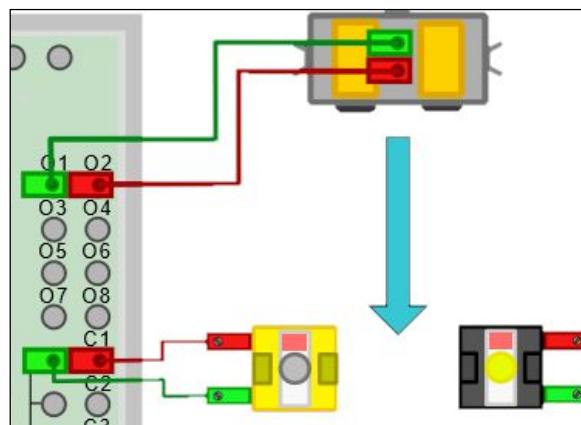


ft:pedia

Heft 4/2018



```
*SimpleDemo.java x
package test;

import ftcomputing.*;

public class SimpleDemo {

    public static void main(String[] args) {
        JavaFish ft = new JavaFish();
        R = ft.openInterface("LPT");
    }
}
```



Herausgegeben von
Dirk Fox und Stefan Falk

ISSN 2192-5879

Editorial

We Made It

Pünktlich zum 20. Geburtstag des Grundbausteins machte fischertechnik 1985 einen großen, zukunftsweisenden und damals sicherlich gewagten Schritt: Mit Interfaces für den Commodore VC20/C64, den Apple II, die Schneider CPC-Familie und den Acorn öffnete sich das Baukastensystem der Welt der Mikrocontroller. Damit war fischertechnik anderem Technik-„Spielzeug“ Längen voraus und legte den Grundstein für die spätere „Controller-Familie“ und die grafische Programmiersprache ROBO Pro, deren Vorläufer als „Lucky Logic“ 1991 das Licht der Welt erblickte.

In den seitdem vergangenen über 30 Jahren hat sich die Informationstechnik rapide entwickelt. So steckt in den heutigen Controllern weit mehr Rechenleistung als damals im Host-PC. Mikrocontrollersteuerungen sind Standard in der Industrie, es haben sich einheitliche Schnittstellen und Protokolle (wie I²C und SPI) für die Anbindung von Sensoren etabliert, und Mikrocontroller wie der Arduino haben Millionen Menschen einen niedrigschwälligen Zugang zur Steuerungstechnik eröffnet.

Mit der Do-it-Yourself- oder „Maker“-Bewegung haben sich in den vergangenen Jahren aber auch die Rahmenbedingungen geändert: Open-Source-Projekte bieten transparente und günstige „Community“-Lösungen, die nicht nur bei der Aktualität, sondern oft auch bei Qualität und Service im Vergleich mit „Closed-Source“-Angeboten die Nase vorn haben. Auch fischertechnik kann da immer schlechter mithalten: Die Controller werden schon heute extern eingekauft und der Preisunterschied

Dirk Fox, Stefan Falk

zu einer Open-Source-Lösung ist immer schwieriger zu rechtfertigen, denn dank 3D-Druck verliert auch das fischertechnik-Gehäuse an Exklusivität.

Zu welchen Leistungen eine hoch motivierte, begeisterte und vernetzte Community im Stande ist, hat die ftc schon bei der Entwicklung der Community-Firmware für den TXT und den [TX-Pi](#) gezeigt. Inzwischen ist daraus eine ganze Familie an Entwicklungen geworden:

- mit [startIDE](#) kann der TXT über den Touch-Screen programmiert werden,
- [Blockly](#) ermöglicht die Programmierung des TXT über eine Webschnittstelle im Stil von Scratch,
- Plug-and-Play-Adapter wie der [ftExtender](#) kompensieren die Schwäche des TXT bei der I²C-Anbindung,
- der [ftDuino](#) verheiratet die fischertechnik- mit der Arduino-Welt – und damit zugleich mit einem riesigen Repertoire an Software, Sprachen und Sensoren,
- und der ftPwrDrive-Controller erweitert die Controller um eine Ansteuerung von Schritt- und Servomotoren.

Eine große Chance für fischertechnik: Die mit dem eigenen Bedarf vertraute Fan-Community übernimmt das Markt- und Entwicklungsrisiko – und jedes erfolgreiche Tool wertet die bunten Klötzchen auf.

Schöne Weihnachten, Euer ft:pedia-Team

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

Inhalt

We Made It.....	2
Mini-Modelle (21): Kleine Radlader mit Knicklenkung.....	5
Fußballflipper: Schuss und – Tooor!	10
Schwungradauto	13
Die Kettenuhr	22
Kirchturmuhruhr mit Huygensschem Aufzug	26
Useless Machine – zwecklose Maschine	33
Die Welt der fischertechnik-Winkelbausteine (Teil 2).....	38
Einbindung des TXT- bzw. TX-Pi-Touchscreens in startIDE	49
startIDE (5): Temperaturmessung an der Community-Akku-Box.....	53
Kurz gemessen – Experimente im Mikrosekundenbereich mit dem ftDuino.....	57
Das Universal-Interface am LPT-Port unter Windows 10/64 bit	64
Der ftPwrDrive-Controller für Schrittmotoren und Servos – Teil 1	67

Termine

Was?	Wann?	Wo?
Modellschau Münster	20.01.2019	Münster-Hiltrup
Clubdag in Veghel	02.02.2019	Veghel
fischertechnik Nordconvention 2019	27.04.2019	Wedemark – Mellendorf
FanClubTag 2019	13.07.2019	Waldachtal/ Tumlingen

Impressum

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

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

Autoren: Bernd Bekehermes, Christian Bergschneider,
Matthias Dettmer, Stefan Falk, Dirk Fox, Stefan Fuss,
Björn Gundermann, Peter Habermehl, Till Harbaum,
Helmut Jawtusch, Christian Lauff, Thomas Püttmann,
Rüdiger Riedel, Dirk Wölffel.

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

Modell

Mini-Modelle (21): Kleine Radlader mit Knicklenkung

Rüdiger Riedel

Drei Minimodelle, in Englisch heißen sie Wheel Loader, natürlich mit beweglichen Schaufeln!

Mini-Radlader

Variante 1

Radlader mit Knicklenkung habe ich in zwei Ausführungen gefunden: Fahrerkabine auf dem Vorderwagen wie hier im Modell dargestellt (selten), oder Fahrerkabine auf dem Hinterwagen, siehe Variante 2.

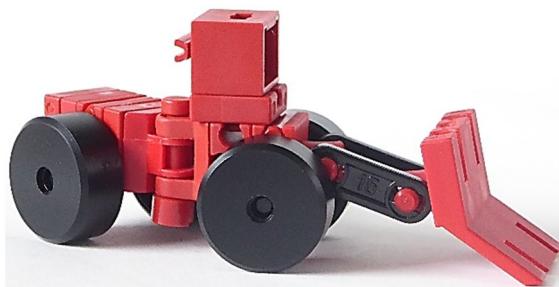


Abb. 1: Der Mini-Radlader

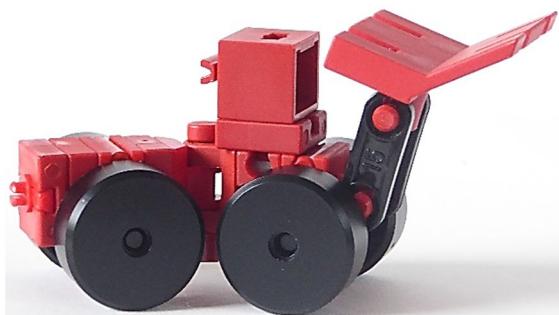


Abb. 2: Hoch die Schaufel

Dieses Modell folgt dem Vorbild Zettelmeyer ZL 2002 (siehe Abb. 6) [1].

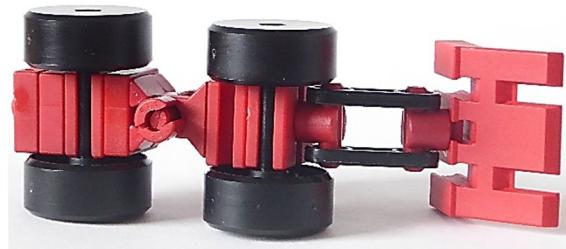


Abb. 3: Die Unterseite

Die Seilklemmstifte ([107356](#)) müssen mit etwas Kraft in die BS7,5 und die Klemmhülsen ([35980](#)) hinein gedrückt werden.

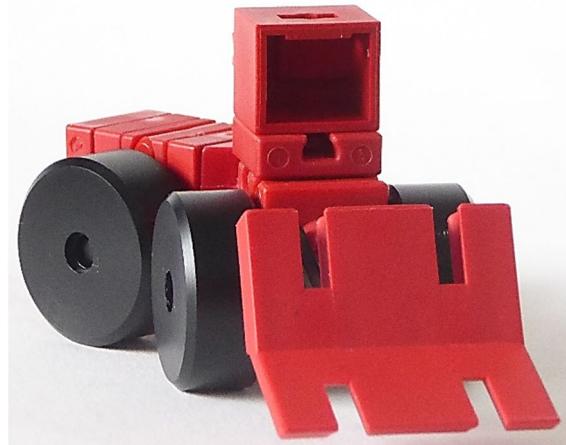


Abb. 4: Frontansicht

Die Schaufel bleibt in allen Zwischenstellungen stehen. Die V-Achse 20 ([31690](#)) im Gelenk dient nur der optischen Verschönerung.

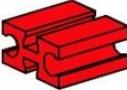
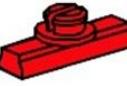
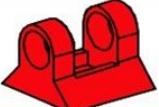
	37 468
3 Stück	
	36 914
2 Stück	
	36 574
4 Stück	
	31 690
1 Stück	
	37 237
2 Stück	
	31 848
1 Stück	
	38 411
1 Stück	
	107 356
4 Stück	
	31 426
1 Stück	
	37 679
1 Stück	
	37 238
2 Stück	
	35 980
2 Stück	
	38 413
2 Stück	
	31 436
1 Stück	
	36 973
1 Stück	
	31 982
1 Stück	

Abb. 5: Einzelteile des Mini-Radladers I



Abb. 6: Vorlage für den Mini-Radlader Variante 1 [1]

Variante 2

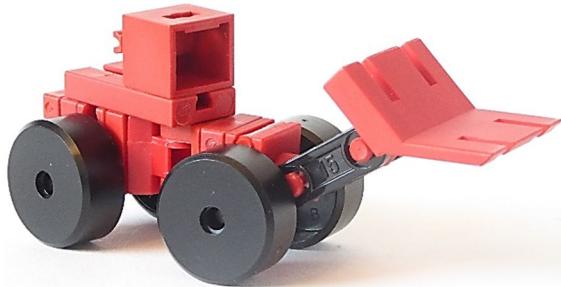


Abb. 7: Radlader mit Führerhaus hinten

Das Führerhaus rutscht jetzt auf den Hinterwagen und entspricht damit dem Vorbild

von Volvo (siehe Abb. 11). Der Umbau ist schnell erledigt: Es entfallen ein Seilklemmstift ([107356](#)), die V-Achse 20 ([31690](#)), die Klemmbuchse 5 ([37679](#)) und ein Baustein 5 ([37237](#)).

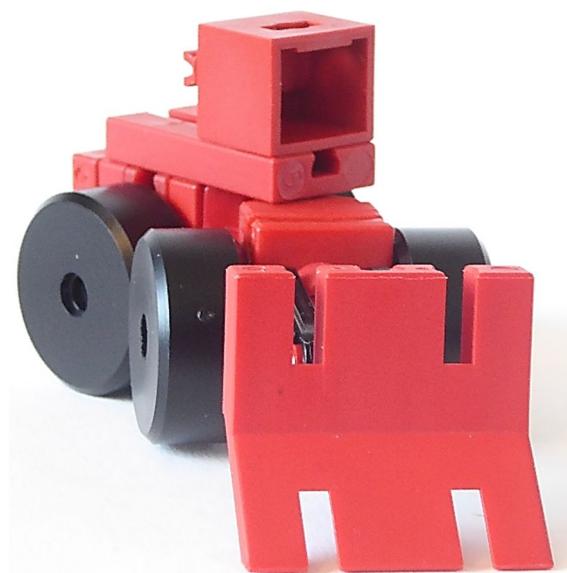


Abb. 8: Lenkung nach rechts

Hinzu kommt ein Baustein 5 15x30 ([35049](#)), auf dem jetzt das Führerhaus sitzt.



Abb. 9: Ansicht von unten

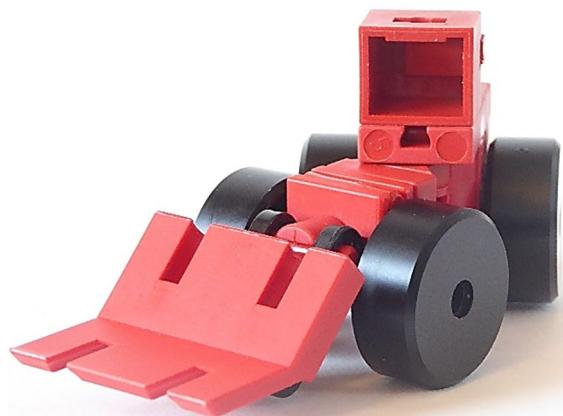


Abb. 10: Blick in die Fahrerkabine

Beim Lenken schaut der Fahrer hier nicht in Fahrt- und Schaufelrichtung.



Abb. 11: Wheel loader Volvo L120E [2]

Mikro-Radlader

Klein aber mit Gummireifen! Als Vorlage habe ich den Wheel Loader *Hitachi ZW 75* genommen (siehe Abb. 17).



Abb. 12: Der Mikro-Radlader

Die Schaufel kann gehoben und gesenkt werden.



Abb. 13: Schaufel oben



Abb. 14: Ansicht von unten

Der Winkelstein 15° ist erforderlich, damit die Schaufel bis auf den Boden kommt.



Abb. 15: Frontansicht



Abb. 17: Vorlage „Mikro-Radlader“ [3]

37 468 2 Stück	31 994 2 Stück	31 848 1 Stück
38 258 4 Stück	31 426 1 Stück	31 436 1 Stück
Ø8 35 677 4 Stück	35 063 2 Stück	36 973 1 Stück
37 237 3 Stück	37 232 2 Stück	31 982 2 Stück
35 738 2 Stück	36 327 1 Stück	31 981 1 Stück

Abb. 16: Einzelteile des Mikro-Radladers

Radlader mit Knicklenkung

Es gibt auch Radlader mit Allradlenkung und gemischte Varianten, doch die betrachten wir hier nicht.

Ein großer Vorteil der Knicklenkung: Vorder- und Hinterräder laufen in der gleichen Spur. Das spart gerade bei Baufahrzeugen auf weichem Untergrund erheblich Energie. Die Lenkgeometrie bleibt gleich bei Vorwärts- und Rückwärtsfahrt.

Wird die Knicklenkung im Stand betätigt, ergibt sich eine seitliche Verschwenkung der Schaufel, das hilft beim exakten Beladen.



Abb. 18: Die Kipplast verringert sich mit dem Lenkeinschlag

Ein wichtiger Kennwert der Radlader ist die Kipplast. Dies ist nicht die Nutzlast, sondern sie gibt an, bei welcher Last auf der Schaufel der Hinterwagen vom Boden

abhebt, also welches Gewicht die Schwinge maximal anheben kann. Bei knickgelenkten Ladern ist der Wert bei eingeknickter Lenkung entscheidend, er ist geringer als bei gerader Ausrichtung.

Zur Veranschaulichung habe ich in Abb. 18 die Aufsetzpunkte der Räder jeder Seite miteinander verbunden und die Mittellinie dazwischen eingezeichnet. Man sieht das seitliche Ausschwenken der Ladeschaufel, wodurch deren Schwerpunkt von der Mittellinie zur Seite wandert.

Ein Lenkeinschlag von z. B. 40° kann eine um 12 % geringere Kipplast bedeuten.

Die Nutzlast darf nach EN 474-3 und ISO 14397-1 in der Ladeschaufel von Radladern 50 % der Kipplast nicht überschreiten, wenn der Radlader maximal eingelenkt ist.

Beim Antrieb wird der Dieselmotor gerne durch ein hydrostatisches Getriebe ergänzt, doch dessen Funktion muss einem anderen Artikel vorbehalten bleiben.

Quellen

- [1] Mit freundlicher Genehmigung von Alexis H.; [Zettelmeyer ZL 2002](#).
- [2] Wikimedia: [Bild eines Wheel Loaders](#).
- [3] Wikimedia: [Bild eines Hitachi ZW75](#).

Modell

Fußballflipper: Schuss und – Tooor!

Thomas Püttmann

Flipper sind klasse. Aber es fehlt der letzte Kick. Diesen Kick liefert – natürlich der Fußball.

Im Kindergarten unserer zwei ältesten Kinder habe ich vor 17 Jahren zum ersten Mal einen Fußballflipper gesehen (Abb. 1). Gleich zum nächsten Anlass habe ich ein Exemplar für die Familie gekauft – eine Anschaffung, die sich voll und ganz gelohnt hat. Das schnelle Reaktionsspiel begeistert nach wie vor Groß und Klein.

Zwei Spieler sitzen einander gegenüber und versuchen, mit ihren Flipperarmen eine Stahlkugel durch die gegnerischen Flipperarme ins gegenüberliegende Tor zu schießen. Der Lauf der Kugel wird dabei meist durch den Zusammenstoß mit den eigenen und gegnerischen Spielern

verändert. Von direkten Treffern über Zickzackschüsse bis zu Eigentoren ist alles möglich. Derjenige Spieler, der als erstes fünf Tore erzielt hat, gewinnt das Spiel.

Obwohl schon viele Flipper aus fischertechnik gebaut wurden, habe ich darunter noch keinen Fußballflipper gesehen.

Das Ziel war natürlich, einen Flipper zu bauen, mit dem zu spielen so viel Spaß macht wie mit dem Original (Abb. 2). Er ist sehr reduziert und lässt sich noch in alle Richtungen ausbauen. Von einer Spielstandsanzeige über eine automatische Trefferanzeige bis zum Roboterspielpartner ist alles denkbar.



Abb. 1: Unser 17 Jahre alter und intensiv bespielter Fußballflipper



Abb. 2: Unser Fußballflipper aus fischertechnik

Jeder der Arme dreht sich um eine Rastachse mit Platte. Durch einen Baustein 5 bekommt diese Rastachse mit Platte einen ausreichenden Abstand zur Grundplatte. Der Arm ist dadurch so hoch, dass die Kugel beim Zusammenstoß nicht über ihn hinwegspringt.

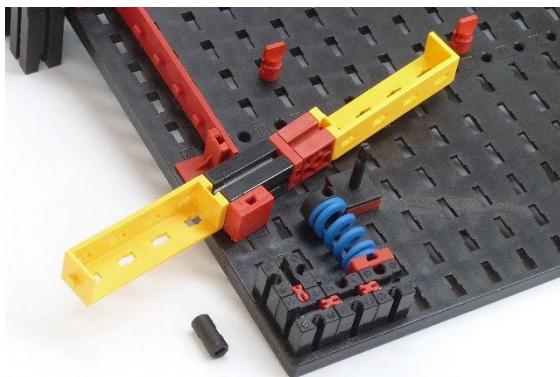


Abb. 3: Der Arm, seine Befestigung und Federung.

