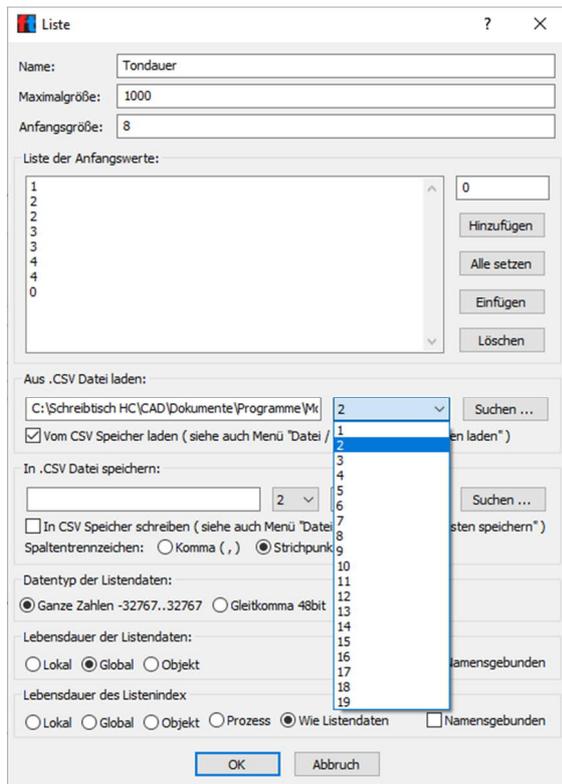


Abb. 14: Eigenschaften der Listenvariable „Tondauer“

Im Programm muss unter den Eigenschaften der Listenvariable die CSV-Datei hinterlegt werden (Rechtsklick auf die Liste – Abb. 14). Unter *Aus .CSV Datei laden:* wird die Datei für das abzuspielende Lied hinterlegt. Ebenso ist bei den anderen Listenvariablen die Datei zu hinterlegen; lediglich die Spalte ist bei allen Listenvariablen unterschiedlich auszuwählen: Tonhöhe aus Spalte 1, Tondauer aus Spalte 2.

### Programmablauf

Durch den Inverter am Ausgang O7 ist die Ton-Wiedergabe nach dem Einschalten des Controllers erst einmal deaktiviert. Zu Beginn des Programmes wird nach einer Zeitverzögerung von 2 Sekunden der Anfang des Lieds eingeleitet (das Zeitglied kann auch entfallen).

Das Programm besteht aus einer Hauptschleife und einer untergeordneten Schleife. Die Hauptschleife ist ein Zähler, der solange aufwärts zählt, bis ein negativer Wert aus der Liste *Tonhöhe* entnommen wird, womit das Ende eines Liedes signalisiert wird. Die Unterschleife wird solange durchlaufen, wie der Ton gehalten werden soll bzw. durch die Variable *Tondauer* vorgegeben wird. Zum Abschluss der Unterschleife wird die Ton-Wiedergabe abgeschaltet.

Für das Spielen des entsprechenden Tons muss der Schalter vom Multiplexmodul in die richtige Position gebracht werden. Die Dezimalzahl aus der Liste *Tonhöhe* wird dazu über das Unterprogramm *Binärcodierung* in eine Binärzahl umgewandelt, die den Multiplexschalter in die richtige Position stellt.

Damit es bei der Einstellung der Binärzahl des nächsten Tones nicht zu falschen Zwischentönen kommt, ist es wichtig, die Ton-Wiedergabe nach dem Abspielen des gerade gespielten Tones kurz zu unterbrechen.

Der an das Unterprogramm *Binärcodierung* übergebene Zahlenwert wird in eine Binärzahl umgewandelt. Dieses Programm kann nur 3 Bit große Zahlen (Dezimal 0-7) in eine Binärzahl umwandeln, aber das Programm kann leicht auf jede beliebige Anzahl von Bits erweitert werden. Eine Überprüfung des Zahlenwertes auf Gültigkeit wird hier nicht vorgenommen.

Die Ausgabe der Bit-Zahl erfolgt über die drei Ausgänge O1, O3 und O5, wobei O1 den niedrigsten Bit-Wert (A0) darstellt und O5 den höchsten Bit-Wert (A2).