Die Arme besitzen einen Auflagepunkt außerhalb der Drehachse. Diese Auflagepunkte verhindern, dass die Flipperarme durch den Druck der Spielerhände in der

Mitte nach oben angehoben werden. Sie bestehen aus einem Baustein 5 und einer Bauplatte 15x15.

Zur Federung der Flipperarme kommen die Fahrwerksfedern 26 mit Boden zum Einsatz. Die Flipperarme lassen sich extrem schnell und trotzdem feinfühlig steuern.

Die Fahrwerksfedern sind jeweils über einen Baustein 5 und einen Baustein 15 an der Grundplatte befestigt. Um zu verhindern, dass sich der Baustein 15 bei härteren Schlägen löst, ist er mit drei weiteren Bausteinen 15 verbunden (Abb. 3). Gleichzeitig liefert dieser Verbund einen guten Ablageplatz für die Kugel.

Die beiden Grundplatten werden an jeder Seite durch zwei Winkelsteine 7,5°, zwei Bausteine 15, zwei Verbindungsstücke 15 und einen Baustein 7,5 verbunden (Abb. 4). Eine Bauplatte 5x15x30 mit einem Zapfen und Längsnut verhindert, dass sich die Steine bei intensiverem Spiel gegeneinander verschieben.



Abb. 4: Verbindung der beiden Grundplatten

An einer dieser Verbindungsbrücken befindet sich die Balleingabe. Sie besteht aus einem Winkelstein 30° und einer weiteren Bauplatte $5 \times 15 \times 30$ (Abb. 5).

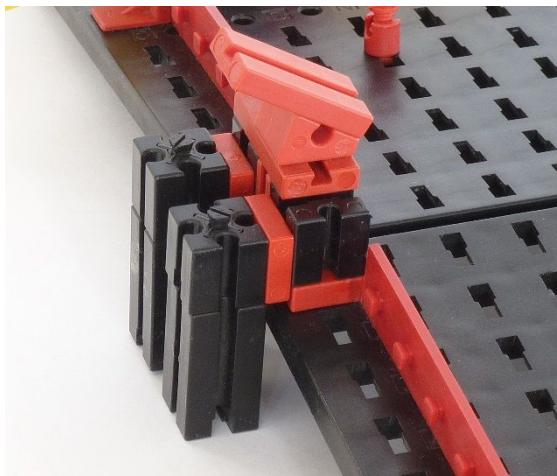


Abb. 5: Balleingabe und seitliche Stützen

Außerdem sind an den Verbindungsbrücken auch die seitlichen Stützen unseres Flippers befestigt. Sie bestehen aus einem Baustein 5 mit zwei Zapfen, einem Baustein 15 und einem Baustein 30.

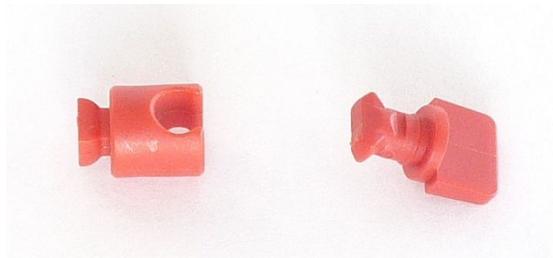


Abb. 6: Der Aufbau der Spieler

Die Spieler bestehen jeweils aus einer Klemmhülse und einem S-Riegel (Abb. 6). Es macht natürlich Spaß, die Anzahl der Spieler und ihre Position auf dem Feld zu verändern und zu beobachten, wie sich das Spiel dabei verändert.

Viel Freude beim Nachbauen, Weiterentwickeln und vor allem beim Spielen!

Modell

Schwungradauto

Rüdiger Riedel

Zurück zu den Wurzeln der fahrenden Spielzeugautos: Da gab es den Federwerk-Antrieb und den Schwungrad-Antrieb. Das Federwerk wurde mit einem Schlüssel aufgezogen (neuere Versionen arbeiten mit einem Ratschen-Antrieb und das Aufziehen geschieht durch Vor- und Rückwärtsziehen); das Schwungrad wurde durch Vorwärtsbewegung auf Drehzahl gebracht.

Vorbemerkung

Denke ich an ein fischertechnik-Schwungrad, fallen mir die Schwungscheibe 60·15 ([31718](#) oder [39006](#)) ein und das Speichenrad 90·15 ([19317](#) oder [36916](#)).

In meinen Versuchen funktionierte die Schwungscheibe wirklich gut (leider ist sie nur noch gebraucht erhältlich), dagegen war das Speichenrad eine Enttäuschung. Recht gut funktionierte auch die Drehscheibe 60·5,5 ([31019](#)). Grundsätzlich sollte sich erstens die Masse eines Schwungrades möglichst weit außen am Durchmesser befinden. Das ist bei den ersten beiden der Fall. Wegen der hohen Drehzahl spielt der Luftwiderstand eine Rolle und da hat die glatte Schwungscheibe einen großen Vorteil. Beim Speichenrad spürt man den Luftzug; es arbeitet wie ein Ventilator und hat deshalb große Luftriebungsverluste.

Zweitens ist die Masse des Schwungrades verantwortlich für die speicherbare Energie. Die Gewichte [4] sind:

- Schwungscheibe: 15 g
- Speichenrad: 8 g
- Drehscheibe 60·5,5: 9 g

Drittens steigt die gespeicherte Energie mit der Drehzahl.

Da haben wir aber auch den großen Schwachpunkt eines Schwungradautos:

Beim Start soll sich das Schwungrad möglichst schnell drehen, das Fahrzeug muss aber aus dem Stand heraus beschleunigt werden, also mit Drehzahl 0 beginnen. Technisch erreicht man das bei unseren Autos mit einer Kupplung, die am Anfang durchrutscht und langsam die hohe Drehzahl überträgt, „schleifende“ Kupplung genannt. Beim folgenden Modell ist der Übergang von der Antriebsachse auf die Räder gleichzeitig die Kupplung, die beim Start immer durchrutschen muss.

Jetzt wird gebaut

Ein Fahrzeug mit Gummantrieb wurde bereits vorgeschlagen [1] und auch Mausefallenautos [2, 3].

Unser Schwungradauto wird mit einem Seilstart ausgerüstet, das erfordert eine besondere Art der Achslagerung.



Abb. 1: Das Fahrzeug mit Schwungrad

Im Betrieb liegt die Achse der Schwungscheibe auf den Hinterrädern, beim

Anwerfen muss sie aber angehoben werden. Wir erreichen das durch Verwendung von Gelenkwürfel-Klauen 7,5 ([31436](#)). Diese haben einen Innendurchmesser von etwa 5,5 mm, also mehr als die 4 mm Achsdurchmesser. Die Schwungradachse treibt die 50 mm-Reifen an, die für die Fortbewegung sorgen.



Abb. 2: Trike von vorn

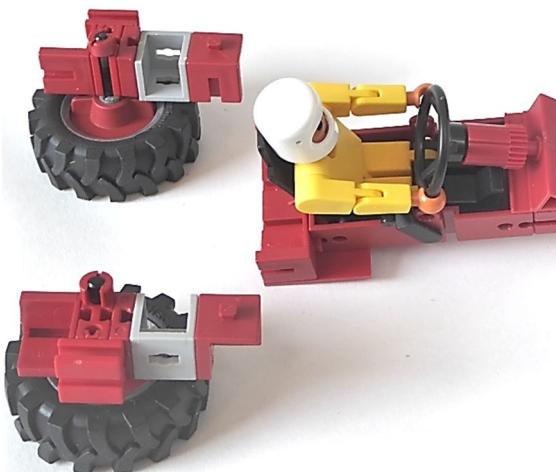


Abb. 3: Die Hinterräder

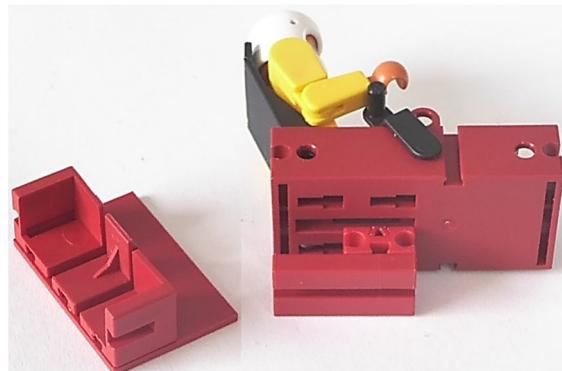


Abb. 4: Befestigung Fahrersitz und der Niederhalter

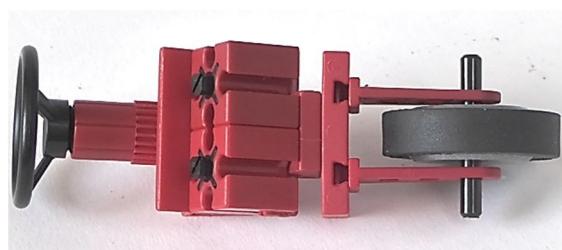


Abb. 5: Das Vorderrad von unten gesehen

Das Fahrzeug wird auf der Service-Station betriebsfertig gemacht. Dazu sitzen die grauen Winkelträger 15 des Fahrzeuges auf den beiden roten Winkelträgern 10 der Station.

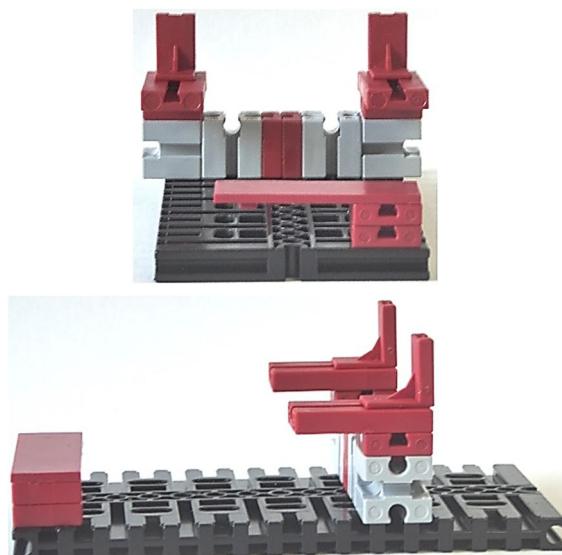


Abb. 6: Die Service-Station von Vorne und von der Seite

Anschließend wird die Service-Station zum Startplatz.

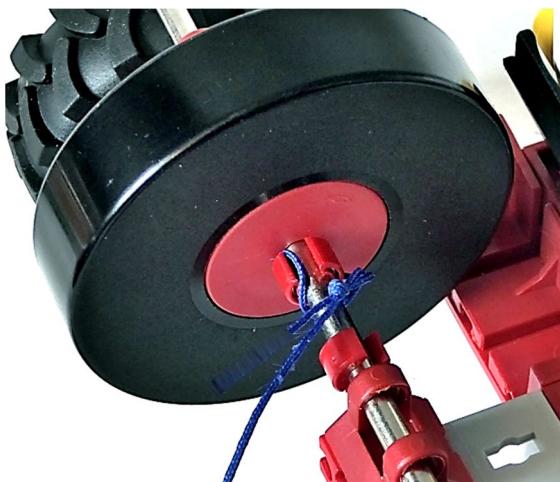


Abb. 7: Das Seil wird eingehängt

Das fischertechnik-Seil ist etwa 60 cm lang und an einem Ende zu einer Schlinge gebunden. Am anderen Ende wird gezogen, dafür verwende ich einen Kranhaken; das Seil wird darin mit einem Seilklemmstift 15 ([107356](#)) fixiert.

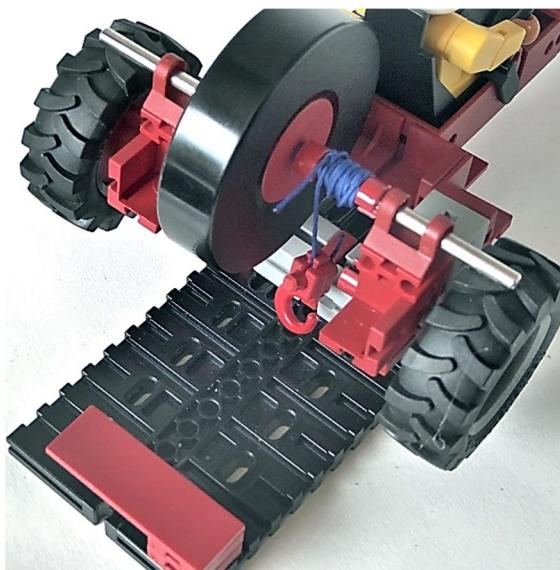


Abb. 8: Auf der Service-Station wurde das Seil aufgewickelt

Gemäß Abb. 7 wird die Seilschlinge in die Klemmbuchse eingelegt, dann wird das Seil auf die Achse gewickelt (Abb. 8).

Zum Start wird die Service-Station herumgedreht, sodass die beiden Bausteine 5 15·30 ([35049](#)) auf die beiden V-Bausteine 15 Eck ([38240](#)) am Ende des Fahrzeugs drücken. Außerdem soll der Baustein 5

15·30 unter dem Fahrzeug von der Bauplatte 15·45 2Z ([38242](#)) an der Service-Station niedergehalten werden.

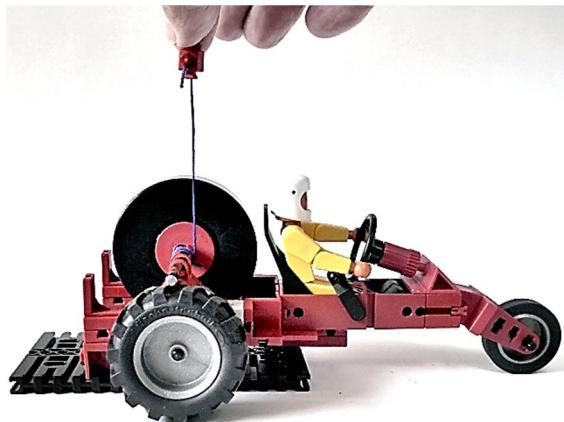


Abb. 9: Wir laden das Schwungrad auf

Nummer	Bezeichnung	Anzahl
31978	Figur-Sitzbaustein 30·60·15	1
31718	Schwungscheibe 60·15	1
31984	Figur-Schalensitz	1
106767	Traktor-Gummireifen 50	2
116913	Felge 30 chrom-farben	2
35033	Nabenmutter	2
37237	Baustein 5	2
38428	Baustein 5 15·30 3N	1
38260	S-Kupplung 22,5	2
31011	Winkelstein 30°	1
31994	Figur-Schalthebel 15	1
35806	Figur weiß + blau (schwarz)	1
31909	Figur-Integralhelm weiß	1
31031	Metallachse 110	1
38413	Kunststoffachse 30	1
38414	Kunststoffachse 40	2

Nummer	Bezeichnung	Anzahl
32230	Lenkradhalter R30	1
31916	Lenkrad2 Inbus	1
31915	Zangenmutter Z22 m0,5	1
116252	Baustein 15 rot	2
37468	Baustein 7,5 rot	5
146535	V-Rad 23·10 chrom-farben (silber)	1
34995	Gummireifen 32,5·8 schwarz	1
32330	Bauplatte 15·30·3,75 1N rot	1
38240	V-Baustein 15 Eck	6
38423	Winkelstein 10·15·15	1
31436	Gelenkwürfel-Klaue 7,5	2
31061	Verbinder 30	1
31982	Federnocken rot	2
36304	S-Winkelträger 15 2Z grau	2
31015	Flachnabe 25 komplett rot	1
37679	Klemmbuchse 5 rot	5

Tabelle 1: Stückliste Schwungradauto

Nummer	Bezeichnung	Anzahl
35129	Grundplatte 120·60·7,5 schwarz	1
37237	Baustein 5	6
38242	Bauplatte 15·45 2Z rot	1
31005	Baustein 15 grau	4
35049	Baustein 5 15·30 rot	2
38423	Winkelstein 10·15·15	2

Tabelle 2: Stückliste Service-Station

Überlegung zum Tuning

Für die Geschwindigkeit des Fahrzeugs ist der Durchmesser der Hinterräder unerheblich; entscheidend ist der Durchmesser der Schwungradachse. Will man das Fahrzeug „frisieren“, kann man zwei Klemmkupplungen 20 ([31024](#)) auf die Enden der Achse stecken. Die haben einen Durchmesser von 8 mm, so dass das Fahrzeug mit fast doppelter Geschwindigkeit losfahren sollte.

Modell-Varianten

Leichtbau-Renner

Auch mit der Drehscheibe 60·5,5 sind ordentliche Fahrleistungen zu erzielen, indem man das Fahrzeug etwas leichter baut.



Abb. 10: Das Leichtbau-Trike



Abb. 11: fischertechnik-Mann fährt mit Sidestick Steuerung

Die Sidestick Steuerung wurde von Daimler in einem Mercedes-Benz SL der Baureihe R 129 erprobt [5].

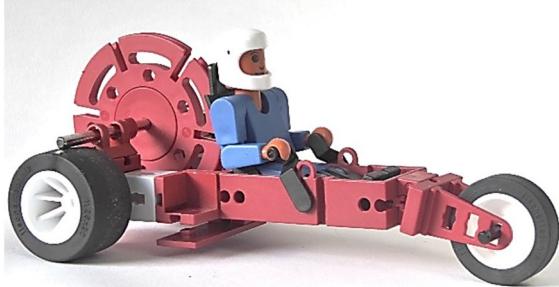


Abb. 12: Sidesticks statt Lenkrad und Pedale

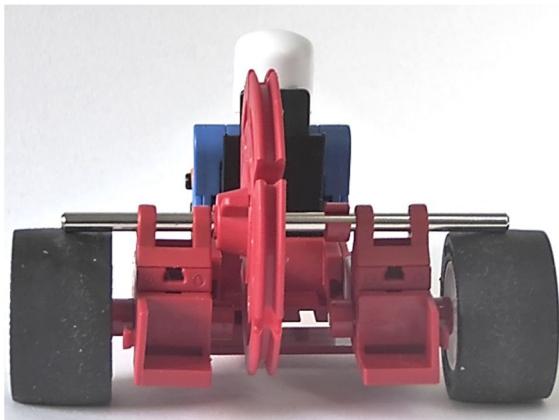


Abb. 13: Heckansicht

Die Antriebsachse erhält nur eine Klemmbuchse. Der Abstand zur Gelenkwürfelaufnahme sollte etwa 1 mm betragen. In diese Klemmbuchse wird die Schlaufe des Seils eingehängt.

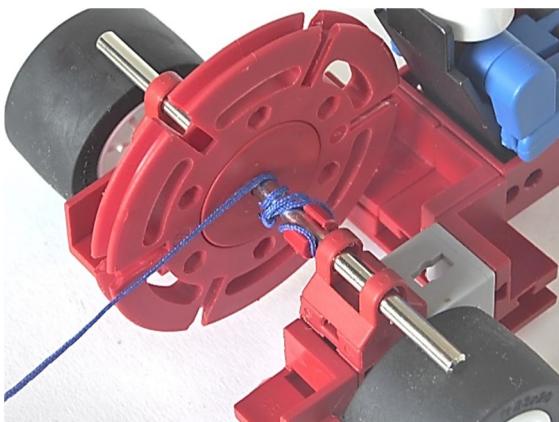


Abb. 14: Der modifizierte Seilzug

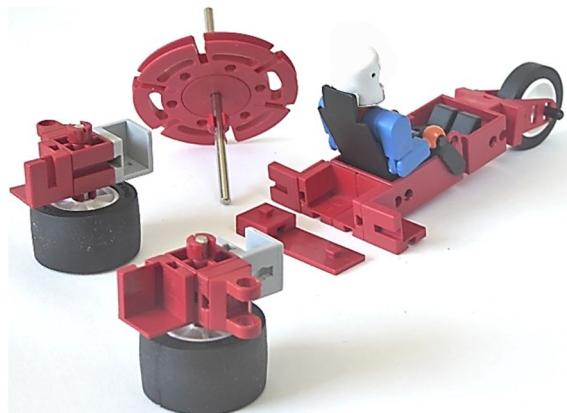


Abb. 15: Zerlegt

Das Aufwickeln des Seils erfolgt wieder auf der Service-Station.

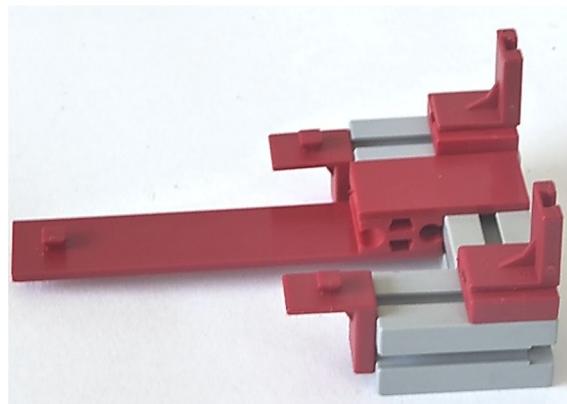


Abb. 16: Die Service-Station

Zum Start wird die Service-Station wieder herumgedreht; sie hält das Fahrzeug beim Anwerfen nieder.

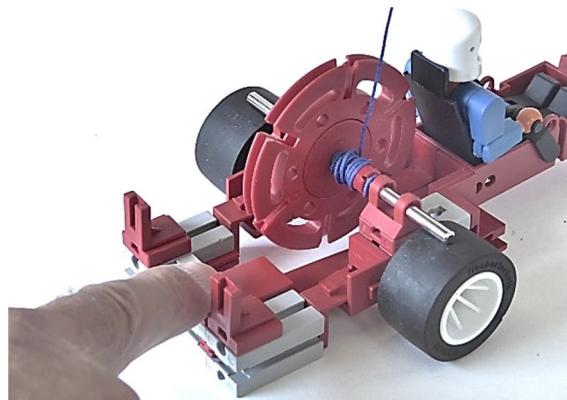


Abb. 17: Start

Ein Doppel-Schwungrad-Fahrzeug

Dies soll eine Anregung für eigene Versuche sein, deshalb verzichte ich auf eine genaue Beschreibung.



Abb. 18: Doppel-Schwungrad-Fahrzeug



Abb. 19: Das doppelte Schwungrad

Experimental-Fahrzeug

Dies ist der Versuch einer radikalen Vereinfachung.

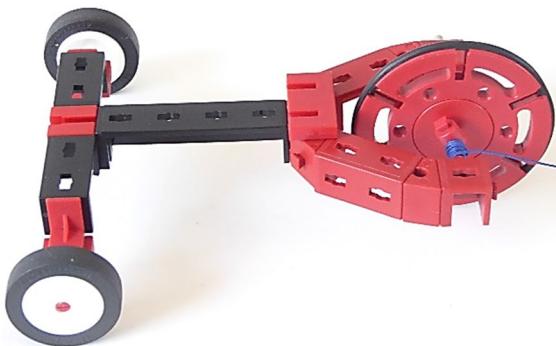


Abb. 20: Auch so etwas lässt sich in Schwung bringen

Das Seil ist jetzt nur etwa 30 cm lang und wird durch eine Klemmhülse ([35980](#)) fest mit der Achse verbunden. Das andere Ende erhält einen Knoten zum besseren Festhalten.

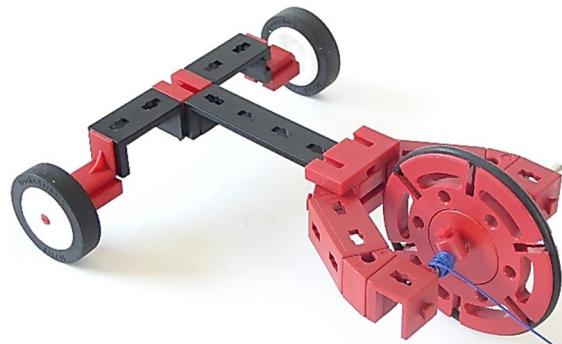


Abb. 21: Schrägangsicht

Das Seil wird wie bisher aufgewickelt; dabei hilft das überstehende Ende der Achse (Abb. 22). Dann hält man das Fahrzeug mit dem Schwungrad nach oben in der Hand und zieht das Seil rasch nach oben, diesmal aber nicht vollständig! Kurz vor dem Ende lässt man los. Das Schwung- und Antriebsrad dreht sich schnell und das Seil wickelt sich verkehrt herum auf. Wir setzen das Fahrzeug auf den Boden und ab geht's!

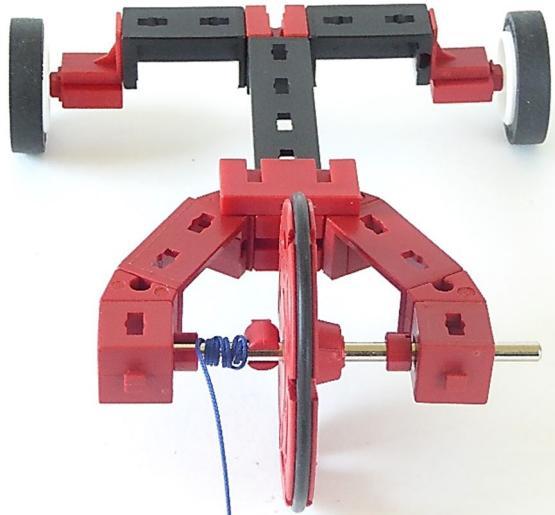


Abb. 22: Experimental-Trike von vorne

Das Seil stört nicht beim Fahren. Zum Neustart wird es abgewickelt und in der richtigen Weise wieder aufgewickelt. Abgeschaut habe ich diese Art des Seilzuges von einem Kreisel, den ich auf einem Flohmarkt erstanden hatte.

Den großen O-Ring 54·3 gibt es z. B. bei Stefan Roth [4] unter der Nummer [35678](#).

Mini

Mein kleinstes Schwungrad-Fahrzeug hat ein horizontales Schwungrad.

Der S-Riegelstein ([32850](#)) dient als Griff zum Anwerfen und Starten. Das Starterseil wird um die Rastkupplung ([35073](#)) gewickelt. Nicht durch die seitliche Öffnung durchstecken! Das Seil könnte sich sonst verhaken und die Rastkupplung würde abgerissen.

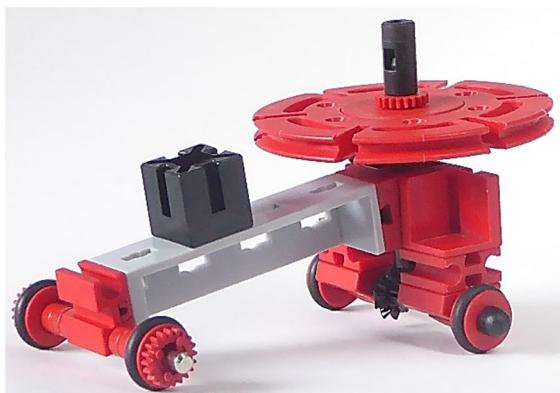


Abb. 23: Das Mini-Fahrzeug



Abb. 24: von vorne

Wie in Abb. 25 zu sehen ist, besteht das Getriebe aus einer „Rastachse 30 + Kegelzahnrad Z10“ ([35061](#)) und einem Rastkegelzahnrad Z10 ([35062](#)). Das rechte Hinterrad ist eine Rast-Seilrolle 12·5,5 ([136775](#)). Die linke Hinterradachse wird von einer „V-Achse 34 Clipachse“ ([32870](#))

gebildet, die in dem Kegelzahnrad klemmt. Das linke Hinterrad ist eine Seilrolle 12·5,5 ([38255](#)), sie sollte so ausgesucht werden, dass sie auf der Clipachse klemmt.

Die Vorderräder werden von Seilrollen 12·5,5 ([38255](#)) gebildet, alle vier Reifen sind O-Ringe 8·3 ([35677](#)) [4]. Für einen guten Geradeauslauf ist die vordere Spurweite recht groß gewählt, die Zwischenräume werden mit sechs Abstandsringen ([31597](#)) aufgefüllt.



Abb. 25: von hinten



Abb. 26: Das Seil ist aufgewickelt

Naja, die Fahrleistungen sind bescheiden: Über 1 m Fahrtstrecke komme ich nicht hinaus.

Elektrischer Schwung

Zum Abschluss geben wir dem Fahrzeug von Abb. 12 neuen Schwung mit dem Anwurf-Motor aus [6] (siehe Abb. 7).

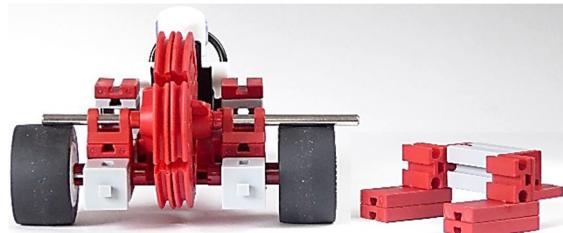


Abb. 27: Doppelschwungrad und Startrampe

Die Heckpartie wird vereinfacht, die Schwungradachse wird verlängert und steht jetzt auf der rechten Seite über.



Abb. 28: Die Startrampe

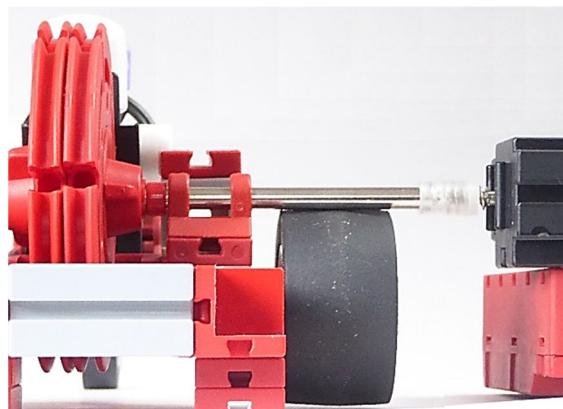


Abb. 29: Anwurf

Die Startrampe ist gerade so hoch, dass die Räder nicht den Boden berühren. Der Silikonschlauch des Anwurf-Motors wird über das freie Achsende geschoben. Nach ein paar Sekunden Motorlauf wird der

Anwurf-Motor abgezogen und das Fahrzeug startet durch leichtes Kippen der Startrampe.

Geschichtliche Entwicklung

Historisch brauchten die Dampfmaschinen und die frühen Verbrennungsmotoren große Schwungräder zur Überwindung der Totpunkte des Kurbelgetriebes bzw. Pleuels [7]. Energiespeicherung und -abgabe erfolgten innerhalb einer Umdrehung, also für sehr kurze Zeit.

Dagegen sollte das mechanische „Kinetic Energy Recovery System“ (KERS) in der Formel 1 über einen Zeitraum von ein bis zwei Minuten Bremsenergie in einem Schwungrad speichern und über ein stufenloses Getriebe wieder an die Antriebsachse abgeben. Von 2009 bis 2013 war KERS in der Formel 1 zur Benutzung freigegeben, allerdings war ihm kein Erfolg beschieden. Durchgesetzt hat sich die Energiespeicherung in Akkus.

Der Gyrobus aus der Schweiz stellte einen Großversuch dar und wurde ab 1953 auf mehreren Strecken erprobt. Das System arbeitete elektro-mechanisch, wurde aber nach wenigen Jahren wieder aufgegeben [8]. Die Reichweite betrug 6 bis 8 km, die Aufladung erfolgte an den Haltestellen über Drehstrom-Stromabnehmer.

Einen neuen Versuch mit Oberleitungen für Busse und LKWs, allerdings mit Batteriespeichern, kann man auf der A5 zwischen Frankfurt und Darmstadt besichtigen. Die Oberleitung ist im Gegensatz zu Bahn und Straßenbahn zweipolig. Wie bei Straßenbahnen wird mit 750 V Gleichstrom gearbeitet. Im Vergleich zu den drei bestehenden O-Bus-Linien in Solingen, Eberswalde und Esslingen sind nur die Stromabnehmer innovativ.

Bekanntermaßen sind Schwungräder nichts anderes als Kreisel und haben auch deren Eigenschaften [6]. Damit sie auch in Fahrzeugen genutzt werden können, sind

besondere Maßnahmen erforderlich. Die auf das Fahrgestell wirkenden Kreiselkräfte würden sonst die Fahreigenschaften stark beeinflussen. Sie lassen sich drastisch verringern durch zwei gegensinnig drehende Schwungräder.

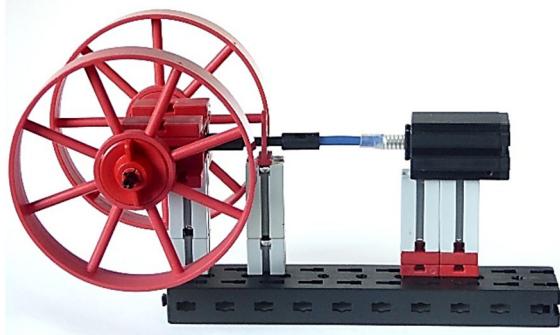


Abb. 30: Kaum Kreiselkräfte auf das Gestell

Beim Modell nach Abb. 30 und 31 treten beim Kippen kaum fühlbare Kreiselkräfte auf.

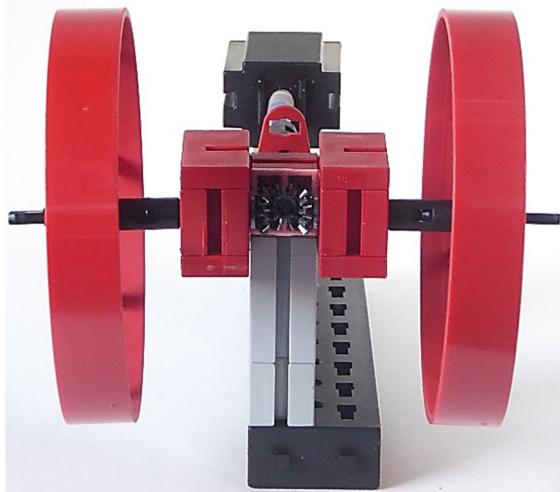


Abb. 31: Gegensinnig drehende Kreisel

Eine kardanische Aufhängung vermeidet jegliche Kräfte nach außen. Sie ist vor einigen Jahrzehnten intensiv diskutiert worden, zusammen mit ultrahoch drehenden Schwungrädern (ca. 50.000 U/min) aus hochfesten Fasermaterialien und mit Vakuum-Gehäusen.

Quellen

- [1] René Trapp: [Das Papa-Mobil](#). Bilderpool der ft-Community (Unsere Modelle | Radfahrzeuge | Fahrzeugtechnik+Prototypen | Gummibandauto | Das Papa-Mobil).
- [2] Esther Mietzsch: [Mausefallenauto 2.0](#). Bilderpool der ft-Community (Unsere Modelle | Radfahrzeuge | Fahrzeugtechnik+Prototypen | Mausefallenauto 2.0)
- [3] Thomas Dragon: [Mausefallenauto](#). „der fischertechnik-blog“, 2016.
- [4] fischerfriendsman: [www.fischerfriendsman.de](#)
- [5] Daimler: [Mercedes-Benz SL der Baureihe R 129 erlaubte 1998 den Blick in die Zukunft des Lenkens](#).
- [6] Rüdiger Riedel: *Kreisel und mehr*. ft:pedia [3/2018](#), S. 9-16.
- [7] Dirk Fox, Thomas Püttmann: *Technikgeschichte mit fischertechnik*. dpunkt.verlag, Heidelberg, 2015, S. 174.
- [8] Wikipedia: [Gyrobus](#).

Modell

Die Kettenuhr

Bernd Bekehermes

Angeregt für den folgenden Text wurde ich durch den Beitrag von Rüdiger Riedel in der ft:pedia-Ausgabe 2/2018 [3]. Darin wird auf Seite 24 in den Abbildungen 1 und 2 eine wegen ihrer Schlichtheit in der Farbe und in ihrer vollendeten Form sehenswerte Uhr vorgestellt. Statt Zeigern dreht sich ein leichter quadratischer Ziffernrahmen frei hängend, aber nur im entscheidenden Bereich sichtbar. Das Design folgt perfekt dem Credo eines der einflussreichsten Industriedesigner der Moderne, Dieter Rams [2]: „Weniger, aber besser.“ Rüdiger Riedel stellt zu seiner Zahnstangenuhr ausführlich mögliche Zahnstangengetriebe und einen Uhrennachbau mit fischertechnik vor. Bei meiner Förderketten-Wanduhr ging es mir darum, das Design einer Wanduhr möglichst schlicht mit fischertechnik-Teilen nachzugestalten.

Hintergrund



Abb. 1: Die Catena-Wanduhr

Auf einer Design-Ausstellung der Gruppe „anthologie quartett“ sah ich 2001 die „catena“ (Abb. 1) Die Catena (lateinisch für Kette) ist eine Wanduhr, die 1993 vom deutschen Designer Andreas Dober [3] entwickelt wurde. Die Catena ist eine mechanische Wanduhr mit Federwerk und Schlag. Das Uhrglas-förmige Gehäuse aus patiniertem Eisen hat einen Durchmesser von 170 mm. Die Tiefe ist mit 75 mm, die Höhe mit 500 mm angegeben. Die der Fahrradtechnik entlehnte Kette trägt 12 kupferne Stundenziffern.

Die Catena besticht durch ihr ungewohntes Design mit schlichtem technischen Aussehen. Das Uhrglas-förmig gewölbte runde Gehäuseschild verdeckt das Uhrwerk und weist gleichzeitig auf eine Uhr hin. Die unsichtbar langsame Auf- bzw. Abbewegung der Ziffernkette ist leicht zu überblicken und zu erfassen. Die Kette lässt Bewegung nur ahnen, die Zeit steht scheinbar still. Vorausgesetzt, man erwartet von einer unbekannten Uhr mit Ziffern die Drehrichtung im Uhrzeigersinn, lässt sich schnell herausfinden, wo und wie die Uhrzeit abzulesen ist. Natürlich ist ein minutengenaues Ablesen nicht möglich, aber Viertel- und halbe Stunden sind gut zu erkennen.