### Anmerkung

Beim Multiplexmodul handelt es sich im Gegensatz zu einem mechanischen Relais um einen Halbleiterschalter. Das heißt, die Trennung der Kontakte wird nicht durch eine physikalische Trennung realisiert, sondern die Verbindung wird hochohmig

(im M $\Omega$ -Bereich). Beim Aufbau der Jukebox ist daher ein geringfügiger Unterschied in der Tonlage festzustellen, wenn die Verbindung über das Multiplexmodul erfolgt, weil auch der leitende Weg durch den Halbleiterschalter einige wenige Ohm aufweist und somit zur Änderung der Tonlage führt.

### Modell „Demo“ mit Steuerung über einen Controller

Dieses Modell verwende ich für Vorführzwecke auf Ausstellungen und Conventions. Es zeigt die Anwendungsmöglichkeiten für die Steuer- und Meldemodule. Darüber hinaus wird auch die Verwendungsmöglichkeit des Inverter-Plus-Modules demonstriert. Das entsprechende Schaltbild zu dem Modell findet Ihr in Abb. 15.

Das gesamte Programm (Hauptprogramm mit Unterprogrammen) findet Ihr als PDF-Darstellung und als fertiges Programm (.RPP-Datei) auf der [www.ftcommunity.de](http://www.ftcommunity.de) im Abschnitt ft:pedia unter der Rubrik „Ergänzende Downloads“.

Das Hauptprogramm unterteilt sich in zwei Prozesse, wobei der eine lediglich als Nebenprozess angelegt wurde, um dem Betrachter zu signalisieren, in welchem Abschnitt sich das Programm gerade befindet. Der Hauptprozess besteht aus drei Teilen:

1. Auslesen der Informationen vom Meldemodul 70017 und speichern in einer Listenvariable.
2. Die Informationen aus der Liste in das Steuermodul 70016 schieben/ausgeben.
3. Nach der Übertragung die einzelnen Speicher des Meldemoduls 70017 gezielt löschen.

Wird einer der drei Abschnitte im Programm erreicht, werden die Variablen *Lesen*, *Schieben* und *Reset* gesetzt und beim Verlassen des Abschnittes zurückgesetzt. Der Nebenprozess wertet diese Variablen aus und signalisiert dem Beobachter über zwei Lampen (gesteuert über die Ausgänge O2 und O4), welcher Abschnitt gerade bearbeitet wird.