Nachbau mit fischertechnik

Der Nachbau eines technischen Gerätes ist sehr reizvoll und nicht immer einfach. Das betrifft nicht nur konstruktive Probleme. Auch das Design, die optische Ausstrahlung, ist zu berücksichtigen. Ein freiliegendes fischertechnik-Räderwerk, zum Beispiel eines Uhrwerks, kann allein schon begeistern. Wenn es aber darum geht, das Erscheinungsbild eines schlicht gestalteten Designobjektes mit fischertechnik-Bausteinen nachzugestalten, bedingen Nuten, Zapfen und Bohrungen der vielseitigen fischertechnik-Bausteine einige Kompromisse.

Mein Nachbau der Catena sollte zwei Anforderungen erfüllen: Möglichst schlichtes Design des Gehäuses, welches nur die Bedingung erfüllen muss, das Uhrwerk zu verstecken. Außerdem war mir wichtig, fischertechnik-Bauteile möglichst unverändert heil zu verbauen.

Die Drehscheibe (31019) eignet sich kaum, um mit Bauplatten bedeckt die glatte Vorderseite eines runden Uhrengehäuses zu bilden. Leichter lässt sich mit rechteckiger Fläche arbeiten. Darum dient bei meiner Kettenuhr eine Grundplatte 90x90x5,5 (31002) mit zentraler 4 mm-Bohrung als Vorderseite (Abb. 2). Das Uhrwerk, ein Quarzwerk mit 1,5 V-Baby- oder Mignonbatterie, wird rundum durch vier Bausteine V15 Eck (38240) und vier rote Bauplatten 30x90 (38251) verdeckt. Um das Werk nicht durch die langen Spalten der Grundplatte sichtbar werden zu lassen, trägt das Uhrwerk im durchscheinenden Bereich rechts und links je einen Streifen aus rotem Isolierband und zwischen den BS V15 Eck rückseitig vier rote Federnocken (31982) neben den Ecken, dazu senkrecht links und rechts je eine rote Bauplatte 15x60 (38464). Diese sind nur so schmal gewählt, damit das Uhrwerk ganz dicht mit Doppelklebestreifen mittig hinter die Grundplatte geklebt werden kann. Das gelingt, weil

mein Uhrwerk kein vorstehendes Gewinderohr zur Verschraubung mit einem „normalen“ Zifferngehäuse hat.



Abb. 2: Rückansicht auf Uhrwerk im fischertechnik-Gehäuse

Zwei von vier Federnocken sind in Abb. 2 oben als Lückenfüller zu erkennen. Zum Antrieb reichen für längere Zeit sogar Batterien, die in anderen Geräten keine rechte Leistung mehr bringen oder als Fundstücke, in der Umwelt aufgelesen, ihre Restlaufzeit abarbeiten.

Vor dem Ankleben ist das Quarzwerk noch etwas anzupassen. Das äußerste Röhrchen zur Aufnahme eines Zeigers ist für den Stundenzeiger gedacht, der erst in 12 Stunden einen vollen Kreis beschreibt. Für den Antrieb der Stundenkette ist eine ganze Umdrehung des nächst kleineren Wellenröhrchens nötig. Den bietet nur die Minutenzeiger-Aufnahme. Mit Puksäge und Staubsauger ließ sich das äußere Röhrchen dicht am Gehäuse vorsichtig absägen. Eine eventuell vorhandene Sekundenachse kann ohne Zeiger bleiben. Sie ist eventuell noch zu kürzen.

Der Minutenzeiger-Antrieb musste nun mit Klebeband oder einem ganz kurzen Plastikröhrchen mit unter 4 mm Stärke so verlängert werden, dass der Antrieb vorne aus

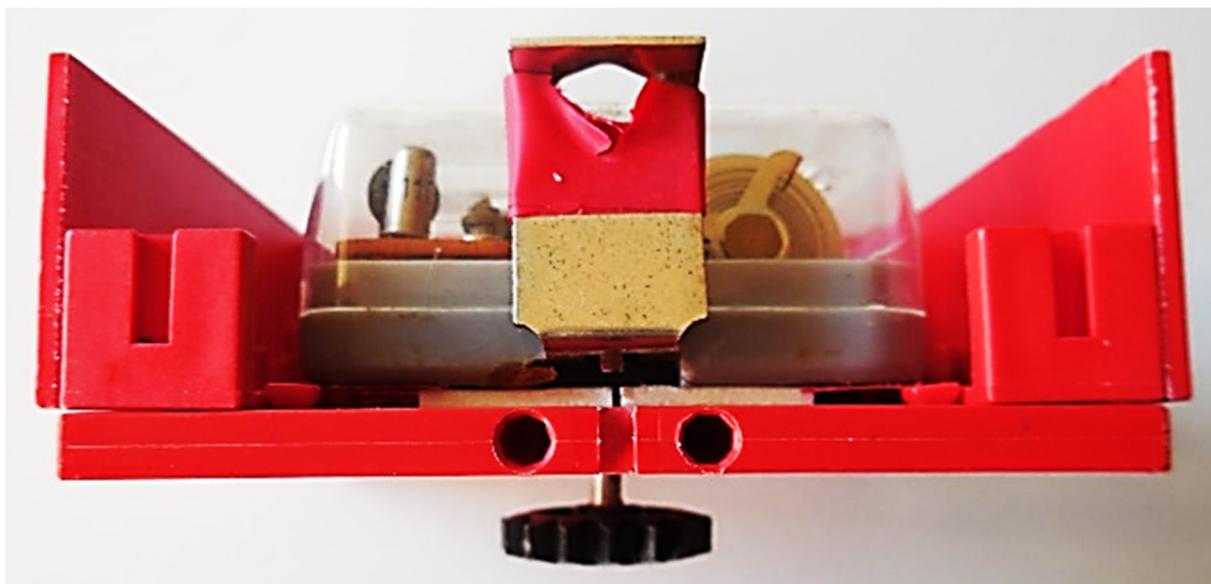


Abb. 3: Draufsicht auf Grundplatten-Oberseite (obere Bauplatte 30x90 fehlt fürs Foto) mit Quarzwerk (mit Metallhaken), fixiert mit zwei weißen Doppelklebestreifen.

der Mittenbohrung der Grundplatte herausragt. Darauf ist eventuell nach leichter Aufweitung auf 4 mm vorn ein Zahnrad als Kettenantrieb festzuklemmen. Ich habe ein schwarzes Klemmzahnrad Z15 ([37685](#)) gewählt (Abb. 3).

Die Metallkette der Catena ist leicht durch fischertechnik-Kettenglieder zu ersetzen, bei meiner Uhr in der ursprünglichen Art ([36263](#)) ohne Rastvierkant. Für die 12-Stunden-Kette benötigt man $12 \cdot 14 = 168$ rote Kettenglieder und für jede Stunde dazu ein Förderglied ([37192](#)) als Träger für 12 Stundenziffern. An diesen habe ich versuchsweise zunächst Bausteine 15 und Winkelsteine 10x15x15 befestigt, jeweils bestückt mit roter Bauplatte 15x15. Sie erwiesen sich wegen des überhängenden Gewichts, besonders aber auch optisch als zu schwer. Auch Radachsen ([36586](#)) mit gelochten Ziffernscheiben wurden erworben.

Schließlich habe ich eine hellgraue Zwischenwand aus einem Sortierkasten geritzt, gebrochen, gefeilt und die Kanten auf das Maß 15 mm · 15 mm geschliffen.



Abb. 4: Kette mit zwei Fördergliedern und zwei Ziffernträgern

Versehen mit 10 mm hohen Selbstklebeziffern folgte, auch mit kleinen Stückchen Doppelklebeband, die Fixierung an die geschlossene Seite von schmalen Raupenbelägen ([37210](#); 14,5x4x9; schwarz; Abb. 4). Damit sich die Raupenbeläge nicht am Uhrgehäuse verhaken, dürfen sie nur wenig über die Förderglieder in Richtung Grundplatte aufgeschoben werden.

Damit besonders die unteren Zifferblättchen durch die seitlich versetzt angreifende Schwerkraft nicht so schräg hängen, dreht sich mit der Kette unten in der Ketten schlaupe, wie bei der Catena, ein weiteres

Klemmzahnrad 15, beschwert mit rückseitig eingelegten Unterlegscheiben zur Erhöhung des Gewichts (Abb. 5).

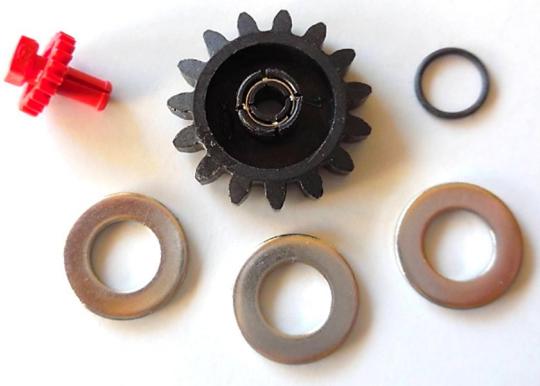


Abb. 5 Das untere Mittdrehzahnrad Z15 mit Zusatzgewichten

Seit 2002 läuft die Kette schon rund. Eine Gangregulierung erfolgt rückseitig am Quarzwerk. Nach Stillstand kann die Kette umgehängt werden. Eine Rutschkupplung in meinem Uhrwerk erlaubt auch ein Verstellen durch leichten Zug an einer Kettenseite – zum Beispiel zur Umstellung auf Sommer- bzw. Winterzeit.

Quellen

- [1] Rüdiger Riedel: [Die Zahnstangen-Uhr](#). ft:pedia 2/2018, S. 24-37.
- [2] Wikipedia: [Dieter Rams](#). Deutscher Industriedesigner, siehe: „Zehn Thesen für gutes Design“.
- [3] Andreas Dober: [Catena Wanduhr](#) (auf adero.de).

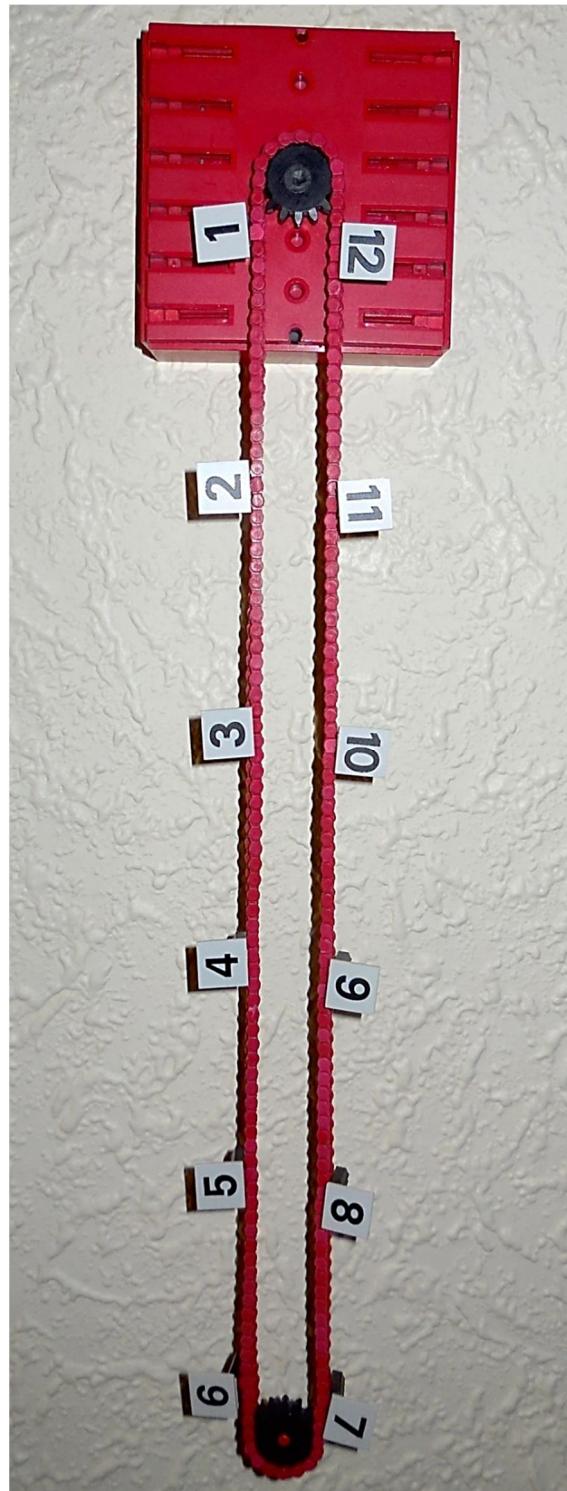


Abb. 6: Die fischertechnik-Kettenuhr um kurz nach halb eins

Modell

Kirchturmuh mit Huygenschem Aufzug

Matthias Dettmer

Das Modell einer Kirchturmuh aus dem Buch „Technikgeschichte mit fischertechnik“ [3] ist inzwischen wohl bekannt und mittlerweile bestimmt viele Male nachgebaut worden. Lässt man das Antriebsgewicht von einem „normalen“ Schreibtisch aus herunterlaufen, dann läuft die Uhr nur etwa eine Stunde: $4 \cdot \text{Umfang Drehscheibe } 60 (2 \cdot \pi \cdot 30 = 188,5 \text{ mm})$ ergeben um die 75,4 cm Seil. Für mein eigenes Modell wollte ich gerne längere Laufzeiten – ein paar Stunden vielleicht – umsetzen.



Abb. 1: Die Trommel rechts treibt die Uhr an, der Hebel darunter hebt das Gewicht (unten in der Mitte). Hemmung und Pendel sind im Bild hinten zu erkennen.

In meiner Kirchengemeinde war ich mal dafür verantwortlich, jeweils am Sonntag nach dem Gottesdienst unsere Kirchturmuh für die nächste Woche aufzuziehen – die Uhr läuft dann sieben Tage, aber das

Gewicht dort hat auch einen knapp 10 m hohen Schacht. Damit diese Uhr während des Aufziehens nicht stehenbleibt, gibt es einen Hebel mit einem Gewicht, der die Aufgabe der Seiltrommel für die Dauer des Aufziehens für etwa fünf Minuten übernimmt (Abb. 2).

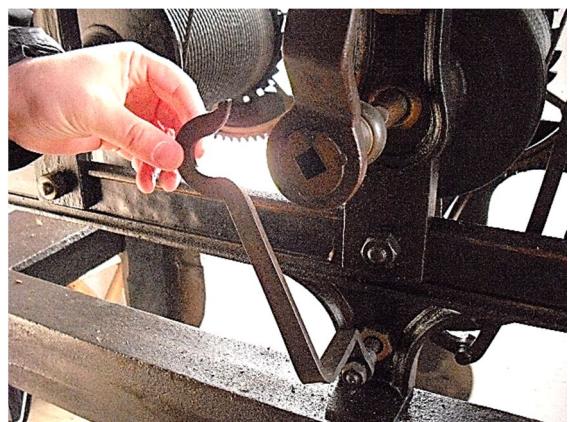


Abb. 2: Vor dem Aufziehen wird mit diesem Hebel die sogenannte „Gangreserve“ angehoben, und die betreibt dann die Uhrenmechanik weiter.

Mein erster Ansatz war es, diese Gangreserve in fischertechnik nachzubilden, um dann zusätzlich eine Mechanik für das Aufziehen selbst „anzuflanschen“. Nun habe ich recht wenig Freizeit, also schob ich dieses komplexe Vorhaben immer wieder nach hinten. Später.

Eines ruhigen Abends bin ich dann auf die hochgenaue Uhr [1] von Martin Roman aka „remadus“ gestoßen – zuerst hatte ich von dessen „Pendelprüfstand“ und den dort verfolgten Ansätzen gelesen. Es gibt auch ein YouTube-Video der hochgenauen Uhr [2]. Ein spezielles Detail hat mich dann elektrisiert: Die Uhr hatte einen motorisierten Aufzug unter Verwendung eines Endlosseils. Weiteres Nachforschen und -lesen ergab, dass diese Art des automatischen Aufzuges einen Namen hat: das ist ein *Huygensscher Aufzug*.



Abb. 3: Kirchturm-Uhr mit Huygenschem Aufzug

Ich glaube schon, dass es vermessen wäre, das einfache Kirchturmuhren-Modell mit dieser hochgenauen Uhr von remadus vergleichen zu wollen. In seiner Uhr steckt so viel Technik und ausgetüftelte Mechanik; hier würden Äpfel mit Birnen verglichen. Aber ich habe mich schon gefragt, ob man diese Technik nicht einfach an das einfache Modell adaptieren könnte.

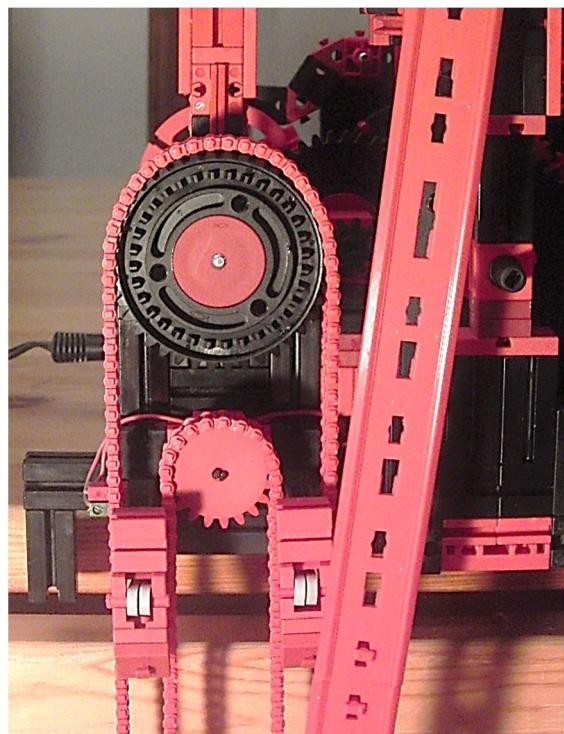


Abb. 4: Anordnung der Zahnräder im Aufzug.
Schwarz: Antrieb Uhrwerk,
rot: Antrieb des Aufzugs

Eines der dort geschilderten technischen Probleme ist die Herstellung eines Endlos-Seils, das man für diesen Aufzug braucht. Die Verbindung der beiden Seilenden darf nicht zu steif sein, sie sollte in etwa die gleiche Dicke wie das restliche Seil haben, und *last, but not least* aber so stabil sein, dass die Enden nicht dauernd auseinanderreißen. fischertechnik bietet ein höchst flexibles Endlosseil an, in der Länge variabel und unter Beachtung aller Vorgaben einfach zusammenzubauen: die Kette.

Der Huygenssche Aufzug kann einfach erklärt werden, indem ich einfach „einmal herum“ die Kette entlanggehe. Der beste

Anfang dafür ist das Antriebsrad der Uhr. Die Drehscheibe 60 wird durch ein Zahnrad Z40 ersetzt, somit haben wir etwa den gleichen Durchmesser.