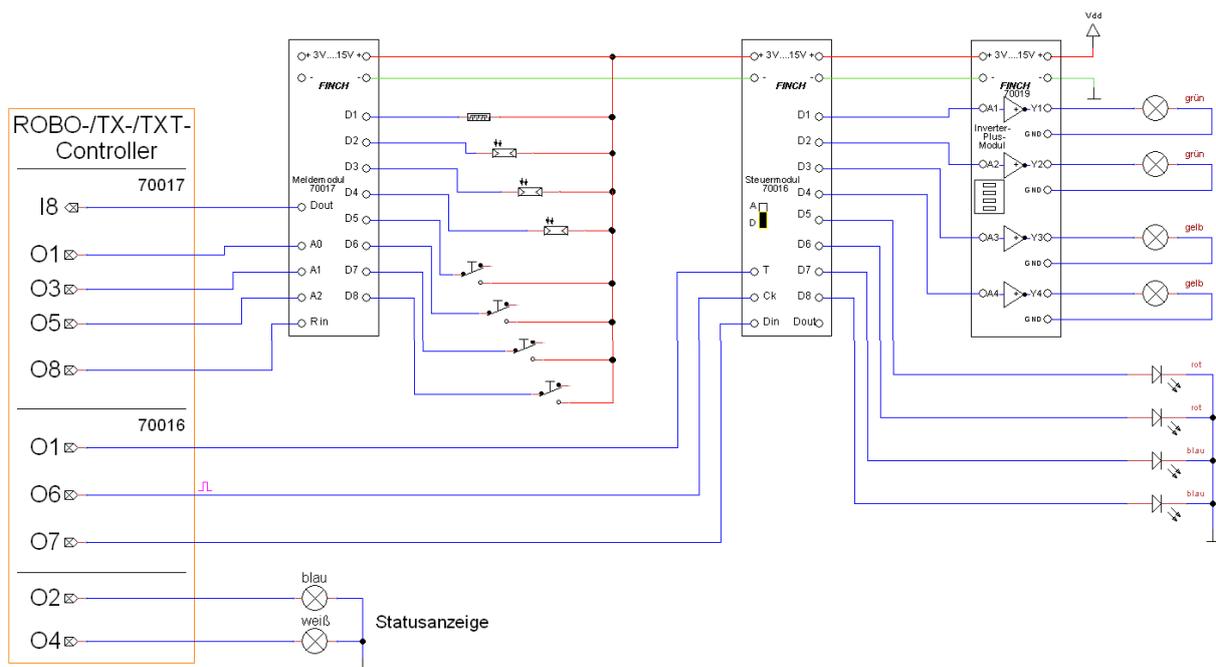


Abb. 15: Schaltplan Modells „Demo“ – Steuermodul 70016 und Meldemodul 70017 am Controller

**Datenerfassung und Vorbereitung** (Demonstriert die Funktionsweise des Meldemoduls): Vor Beginn des Programmablaufes wird über verschiedene Sensoren, die direkt an die Eingänge des Meldemoduls angeschlossen sind, ein zufälliges Datenmuster im Meldemodul erzeugt.

**Programmlauf:** Über eine Zählschleife wird der Eingang I8 insgesamt achtmal abgefragt und der Wert in die Liste *Schalterdaten* eingetragen.

Die Auswahl der verschiedenen Speicher des Meldemoduls erfolgt durch einen Binärwert, der über die Ausgänge O1, O3 und O5 bereitgestellt wird. Hier wurde der Zählwert des Schleifenzählers mit Hilfe des Unterprogramms *Binärcodierung* in einen Binärcode umgewandelt und an die Ausgänge O1, O3 und O5 übergeben. Sehr wichtig ist die Wartezeit nach dem Einstellen des Binärcodes! Die Endstufen von den Ausgängen des Controllers müssen genug Zeit bekommen, die neuen Einstellungen zu übernehmen. Die digitale Technik (CPU des Controllers und die Elektronikmodule) sind sehr schnell und somit muss das Programm immer wieder ausgebremst werden, damit alle Signale korrekt an den Ausgängen des Controllers zur Verfügung stehen.

Zum Schluss dieses Abschnittes wird der Binäre-Ausgang (O1, O3, O5) auf null gesetzt und das Steuermodul für den nächsten Schritt vorbereitet, indem über das Unterprogramm *Löschen* alle internen Speicher des Steuermoduls gelöscht werden.

**Daten schieben** (Demonstriert die Funktion des Steuermoduls): Die durch das Meldemodul aufgenommenen Daten, die in der Liste *Schalterdaten* abgelegt wurden, werden in diesem Abschnitt des Programmes wieder der Liste entnommen und über das Steuermodul ausgegeben.

Über eine Zählschleife – diesmal Rückwärts – werden die Daten aus der Liste

*Schalterdaten* ausgelesen und dem Steuermodul am Eingang D<sub>IN</sub> (O7) zur Verfügung gestellt. Danach werden die Daten über das Unterprogramm *Übergabe* in das Schieberegister geschoben. Das kleine Unterprogramm *Übergabe* wurde nur deshalb angelegt, weil dieser Teil auch zum Zurücksetzen des Steuermoduls benötigt wird und somit die Redundanz vermindert und die Programmübersicht verbessert wird.

Damit der Zuschauer mehr Transparenz über die Abläufe im Programm erhält, habe ich mich entschieden, nach jedem Bit, das in das Schieberegister geschoben wird, dies gleich für die Zuschauer sichtbar zu machen. Allerdings gab es keinen freien Steuerausgang mehr, so dass ich mich für eine Doppelbelegung des Steuerausgangs O1 entschieden habe, der in diesem Teil des Programmabschnitts nicht verwendet wird. Sobald der Ausgang O1, der mit dem Transfer-Eingang des Steuermoduls verbunden ist, auf high geht, werden die Daten im Schieberegister an die Ausgang-Flip-Flops weitergereicht und über die LEDs und Lampen – die über ein Inverter-Plus-Modul 70019 angeschlossen sind – sichtbar gemacht.