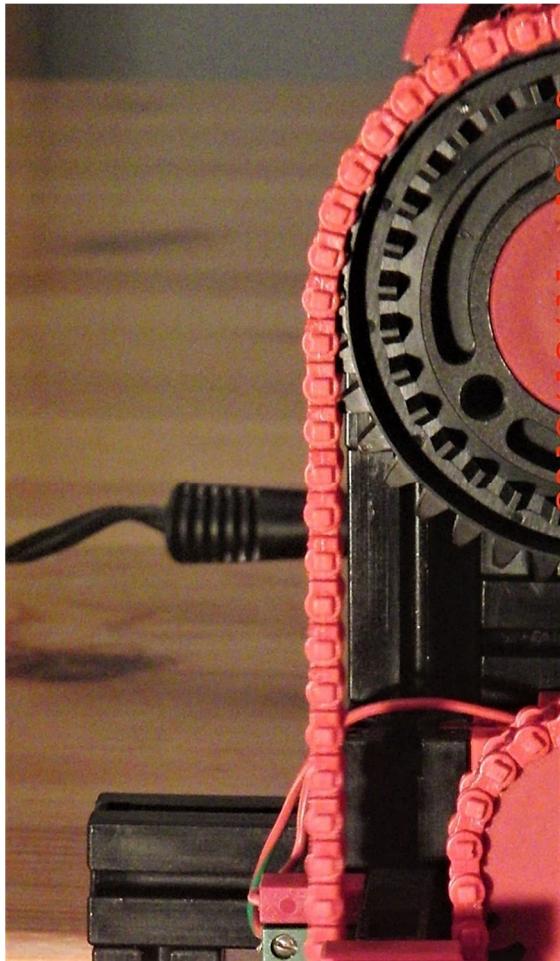


Abb. 5: Das Antriebsrad des Uhrwerks

Ein Stückchen weiter unten an der Kette hängt das Antriebsgewicht an einer Rolle. Hier habe ich einfach eine der uralten Seiltrommeln verwendet, durch die die Kette gerade durchrutscht.

Nach der Antriebsgewicht-Rolle wandert die Kette wieder nach oben, bis hin zum Antrieb des Aufzugs. Das ist ein Zahnrad Z20, das von einem Getriebemotor (einem Power-Motor 50:1) festgehalten wird. Ursprünglich war hier ein selbsthaltendes Schneckengetriebe geplant. Würde dieses Zahnrad nicht festgehalten, dann zöge das Gewicht alles, was es an Kette finden kann, nach unten. Der Powermotor mit seiner

Untersetzung von 50:1 hält das Aufzugsantriebsrad aber sicher fest.

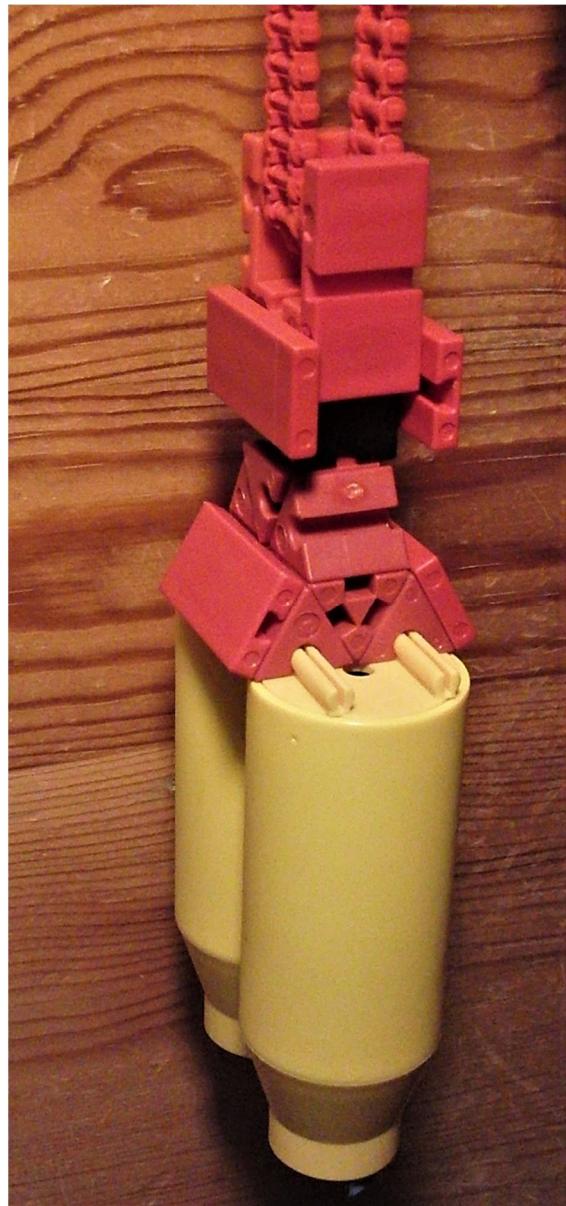


Abb. 6: Antriebsgewicht

Auf der anderen Seite des Aufzugsantriebs wandert die Kette nach unten zum Ausgleichsgewicht, dort wieder durch eine „Seiltrommel-Rolle“ und nach oben zurück zum Uhrenantriebszahnrad. Wozu ist das Ausgleichsgewicht gut? Damit der Aufzugsantrieb nicht dauernd laufen muss, braucht es so etwas wie eine Seil-, pardon, Kettenreserve. Und würde diese Reserve einfach so herumhängen, dann würde sich

die Kette irgendwann aufwickeln, verkannten, Stau bilden. Deswegen wird die Kette durch das Ausgleichsgewicht in eine definierte Richtung geführt. Wer's nicht glaubt, sollte das einmal nachbauen und das Ausgleichsgewicht weglassen. Das wird nicht lange gut gehen; die fischertechnik-Kette hat genügend Eigenreibung von Gelenk zu Gelenk, sodass es zu Stockungen kommt – bei mir ist die Kette auch mal vom Aufzugsantriebszahnrad abgesprungen.

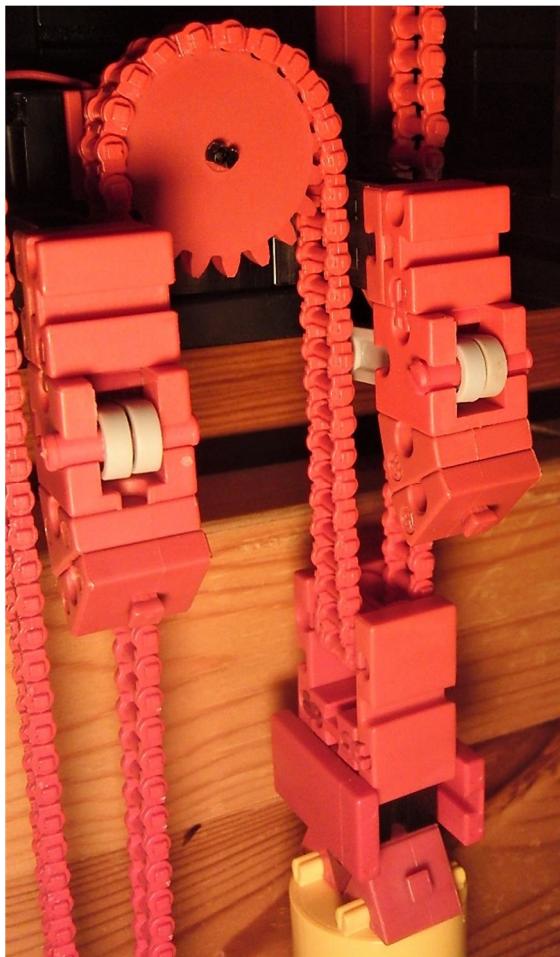


Abb. 7: Aufzugsantrieb

Wie schwer müssen dann die Gewichte sein? Fangen wir diesmal bei dem Ausgleichsgewicht an. Hier wird nicht so viel Gewicht benötigt, wenige 10 Gramm reichen aus. Nach Annäherung sind das bei mir 30 Gramm.

Für den Antrieb der Uhr wurde bei der Seilrolle ein Gewicht von ca. 80 Gramm benötigt. Fängt man damit in der neuen

Anordnung an, dann bewegt sich nichts: Das sich aus Gewicht mal Hebelarm ergebende benötigte Drehmoment \vec{M} reicht nicht für den Antrieb:

$$\begin{aligned} \text{Drehmoment} &= \text{Masse} \\ &\cdot \text{Erdbeschleunigung} \\ &\cdot \text{Hebelarm} \end{aligned}$$

$$\begin{aligned} \vec{M} &= m \cdot g \cdot l \\ \vec{M} &= 0,08 \text{ kg} \cdot 9,81 \frac{\text{m}}{\text{s}^2} \cdot 0,03 \text{ m} \\ &= 0,0235 \text{ Nm} \\ &= 2,35 \text{ Ncm} \end{aligned}$$

Technisch ist das klar: Wo vorher ein Seil das ganze Gewicht getragen hat, wird dieses Gewicht jetzt von zwei Kettensträngen getragen – die wirksame Gewichtskraft wird halbiert. In der Formel oben stehen dann nicht 0,08 kg, sondern nur 0,04 kg.

Nächster Schritt also: Das Gewicht verdoppeln, damit die Formel wieder stimmt. Und es geht immer noch nicht...

Natürlich dürfen wir die Hälfte des Ausgleichsgewichtes und die Reibungsverluste an und in der Kette nicht vergessen, die wollen auch noch „angetrieben“ werden. Am Ende ergibt sich ein Gesamt-Antriebsgewicht von ca. 380 g.

Nachdem damit das Prinzip erklärt ist, bleibt die Frage: Wie kann ich jetzt das Ganze automatisieren? Der erste Gedanke war, Lichtschranken und den TXT oder sogar die uralten Silberling-Relais mit Vorverstärkern zu verwenden. In der Tat kann man das aber auch nur mit Tastern und „normalen“ Relais umsetzen.

Dazu habe ich je einen Taster in die Ketten schlaufe des Antriebsgewichtes und des Ausgleichsgewichtes hineingebaut. Wenn die Uhr ordentlich angetrieben wird, dann wandert das Antriebsgewicht ganz langsam nach unten (siehe oben, ca. 18,8 cm pro Viertelstunde). Gleichzeitig wandert das Ausgleichsgewicht mit der gleichen Geschwindigkeit nach oben. Der Taster

über dem Ausgleichsgewicht wird betätigt, sobald die Rolle „oben ankommt“.

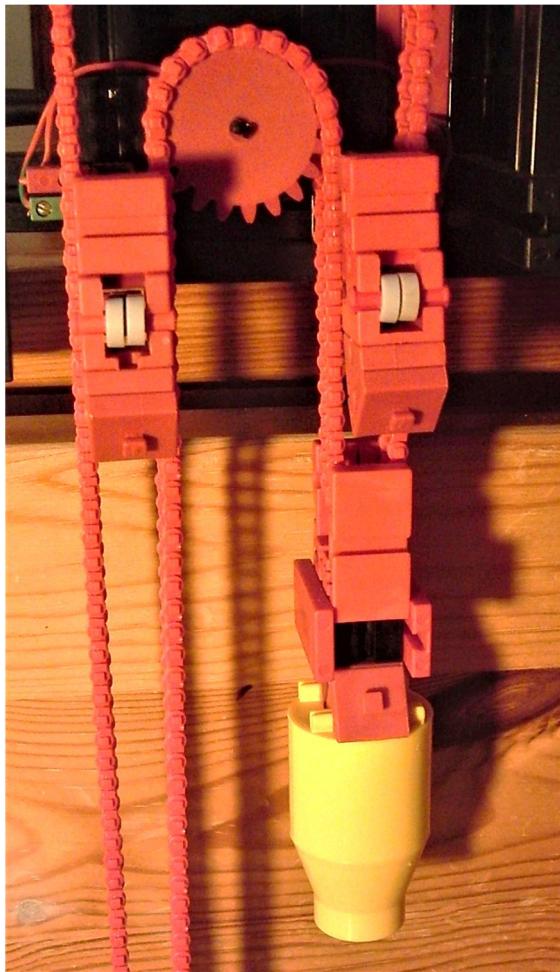


Abb. 8: Gleich wird geschaltet...

Der Taster ist ein Ein-Schalter und schaltet das Relais über dessen ersten Umschalter in eine Selbsthaltung. Der zweite Umschalter des Relais schaltet den Aufzugsmotor ein. Das geht so einfach, weil der Motor ja immer nur in einer Richtung betrieben wird: von vorne gesehen im Uhrzeigersinn.

Der Motor läuft also so lange, bis die Selbsthaltung getrennt wird. Das passiert, sobald das Antriebsgewicht oben ankommt. Der Taster in dieser Kettenschlaufe ist ein Ausschalter, bei Betätigung trennt er die Selbsthaltung des Relais auf, der Motor bleibt stehen. Hier hätte man auch eine Kurzschluss-Bremse umsetzen können, die aber keinen zusätzlichen Nutzen bringen würde. Abb. 11 zeigt die elektrische Schaltung.

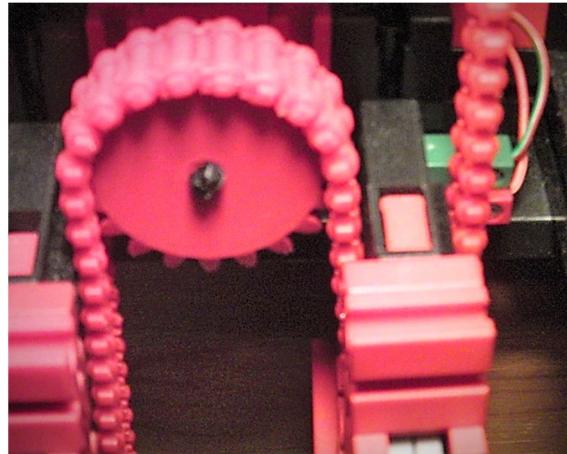


Abb. 9: Hier ist der Taster betätigt, für das Ausgleichsgewicht geht's jetzt abwärts.

Beim Ausprobieren gibt es eine mögliche Fehlerquelle: Ist der Motor falsch herum gepolt, dann wird das Ausgleichsgewicht noch stärker gegen den Auslösetaster gezogen, der Motor wird mit der Zeit überlastet. Deswegen rege ich an, beim ersten Versuch den Tasterhebel unter dem Ausgleichsgewicht mit dem Finger zu betätigen, während dieses noch etwas weiter unten hängt, und den anderen Finger zwecks schnellem Ausschalten am zweiten Tasterhebel zu haben. Wandert das Ausgleichsgewicht nach oben, dann muss der Motor umgepolt werden. Wandert es aber nach unten, dann ist alles in Ordnung. Bei meinem Modell läuft die Uhr auch während des Aufziehvorganges ohne merkbare Stockungen weiter.

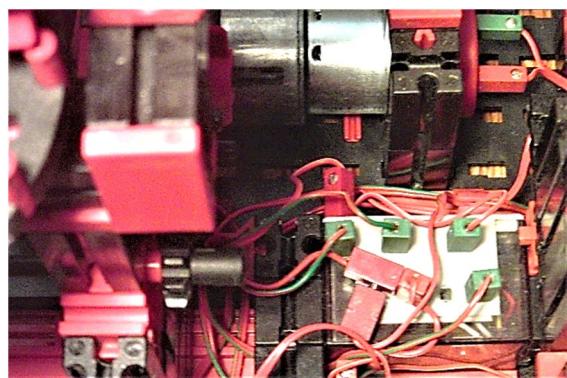


Abb. 10: Antriebsmotor für den Uhrwerks-Aufzug und Relais für die Steuerung

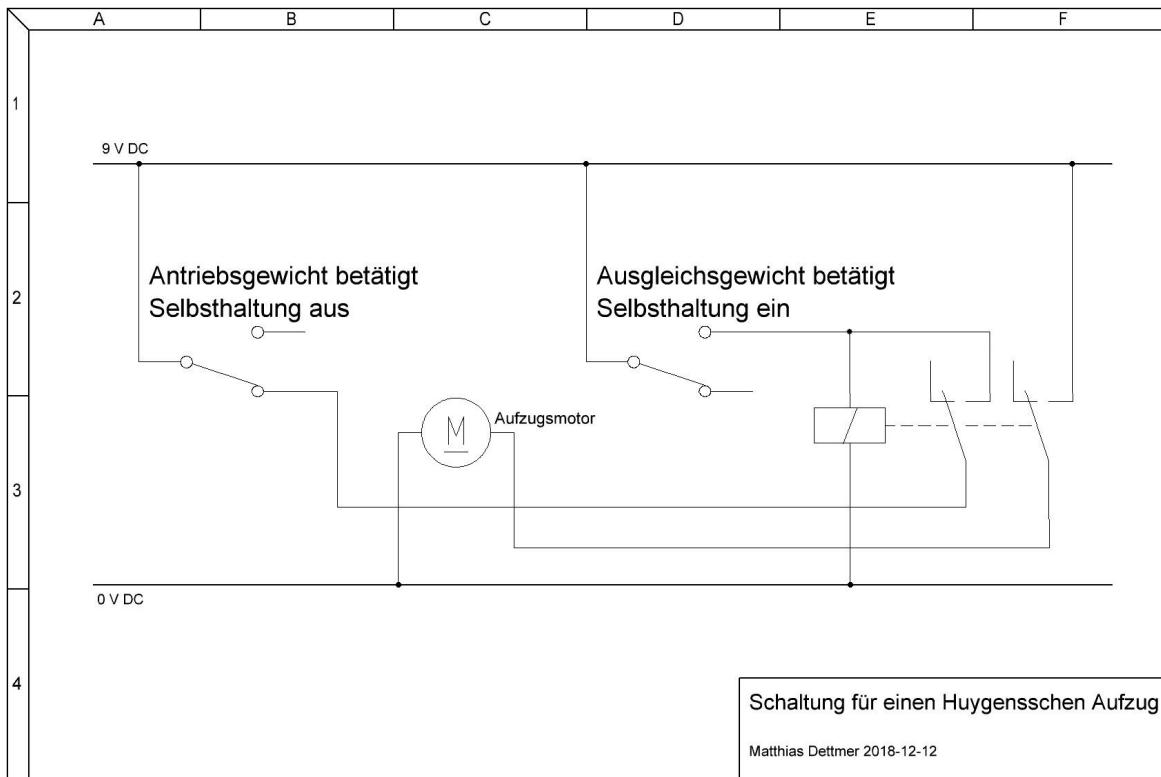


Abb. 11: Die elektrische Schaltung

Abschließend noch ein paar Hinweise zu weiteren Änderungen mit Bezug auf das Original:

- Der Antrieb des Schlagwerkes ist eine genaue Kopie des Uhrwerksantriebes, und zur Ausgestaltung des richtigen benötigten Gewichtes habe ich mir keine weiteren Gedanken gemacht. Das gleiche Gewicht reicht hier wie dort.
- Im Uhrwerk sind alle Achsen kugelgelagert.
- Weil die Achse des Pendels ein wenig weiter über die Tischkante ragen sollte, habe ich die Mechanik „umgedreht“. Deswegen baut das Uhrengetriebe etwas schmäler, und das Antriebsrad mit den Stiften dreht anders herum als beim Original.
- Auch das Schlagwerk ist nahezu vollständig kugelgelagert.