Sowohl in diesem als auch im nächsten Abschnitt sind wieder Wartezeiten in den Programmablauf eingefügt worden. Es hat sich in der Praxis bewährt, eine Wartezeit von mindestens 10 ms zu verwenden, damit die Endstufe des Controllers sicher die eingestellten Werte übernommen hat.

**Daten löschen** (Demonstriert die Funktionsweise des Meldemoduls): Nachdem sämtliche Daten ausgelesen und für den Zuschauer sichtbar gemacht wurden, wird eine Pause von 10 Sekunden eingelegt. Diese Pause soll dem Zuschauer die Möglichkeit einräumen, die Daten vom Meldemodul mit den Daten am Steuermodul zu vergleichen.

Im letzten Abschnitt wird noch der Beweis angetreten, dass sämtliche Speicher vom Meldemodul einzeln und gezielt gelöscht werden können. Dafür wurde eine weitere Listenvariable *Resetverlauf* angelegt. Die Listenvariable kann mit Dezimalzahlen 1-8 in beliebiger Reihenfolge bestückt werden. Die Zahlen werden über eine Schleife nacheinander ausgelesen und in Binärcode umgewandelt, womit der zu löschende Speicher ausgewählt wird. Nach der Auswahl wird der Eingang  $R_{IN}$  (O8) am Meldemodul gesetzt (high) und der entsprechende Speicher bzw. das Flip-Flop zurückgesetzt.

Der an das Unterprogramm *Binärcodierung* übergebene Zahlenwert wird in eine Binärzahl umgewandelt. Dieses Programm kann nur 3 Bit große Zahlen (Dezimal 0-7) in eine Binärzahl umwandeln, aber das Programm kann leicht auf jede beliebige Anzahl von Bits erweitert werden. Eine Überprüfung des Zahlenwertes auf Sinnhaftigkeit wird hier nicht vorgenommen.

Die Ausgabe der Bit-Zahl erfolgt über die drei Ausgänge O1, O3 und O5, wobei O1 den niedrigsten Bit-Wert (A0) darstellt und O5 den höchsten Bit-Wert (A2).

Mit dem Unterprogramm *Löschen* wird das Schieberegister vom Steuermodul initialisiert, in dem alle Speicher auf null gesetzt bzw. gelöscht werden. Um das zu erreichen, müssen acht Takte an den Clock- oder Takt-Eingang (O6) des Schieberegisters gesendet und der Dateneingang  $D_{IN}$  (O7) auf low gesetzt werden.

Aufgabe des Programmteils *Übergabe* ist es, die am Eingang  $D_{IN}$  (O7) befindliche Information ins Schieberegister zu transportieren (1 Bit schieben). Der Ausgang O6 steuert den Clock- bzw. Takt-Eingang am Steuermodul.

Auf youtube habe ich ein [kleines Video](#) hinterlegt. Dort könnt Ihr das Modell und die Funktionsweise sehen.

## Schlusswort

Mit diesem Teil sind alle bisher erschienenen Elektronikmodule einmal beschrieben worden, teilweise sogar mit praktischen Beispielen oder Modellen hinterlegt.

In dem von mir in Vorbereitung befindlichen Dokument „Grundlagen der Elektronikmodule“ (70099), in dem auch Schaltungsbeispiele enthalten sind, trage ich noch einmal alle technischen Daten über die Elektronikmodule zusammen. Das Dokument kann allerdings im Moment nur direkt bei mir per E-Mail angefordert werden und befindet sich noch in der Entstehungsphase.

Im nächsten Jahr wird es eine Fortsetzung geben mit der Vorstellung von wenigsten zwei neuen Elektronikmodulen, die bis dahin fertiggestellt werden sollen.

Für Fragen, weiteren Ausführungen oder Anregungen aller Art stehe ich gern zur Verfügung ([elektronik@funke4you.de](mailto:elektronik@funke4you.de)).

Fortsetzung folgt.

## Quellen

[1] Wikipedia: [Morsecode](#).