- Das ursprünglich verwendete Differenzial wurde auf Vorschlag von Thomas Püttmann durch ein Planetengetriebe ersetzt – siehe hierzu auch die Anleitung im Internet.
- Es gibt jetzt ein richtiges „Ziffernblatt“.
- Das Schlagwerk besteht aus Klangrohren (aus dem Kugelbahn-Kasten „M“)

Die Entwicklung des Modells hat etwa 80 Stunden gedauert; insgesamt ist das Ganze viermal „von der Grundplatte her“ aufgebaut worden. Aber die Mühe ist belohnt worden: Während der Modellausstellung auf dem MINT-Feriencamp 2018 in Karlsruhe lief die Uhr über fast sieben Stunden mit nur minimalen Unterbrechungen. Die Ungenauigkeit war kleiner als zwei Minuten.

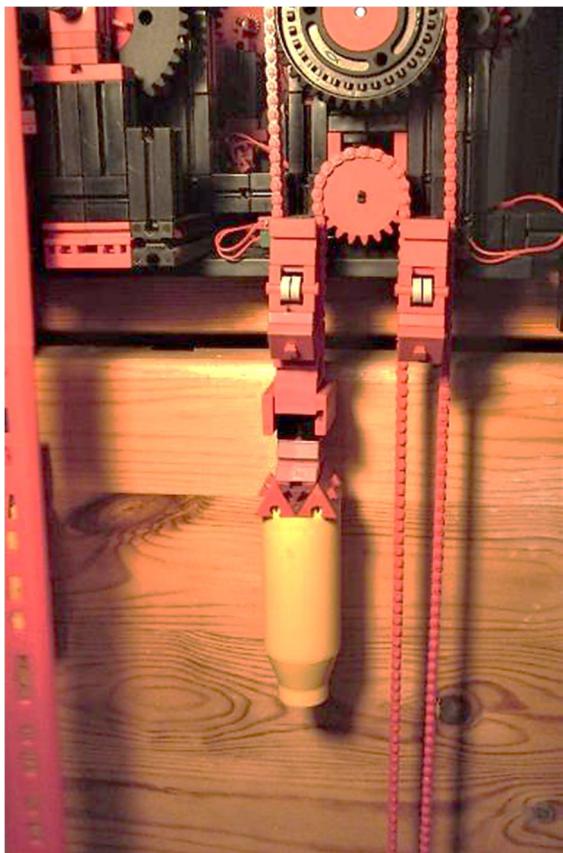


Abb. 12: Der Aufzug für das Schlagwerk ist 1:1 wie der Uhrwerksaufzug gebaut.

Und auch auf dem Fan-Club Tag Ende Juni bei fischertechnik lief die Uhr sechs Stunden durch. Zwecks Demonstration der Aufzugsfunktion hatte ich die Ketten für eine Betriebsdauer zwischen zwei Aufzügen auf etwa 35 Minuten abgelängt.

Quellen

- [1] Martin Romann: [Entwicklung einer Pendeluhr von der Hemmung bis zur fertigen Maschine](#) auf der ft-Community, 2010.
- [2] Martin Romann: [fischertechnik Präzisions-Pendeluhr](#). YouTube-Video, 2011.
- [3] Dirk Fox, Thomas Püttmann: [Technikgeschichte mit fischertechnik](#). dpunkt Verlag, 2015.

Modell

Useless Machine – zwecklose Maschine

Dirk Fox

Wer ein fischertechnik-Modell konstruiert, möchte damit eine bestimmte Funktion realisieren. In der Regel dient diese Funktion auch einem (meist nützlichen) Zweck – wie der Fortbewegung, dem Transport oder dem Vergnügen. Aber natürlich ist oft die raffinierte Realisierung der Funktion selbst das Faszinierende – manchmal sogar dann, wenn sie ihren eigentlichen Zweck nur eingeschränkt erfüllt. Was liegt da näher, als die Idee einer Funktion um ihrer selbst willen zu realisieren – „zweckbefreit“, gewissermaßen. Nicht neu, aber bestechend.

Geschichte

Die Geschichte der „Useless Machines“ (nutz- oder sinnlosen Maschinen), also von Maschinen, die eine Funktion, aber keinen Zweck haben, geht bis in die 1930er Jahre zurück. Der Künstler *Bruno Munari* (1907–1998), später ein einflussreicher Industriedesigner, konstruierte wohl als erster solche zweckbefreiten Mechanismen und verstand sie als Kritik an der zunehmenden Beherrschung unserer Welt durch Maschinen.

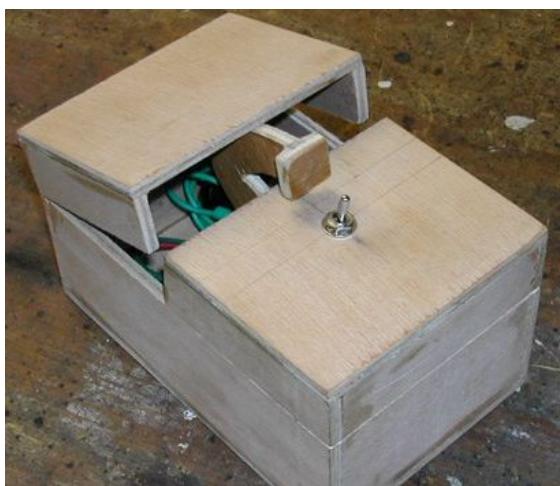


Abb. 1: Most Useless Machine von Marvin Minsky (1952)

Bekannter wurde die Useless Machine des MIT-Professors *Marvin Minsky* (1927–2016), einem der Urväter der Künstlichen Intelligenz, aus dem Jahr 1952, die nur eine

einige Funktion besaß: sich selbst ausschalten (Abb. 1 [1, 2]). Für einen theoretischen Informatiker ein faszinierendes mechanisches Gegenkonzept zum „Halteproblem“ – der Schwierigkeit algorithmisch nachzuweisen, dass ein Programm auch tatsächlich terminiert (endet).

Solche Useless Machines wurden von der fischertechnik-Community in zahlreichen Varianten konstruiert. Eine einfache, aber sehr kompakte Version der sich selbst ausschaltenden Maschine stelle ich hier vor – zugleich als Anregung, daraus eigene, komplexere Varianten abzuleiten.

Wer zunächst einen Eindruck von der Funktionsweise dieser sich selbst abschaltenden Maschine gewinnen möchte, der werfe einen Blick in das zugehörige kurze [Video](#). Für den Nachbau habe ich eine [fischertechnik-Designer-Datei](#) erstellt; sie kann zusammen mit einer [Einzelteilliste](#) von den Seiten der fischertechnik-AG heruntergeladen werden.

Konstruktion

Grundsätzlich gibt es (mindestens) zwei einfache mechanische Basis-Konstruktionen, mit denen der „Finger“ zum Ausschalten der Maschine mechanisch aus der „Kiste“ herausbewegt und wieder hinein-

„gezogen“ werden kann: ein Schubkurbelgetriebe oder eine Drehscheibe. Das Schubkurbelgetriebe benötigt zur Überbrückung des Endlagentasters (wie auch in der Konstruktion von [Kieseleck](#) aus dem Jahr 2011) lediglich einen Kippschalter; die Drehscheibe erfordert hingegen einen Polwendeschalter.

Die im Folgenden vorgestellte Maschine verwendet eine Drehscheibe, da sie damit besonders kompakt konstruiert werden kann: Sie passt – inklusive fischertechnik-Akku – in einen 90 x 90 x 90 mm³ großen Würfel, der sich mit Grund- und Statik-Platten verkleiden lässt. Die kompakte Mechanik aus Drehscheibe, Motor und Getriebe benötigt allerdings eine etwas raffiniertere Schaltung als die Schubkurbel.

Antrieb

Den Antrieb der Drehscheibe 60 übernimmt ein Mini-Motor mit U-Getriebe, der auf einer Grundplatte 90 x 90 montiert wird. An den Grundbausteinen, die das U-Getriebe halten, befestigen wir seitlich (über zwei BS 5 und einen BS 7,5) „hängend“ einen Mini-Taster (siehe Abb. 2, rechts unten) und direkt darüber einen Polwendeschalter. Das U-Getriebe erhält eine U-Achse 40 (mit Z28), auf die wir später die Drehscheibe setzen.

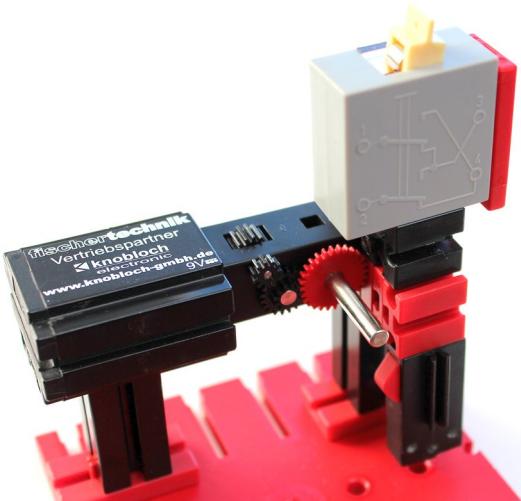


Abb. 2: Antrieb der Useless Machine mit Polwende- und Endlagenschalter

Ausschalt-Mechanik

Den „Ausschalt-Finger“ befestigen wir nun an einer Drehscheibe 60. Er ragt über die Drehscheibe hinaus, damit er später den vor dem U-Getriebe liegenden Polwendeschalter betätigen kann. In der Ruhestellung muss der „Finger“ den hängend montierten Endlagentaster betätigen und dabei komplett in der späteren Box verschwinden.

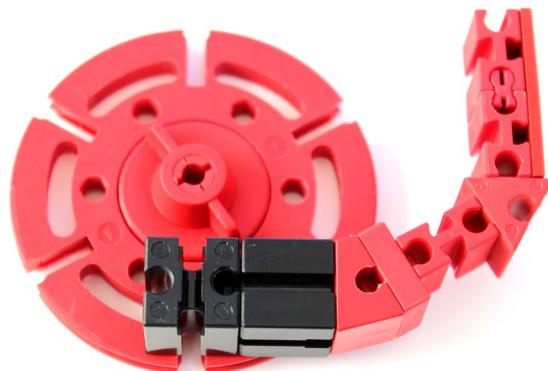


Abb. 3: Drehscheibe mit „Finger“ in Ruheposition

Abb. 3 zeigt die Konstruktion des „Fingers“ auf der Drehscheibe 60, die wir so am äußersten Ende der U-Achse aufsetzen, dass der „Finger“ den Mini-Motor gerade nicht berührt.

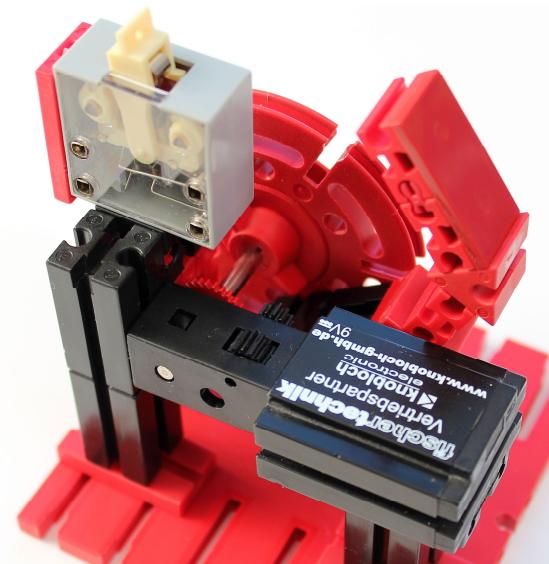


Abb. 4: Komplette Mechanik der Useless Machine

Die Nabennutter müssen wir sehr fest anziehen, damit die Drehscheibe stabil auf der

Achse sitzt und beim Drehen den Widerstand des Polwendeschalters überwindet ohne durchzurutschen. In Abb. 4 sieht man die komplette mechanische Konstruktion.

Schaltung

Der Anschluss des Polwendeschalters ist einfach: Er wird zwischen unsere Energiequelle (einen fischertechnik-Akku) und den S-Motor geschaltet und kehrt bei jeder Betätigung die Bewegungsrichtung des Motors (und damit der Drehscheibe) um.

Wo aber kommt der Endlagentaster hin? Er muss bei der „Rückdrehung“ des „Fingers“ (wenn dieser nach Betätigung des Polwendeschalters wieder in der Box verschwindet) den Stromkreis unterbrechen. Sobald aber die Stromrichtung durch manuelle Betätigung des Polwendeschalters umgekehrt wird, muss er überbrückt werden.

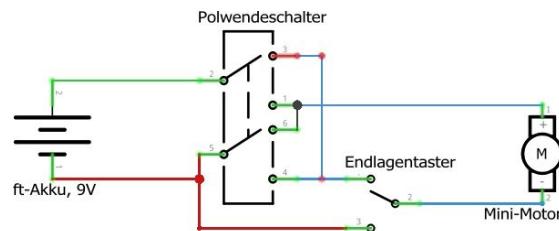


Abb. 5: Schaltbild der Useless Machine
(erstellt mit Fritzing)

Das gelingt, indem wir den Endlagentaster zwischen den einen Ausgang des Polwendeschalters und einen Motoreingang schalten, und zwar so, dass der Motoreingang bei Betätigung des Endlagentasters mit Masse verbunden wird (Abb. 5). Steht der Polwendeschalter nun so, dass der zweite Motoreingang mit +9V verbunden ist, läuft der Motor trotz betätigtem Endlagentaster. Ist der zweite Motoreingang über den Polwendeschalter hingegen mit Masse verbunden, stoppt der Motor bei betätigtem Endlagentaster. Solange der Endlagentaster nicht betätigt ist, dreht sich der Motor je nach Stellung des Polwendeschalters entweder in die eine oder in die andere Richtung.

Die Schaltung hat einen wichtigen Nebeneffekt: Dreht sich der Motor zurück und betätigt der „Finger“ den Endlagentaster, so wird der Motor kurzgeschlossen – und blockiert sofort [3].

Stromversorgung

Als Stromversorgung für unsere Useless Machine können wir natürlich ein fischertechnik-Netzteil verwenden – aber ein aus der Box herausgehängendes Kabel lässt sie weit weniger „autark“ wirken. Also montieren wir einen fischertechnik-Akku – der verbleibende Platz in unserer Box eignet sich exakt dafür, als hätten wir das genau so geplant (Abb. 6)...

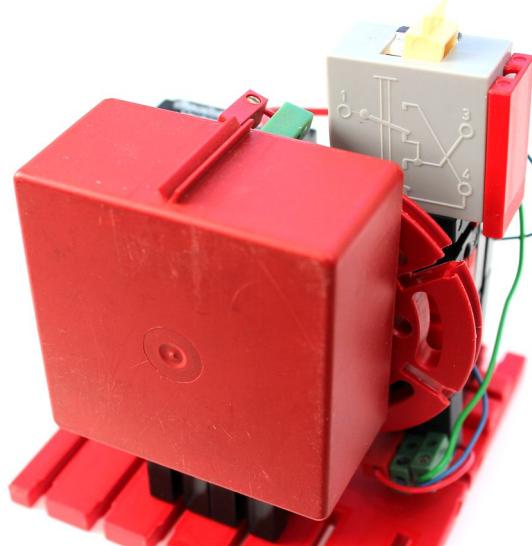


Abb. 6: Stromversorgung

Aufgesockelt mit zwei BS 15 und zwei über Kreuz montierten Bausteinen 15 x 30 x 5 mit Nut und Zapfen ragen die Akku-Anschlüsse, die wir mit dem Polwendeschalter verbinden müssen, gerade eben über die Drehscheibe 60 hinaus – und der „Finger“ der Maschine stoppt beim Drehen wenige Millimeter vor den beiden Steckern.

Wer keinen fischertechnik-Akku besitzt kann die Useless Machine alternativ mit einem Batteriehalter oder einem Batteriegehäuse mit 9V-Blockbatterie ausstatten.

angeschlossen an die Stromquelle (und bei richtiger Polung des Mini-Motors) funktioniert unsere Useless Machine bereits jetzt.

Verkleidung

Aber ohne eine geeignete „Verkleidung“ fehlt ihr der entscheidende „Pfiff“: Was wir eigentlich haben möchten ist eine geschlossene, $90 \times 90 \times 90 \text{ mm}^3$ große „Black Box“, aus der nur unser Polwendeschalter heraus schaut – und die sich nach dem Einschalten auf „magische Weise“ selbsttätig deaktiviert.

Das gelingt, indem wir in den Ecken der Grundplatte je zwei Statik-Winkelträger 60 und 30 montieren und daran an vier Seiten Statik-Platten 90×90 befestigen. Wer keine Statik-Platten besitzt kann sich auch mit V-Bauplatten oder Grundbausteinen und Bauplatten behelfen. In der Ecke, in die der Mini-Motor hereinreicht, dürfen wir nur zwei Statik-Winkelträger 30 zur Verbindung der Statik-Platten verwenden und müssen die Mitte aussparen (Abb. 7).



Abb. 7: Verkleidung der Box mit Statik-Platten, Aussparung für den Mini-Motor (rechts)

Deckel

Für den Deckel, der vom „Finger“ angehoben werden soll, ersetzen wir links und rechts neben dem „Finger“ den Statik-Winkelträger 30 durch einen Winkelträger 15 mit zwei BS 5 (Abb. 7, hinten links und rechts). Darauf setzen wir auf jeder Seite ein

Statik-Scharnier (36329), das je eine Ecke des Deckels halten wird.

Den Deckel wiederum setzen wir aus zwei Bauplatten 30×15 und zwei Bauplatten 15×60 zusammen, die wir über vier Bauplatten $15 \times 30 \times 3,75$ mit Nut stabil miteinander verbinden (Abb. 8).

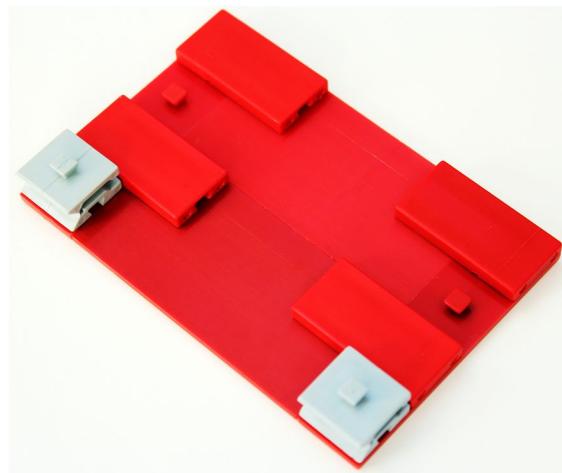


Abb. 8: Unterseite des Deckels

Mit zwei Bauplatten 30×45 , die wir mit je einem Zapfen in die Nuten der Enden der beiden vorderen Statik-Winkelträger schieben, schließen wir die Box von oben (Abb. 9).



Abb. 9: Gesamtansicht der Useless Machine

Als kleinen Nutzerhinweis können wir nun noch den Polwendeschalter mit einer Einschalt-Aufforderung beschriften:

EIN ►

Abb. 10: Beschriftung des Polwendeschalters

Hinweise zur Konstruktion

Der Platz in unserer kleinen Box ist ziemlich knapp bemessen, und die Konstruktion reagiert empfindlich auf Reibung (vor allem der Drehscheibe 60 am Akku oder der Verkleidung und der des „Fingers“ am Mini-Mot). Daher sollte man beim Aufbau der Box immer wieder kleine Tests vornehmen, ob der „Finger“ sich schnell und mit ausreichendem Drehmoment dreht.

Der Akku lässt sich ohne Ausbau laden: Es genügt, den Deckel abzunehmen, um das Ladekabel einzustecken. Bei der Verlegung der Kabel muss man allerdings darauf achten, dass sie sich nicht in den Nuten der Drehscheibe 60 verhaken können.

Falls sich die „Useless Machine“ starker Beanspruchung erfreut, sollte man den hängenden Endlagentaster mit einem quer montierten Baustein 15 x 30 x 5 mit Nut und Zapfen gegen Verrutschen sichern, da der „Finger“ trotz Kurzschließen der Stromversorgung mit großer Geschwindigkeit daran anschlägt (Abb. 11).

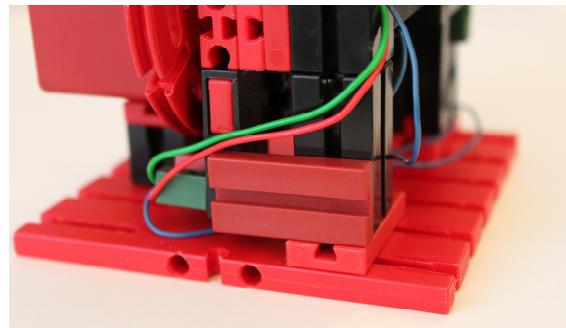


Abb. 11: Sicherung des Endlagentasters gegen Verrutschen

Weiterentwicklung

Das vorgestellte Modell ist eine sehr simple Useless Machine. Insbesondere von [freddy](#), aber auch von anderen Fans in der fischertechnik-Community wie [Hans Wijnsouw](#) gibt es raffinierte Weiterentwicklungen, die mehrere verschiedene Taster ausschalten können, mit einem Controller gesteuert werden und schnelle Pneumatik-Kolben als „Ausschalt-Finger“ verwenden. Der Fantasie sind bei der Umsetzung – wie so oft mit fischertechnik – fast keine Grenzen gesetzt.

Anregungen für die unterschiedlichsten Erweiterungen findet man auch mit einer Suche in [Youtube](#).

Quellen

- [1] Abigail Pesta: [Looking for Something Useful to Do With Your Time? Don't Try This.](#) The Wall Street Journal, 12.03.2013.
- [2] Video-Interview mit Marvin Minsky: [Making the most useless machine.](#)
- [3] Stefan Falk: [Motorsteuerungen \(Teil 1\)](#). ft:pedia 1/2011, S. 4-8.

Tipps & Tricks

Die Welt der fischertechnik-Winkelbausteine (Teil 2)

Rüdiger Riedel

Fortsetzung des Artikels in der ft:pedia 2017-1: Jenseits von Sinn und Nützlichkeit geht es mir hier um den Eindruck der Bilder; Ausgangspunkt ist eine einfache Systematik. Mir persönlich gefällt die Vielfalt an Formen und Gestalten, die sich hier ergibt.

Bausteine 15 als Winkelbausteine

Die hatte ich tatsächlich übersehen: Die „Winkelbausteine“ 90° !

Wieder beginnen wir mit einer Definition: Ergibt die Summe aller Winkel der verbauten Steine 360° , dann ist das ein geschlossener Ring, auch wenn die Figur verbeult und verbogen aussieht wie in Abb. 3, 4 und 5, und natürlich sind das immer Vielecke.

In den Beschreibungen benutze ich wieder folgende Darstellung:

- Winkelstein $7,5^\circ \rightarrow WS7,5$
- Winkelstein $15^\circ \rightarrow WS15$
- Winkelstein $30^\circ \rightarrow WS30$
- Winkelstein $60^\circ \rightarrow WS60$

und neu:

- Winkelstein $90^\circ \rightarrow WS90$



Abb. 1: Ein Winkelstein 90° (WS90)

Wie in Abb. 1 gezeigt, stellen wir uns einen halbierten Baustein 15 vor. Zeigt dessen 90° -Spitze in das Ringinnere, dann wird der Winkel 90° positiv gezählt, zeigt er nach außen, dann wird der Winkel 90° negativ gezählt [1].



Abb. 2: Zweimal 4xWS90

Für den WS90 haben wir mehrere Kandidaten:

- Die verschiedenfarbigen BS15 (es gehen natürlich auch senkrecht gestellte BS30).
- Baustein 15 Ansenkung ([32321](#)).
- Baustein 15 Bohrung ([32064](#))
- V-Baustein 15 Eck ([38240](#))
- Winkelstein 10x15x15 ([38423](#))
- S-Riegelstein 15x15 ([32850](#))
- Gelenkwürfel-Zunge $7,5$ ([31426](#)) mit Gelenkwürfel-Klaue $7,5$ ([31436](#))
- S-Scharnier ([36329](#))

Die letzten beiden sind natürlich nur mit Einschränkung zu gebrauchen; wegen der Beweglichkeit kann sich der Winkel verändern.

Je nach Bausteinen können Federnocken ([31982](#)) erforderlich sein.

Zweizählige Ringe

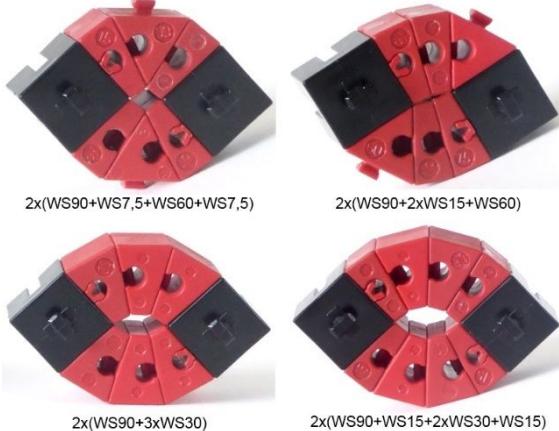


Abb. 3: Verschiedene zweizählige Ringe



Abb. 4: Zweizählige Ringe mit unterschiedlichen WS90

Durch Änderung der Reihung können weitere zweizählige Ringe erzeugt werden. Hier sind einige dargestellt:



Abb. 5: Weitere zweizählige Ringe

Dreizählige Ringe

Es gibt nur 4 dreizählige Ringe.

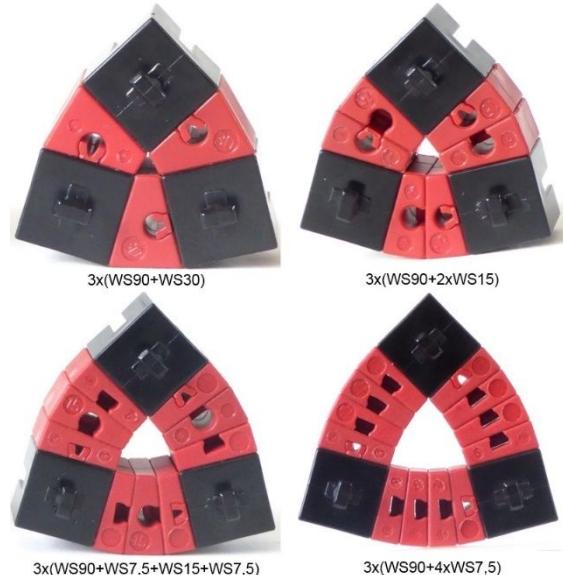


Abb. 6: Die dreizählichen Ringe



Abb. 7: Variante

Der Ring in Abb. 7 hat die gleichen Steine wie derjenige links unten in Abb. 6, jedoch mit anderer Reihung.

Negative Winkel

Baut man die echten Winkelsteine verkehrt herum ein, also mit der breiten Seite ins Innere der Ringe und sorgt trotzdem für eine Winkelsumme der Ringe von 360° [1], dann ergibt sich eine Fülle weiterer Möglichkeiten. Die WS90 sollen zunächst wie in Abb. 1 eingebaut werden, also mit der (virtuellen) breiten Seite nach außen.

Sechszählige Ringe

Es gibt jetzt vier Ringe mit sechszähliger Symmetrie:



Abb. 8: Sechszählige Ringe

Ändert man die Reihung des Ringes links unten in Abb. 8, dann erhält man den Ring Abb. 9:



Abb. 9: Variante

Wir können auch die sechszählige Symmetrie durchbrechen und z. B. aus dem Ring rechts unten in Abb. 8 solche wie in Abb. 10 und 11 bilden:

Abb. 10: $2 \cdot (WS90 - 2 \cdot WS7,5 + WS90 - 4 \cdot WS7,5 + WS90 - 6 \cdot WS7,5)$ Abb. 11: $2 \cdot (WS90 - 6 \cdot WS7,5 + WS90 - 6 \cdot WS7,5 + WS90)$

Achtzählige Ringe

Von achtfach symmetrischen Ringen finden wir 6 Stück, wenn wir die Reihung der Steine nicht variieren. Vier davon sind in den Abb. 12 bis 15 dargestellt, die vollständige Liste findet ihr in [2].



Abb. 12: $8 \cdot (WS90 - WS30 - WS15)$



Abb. 14: $8 \cdot (WS90 - 3 \cdot WS15)$



Abb. 13: $8 \cdot (WS90 - WS7,5 - WS30 - WS7,5)$

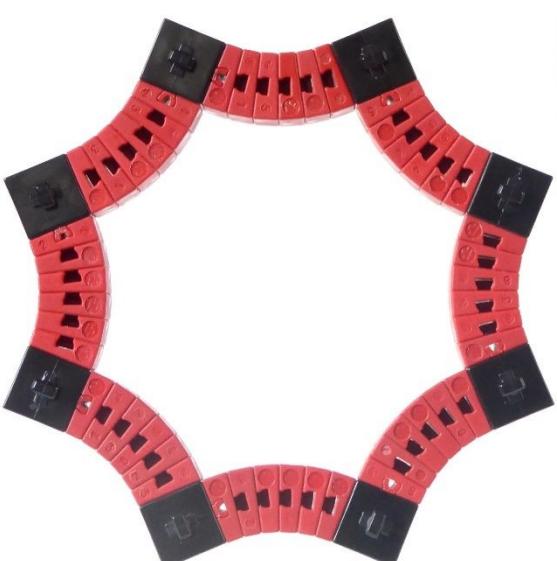


Abb. 15: $8 \cdot (WS90 - 6 \cdot WS7,5)$

Sie sehen aus wie achtfach symmetrisch, aber sie sind es nicht (Abb. 16):



Abb. 16: Vierfache Symmetrie

Zwölfzählige Ringe

Entsprechend der bisherigen Definition gibt es 10 Ringe mit zwölfzähliger Symmetrie.



Abb. 17: $12 \cdot (WS90 - WS60)$

Abb. 18 zeigt eine Variation von Abb. 17:

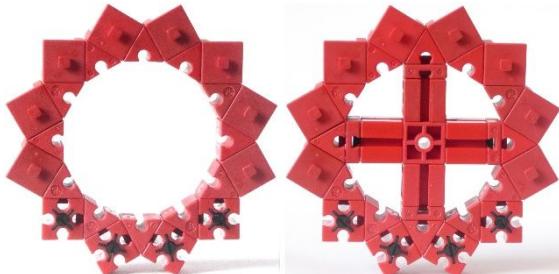


Abb. 18: WS90 geändert und Speichen mit Nabe hinzugefügt



Abb. 19: $12 \cdot (WS90 - 2 \cdot WS30)$

Sechzehnzählige Ringe

Sechzehnfache Symmetrie habe ich ebenfalls zehnmal gefunden. Drei Beispiele sind nachfolgend dargestellt:



Abb. 20: $16 \cdot (WS90 - WS60 - WS7,5)$



Abb. 21: $16 \cdot (WS90 - WS30 - 2 \cdot WS15 - WS7,5)$



Abb. 22: $16 \cdot (WS90 - WS30 - WS7,5 - WS30)$

24-zählige Ringe

Es folgen 24-fache Ringe:

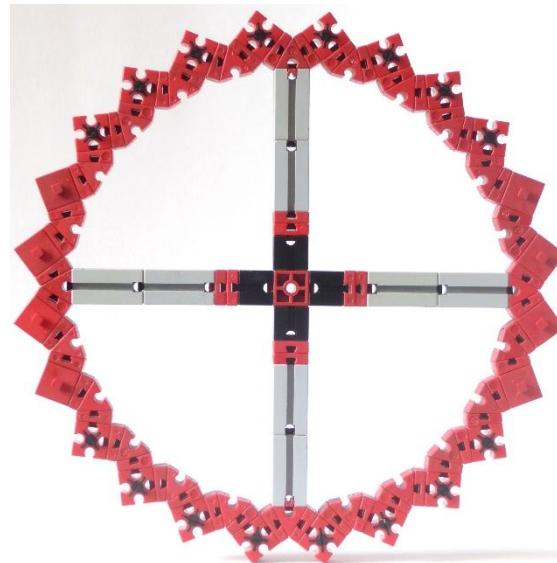


Abb. 23: $24 \cdot (WS90 - WS7,5 - WS60 - WS7,5)$
mit Speichen und Nabe



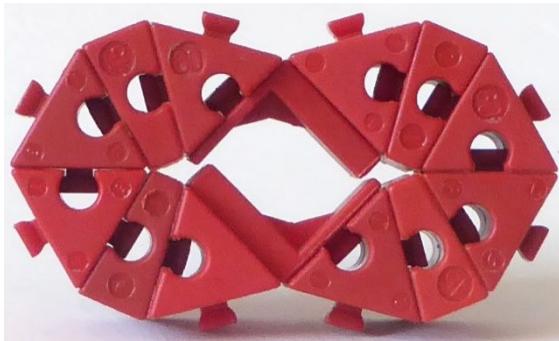
Abb. 24: $24 \cdot (WS90 - WS60 - WS15)$

WS90 mit negativem Winkel

Auch der WS90 kann mit negativem Winkel in die Ringberechnung eingehen.



*Abb. 25: Zwei Varianten von
2 · (4 · WS60 + WS30 - WS90)*



*Abb. 26: 2 · (WS60 + WS15 + 2 · WS60 +
WS15 + WS60 - WS90)*

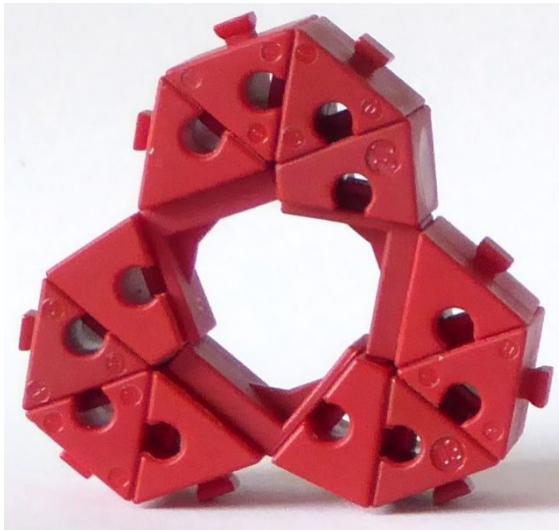


Abb. 27: 3 · (3 · WS60 + WS30 - WS90)

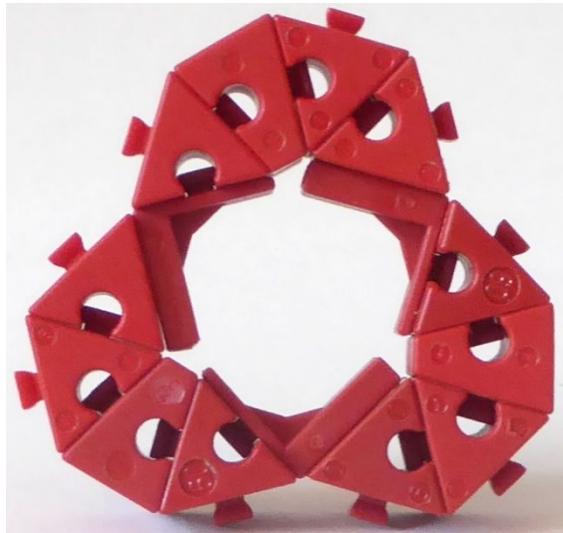
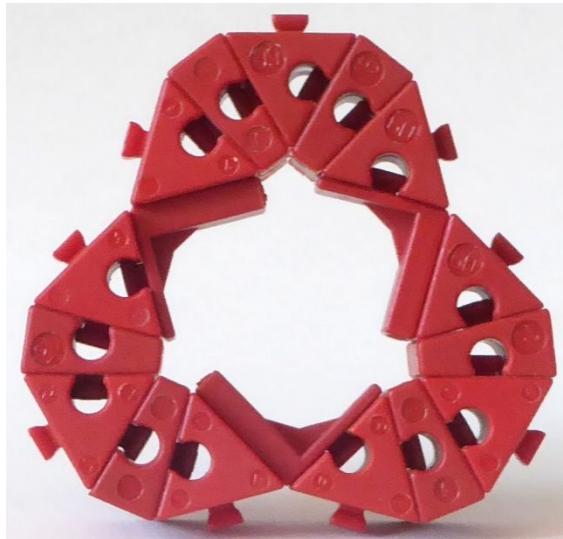
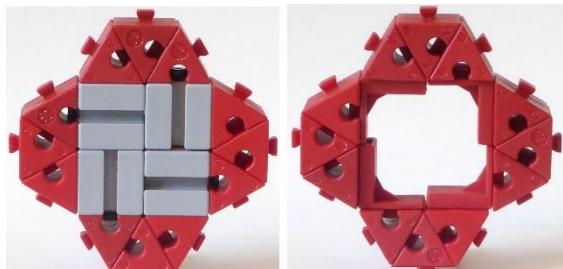


Abb. 28: 3 · (WS60+WS30+2xWS60-WS90)



*Abb. 29: 3 · (WS60 + WS15 + WS60 + WS15
+ WS60 - WS90)*



*Abb. 30: Zwei Varianten von
4 · (3 · WS60 - WS90)*

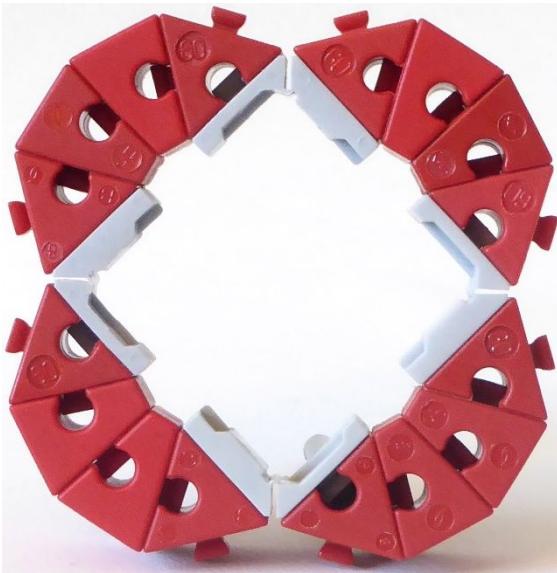


Abb. 31: $4 \cdot (WS60 + 2 \cdot WS30 + WS60 - WS90)$

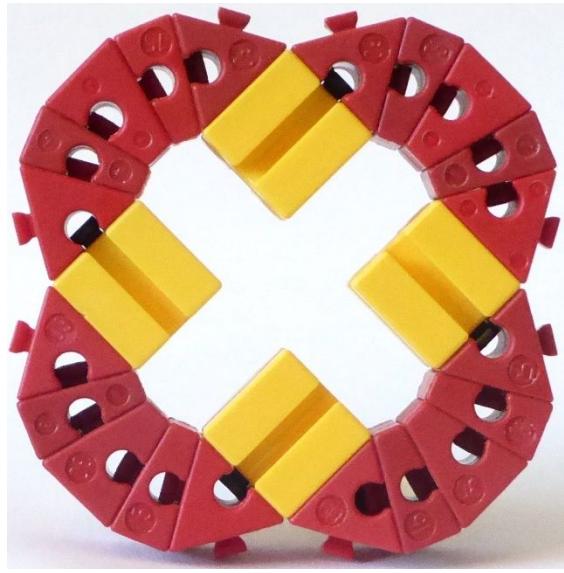


Abb. 33: $4 \cdot (WS60 + WS15 + WS30 + WS15 + WS60 - WS90)$

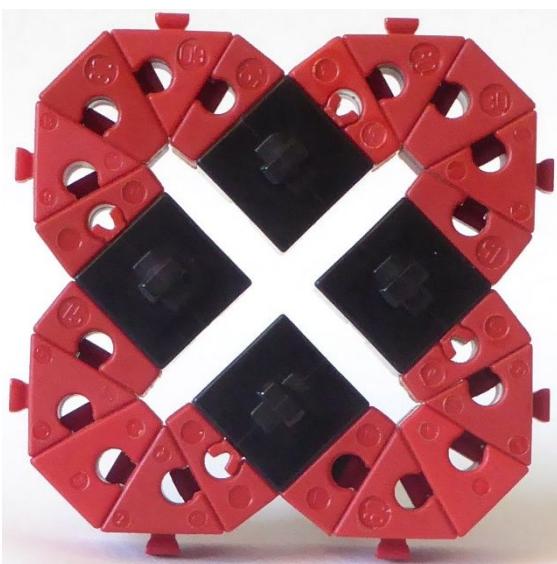


Abb. 32: $4 \cdot (WS15 + WS60 + WS30 + WS60 + WS15 - WS90)$



Abb. 34: $6 \cdot (WS60 + WS30 + WS60 - WS90)$



Abb. 35: $8 \cdot (WS60 + WS15 + WS60 - WS90)$



Abb. 37: $12 \cdot (WS30 + WS60 + WS30 - WS90)$



Abb. 36: $12 \cdot (2 \cdot WS60 - WS90)$



Abb. 38: $16 \cdot (WS60 - WS7,5 + WS60 - WS90)$



Abb. 39: $16 \cdot (WS30 + WS15 + 3 \cdot WS7,5 + WS15 + WS30 - WS90)$

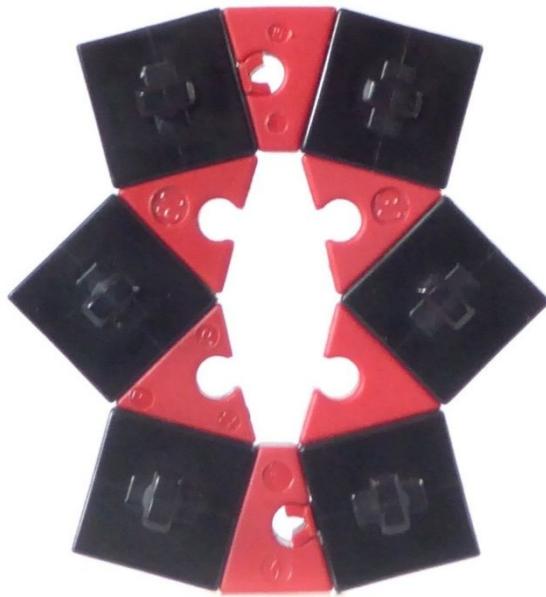


Abb. 42: $2 \cdot (WS90 - WS60 + WS90 - WS60 + WS90 + WS30)$

Gemischt



Abb. 40: $1 \cdot (3 \cdot WS60 + 2 \cdot WS90)$ und $1 \cdot (2 \cdot WS60 + WS90 + WS60 + WS90)$



Abb. 41: $2 \cdot (WS90 - WS60 + WS90 - WS60 + WS90 + WS60 - WS90 + WS60)$

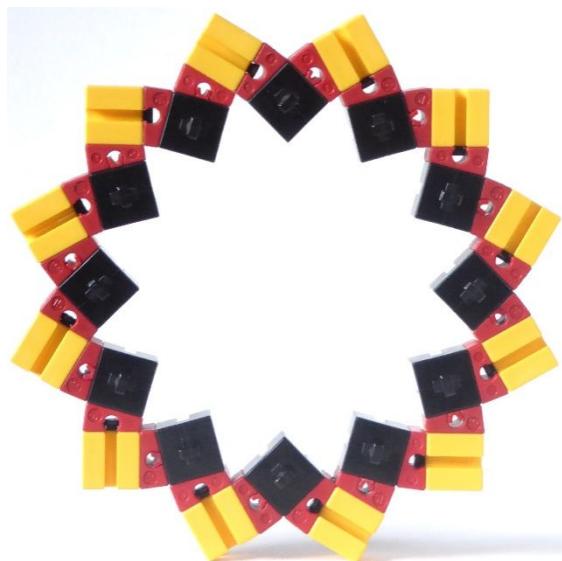


Abb. 43: $12 \cdot (WS90 + WS15 - WS90 + WS15)$

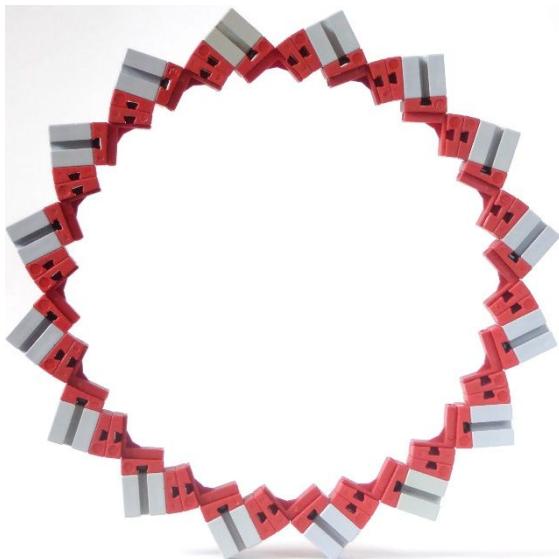


Abb. 44: $16 \cdot (WS90 + WS7,5 - WS90 + 2 \cdot WS7,5)$



Abb. 46: Traumfarben

Zum Abschluss ein Traum

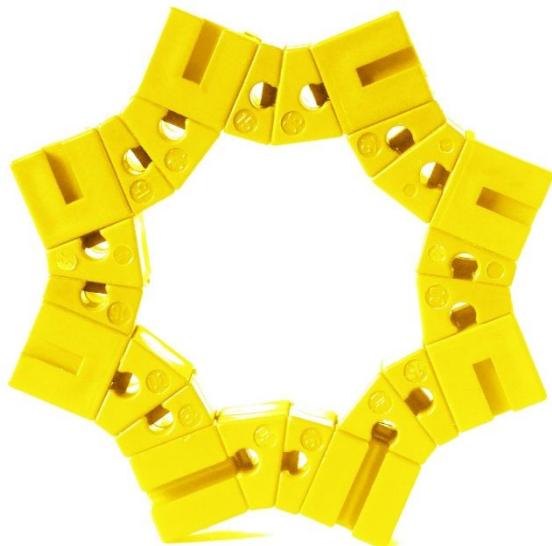


Abb. 45: Mein Traum: Winkelsteine in Grün, Gelb oder Blau oder... wenigstens Grau



Abb. 47: Grau hat etwas für sich

Quellen

- [1] Rüdiger Riedel: [Die Welt der fischertechnik-Winkelbausteine.](#)
ft:pedia 1/2017, S. 17-23.
- [2] Rüdiger Riedel: [Kreisringe aus ft Winkelbausteinen Teil 2.xlsx.](#)
Downloadbereich der ft-Community, 2018.

Computing

Einbindung des TXT- bzw. TX-Pi-Touchscreens in startIDE

Peter Habermehl

Stay in touch!

Einleitung

Mit der kürzlich veröffentlichten Version 1.5 hat die Community Firmware App „startIDE“ Funktionen zur Abfrage von Berührungs-Ereignissen auf dem Display von TXT bzw. TX-Pi erhalten. Diese neuen Funktionen sollen im Folgenden kurz vorgestellt werden.

Die Funktionen

Alle Touch-Funktionen stehen sinnvollerweise nur bei eingeblendetem Grafik-Leinwand zur Verfügung, da nur der Grafik-Zeichenbereich von ihnen erfasst wird.

Sie sind in der Funktionsgruppe „Interaktion → Touch“ der startIDE-Befehlsliste zu finden. Zur Auswahl stehen die Funktionen „WaitForTouch“, „WaitForRelease“ und „IfTouchArea“. Weiterhin stehen zur Auswertung die Systemvariablen touchXPos und touchYPos zur Verfügung, die mit der bekannten Funktion „FromSys“ ausgelesen werden können.

„WaitForTouch“ wartet, wie der Name nahelegt, auf die Berührung des Touchscreens durch den Benutzer. Die Funktion hat keine Parameter; der Programmablauf wird so lange angehalten, bis ein Antippen des Bildschirms (bzw. des Grafikbereichs) erfolgt. Analog dazu wartet „WaitForRelease“ auf das Wiederloslassen des Bildschirms.

Nach einem Berührungsevent stellen die Systemvariablen „touchXPos“ und „touchYPos“ die Koordinaten bereit, an denen das Berühren bzw. Loslassen erfolgte.

Um festzustellen, ob die Berührung innerhalb eines definierten rechteckigen Bereiches stattgefunden hat, gibt es schließlich die Funktion „IfTouchArea <x1> <y1> <x2> <y2> <Sprungziel>“, siehe Abb. 1.

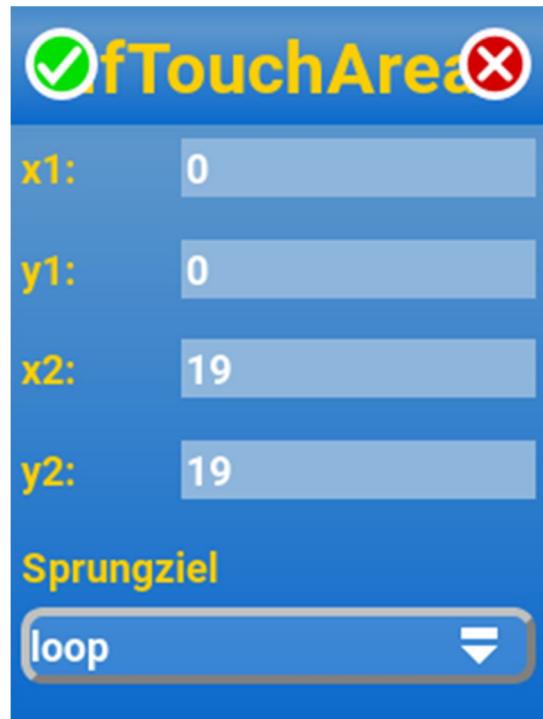


Abb. 1: Kommando IfTouchArea

Lag die Berührung innerhalb des durch die Koordinaten (x1, y1) und (x2, y2) aufgespannten Rechtecks, so wird mit der Programmausführung an der mit „Tag

<Sprungziel>“ gesetzten Marke fortgefahrene, andernfalls an der auf IfTouchArea folgenden Programmzeile.

Ein ganz simples Beispiel

Nach dem Einblenden der Grafik-Leinwand soll auf ein Antippen derselben gewartet werden. Danach soll die Leinwand wieder ausgeblendet und die Koordinaten der Berührung ausgegeben werden (Listing 1).

```

01 # Touch demo 1
02 Init x 0
03 Init y 0
04 #
05 Canvas show
06 Canvas clear
07 WaitForTouch
08 Canvas hide
09 #
10 FromSys x touchXPos
11 FromSys y touchYPos
12 #
13 QueryVar x
14 QueryVar y

```

Listing 1: Warten auf eine Bildschirmberührung

In den Zeilen 2 und 3 werden die Variablen für die x- und y-Koordinate initialisiert, in den Zeilen 5 und 6 die Leinwand eingeblendet und gelöscht, in Zeile 7 wird dann auf das Antippen gewartet. Nach dem Berühren des Bildschirms wird in Zeile 8 die Leinwand wieder ausgeblendet.

In den Zeilen 10 und 11 werden die Koordinaten der Berührung ermittelt und schließlich in den Zeilen 13 und 14 auf dem Bildschirm ausgegeben.

Die einfachste Anwendung von WaitForTouch wäre also z. B. das Warten auf eine Bildschirmberührung nach einer Grafikausgabe, um anschließend die Leinwand auszublenden und mit dem Programmablauf fortzufahren.

Ein reaktiver Button

Eine weitere, recht einfache Anwendung ist die Darstellung eines Startknopfes auf dem Display, mit dem eine Aktion ausgelöst werden kann.

In diesem Fall soll, während der Button betätigt und gehalten wird, der Ausgang O1 des TXT eingeschaltet werden, an dem z. B. eine Lampe oder ein Motor angeschlossen sein kann. Bei Nichtbetätigung soll der Ausgang ausgeschaltet sein. Ein zweiter, kleinerer Button in der oberen, linken Ecke des Displays soll zum Beenden des Programms dienen.

Das so erzeugte grafische User Interface (GUI) ist in Abb. 2 zu sehen, das dazugehörige Programm in Listing 2.



Abb. 2: User Interface aus Listing 2

```

01 # Touch demo 3
02 Canvas show
03 Canvas clear
04 #
05 Call background 1
06 Tag top
07 Call foreground_0 1
08 Call decoration_1
09 Canvas update
10 #
11 WaitForTouch
12 IfTouchArea 0 0 30 30 end
13 Output TXT 1 512
14 #
15 Call foreground_1 1
16 Call decoration_1

```

```

17 Canvas update
18 #
19 WaitForRelease
20 Output TXT 1 0
21 Jump top
22 #
23 Tag end
24 Stop
25 #
26 Module background
27 Color pen 255 255 255
28 Pen move 60 60
29 Pen discTo 180 180
30 Pen move 0 0
31 Pen discTo 30 30
32 Color pen 255 0 0
33 Pen move 4 4
34 Pen discTo 26 26
35 Color pen 255 255 255
36 Text Courier 20 X
37 Pen text 6 24
38 Pen text 7 24
39 Pen text 8 24
40 MEnd
41 #
42 Module foreground_0
43 Color pen 0 127 0
44 Pen move 66 66
45 Pen discTo 174 174
46 MEnd
47 #
48 Module foreground_1
49 Color pen 0 255 0
50 Pen move 66 66
51 Pen discTo 174 174
52 MEnd
53 #
54 Module decoration
55 Color pen 255 255 255
56 Pen move 117 90
57 Pen boxTo 123 150
58 MEnd

```

Listing 2: Reaktiver Button

Das Listing zu diesem Beispiel ist mit fast 60 Zeilen nun schon etwas länger, aber der größere Teil des Programmes besteht aus den Grafikbefehlen zum Zeichnen des User Interfaces, so dass die Funktion recht leicht verständlich ist.

Wie schon mehrfach gezeigt wird in den Zeilen 2 und 3 die Leinwand eingeblendet und gelöscht.

Das in Zeile 5 aufgerufene Modul *background*, Zeilen 26-40, zeichnet nun einen

ausgefüllten weißen Kreis mit 120 Bildpunkten Durchmesser mittig auf dem Bildschirm.

Anschließend wird das „Ende“-Symbol gezeichnet. Es besteht aus einem kleinen, weißen Kreis oben links, darüber dann einem nochmals kleineren roten Kreis und in diesen wird schließlich, drei Mal um jeweils einen Pixel versetzt, der Text „X“ geschrieben.

Das Modul *foreground_0* zeichnet nun einen dunkelgrünen Kreis mittig in den bereits vorhandenen weißen Kreis, und das Modul *decoration* dann noch den senkrechten weißen Strich mittig in den Button. Der Aufruf der drei Zeichenmodule wird mit *Canvas update* zum Auffrischen des Bildschirminhalts in Zeile 9 abgeschlossen.

Damit ist die grafische Benutzerschnittstelle – unsere zwei Buttons – auf dem Bildschirm dargestellt, und es wird in Zeile 11 auf eine Berührung des Bildschirms gewartet.

Wurde der Bildschirm angetippt, wird in Zeile 11 per „IfTouchArea“ geprüft, ob die Berührung in der oberen linken Ecke, also in dem Rechteck innerhalb der Koordinaten (0, 0) und (30, 30) stattfand. Ist dies der Fall, wird zu Zeile 23 gesprungen und damit das Programm in Zeile 24 beendet.

Ist die Berührung hingegen außerhalb dieses Bereiches erfolgt, so wird mit Zeile 13 fortgefahren und zunächst Ausgang O1 des TXT auf volle Leistung geschaltet. Eine daran angeschlossene Lampe leuchtet jetzt.

Danach werden in den Zeilen 15 und 16 die Module *foreground_1* und *decoration* aufgerufen. Dadurch wird das Innere des Startbuttons hellgrün dargestellt. Anschließend wird in Zeile 19 auf ein Loslassen des Startbuttons gewartet.

Nachdem dies erkannt wurde, wird umgehend der Ausgang O1 des TXT wieder abgeschaltet – Zeile 20 – und zur Sprungmarke „top“ gesprungen. Dort wird der

Startbutton wieder in dunkelgrün umgefärbt und es wird von Neuem auf eine Bildschirmberührungssteuerung gewartet.

Ausblick

Auch wenn die Erstellung einer grafischen Benutzerschnittstelle mit startIDE recht aufwändig ist, so bieten sich hier doch neue Möglichkeiten.



Abb. 3: Bedienoberfläche für Greif-Roboter

So könnte man z. B. über eine Bedienoberfläche, wie in Abb. 3 skizziert, einen Säulenroboter mit Greifer steuern. Oder programmiert jemand Tic Tac Toe auf dem TXT in startIDE?

Die Entwicklung von startIDE selbst wird zunächst in eine andere Richtung weitergehen – gäbe es Feldvariablen, so könnte man z. B. mit dem skizzierten Roboter User Interface eine Liste von Positionen abspeichern, die dann wieder angefahren werden können... stay in touch for v1.6!

Quellen

- [1] Peter Habermehl: [startIDE für die Community Firmware – Programmieren direkt auf dem TXT oder dem TX-Pi](#). ft:pedia 01/2018, S. 102-107.
- [2] Rolf Meingast: [startIDE \(1-4\)](#). ft:pedia 1/2018, S. 108-121.
- [3] Peter Habermehl: [Grafik auf dem TXT mit startIDE](#). ft:pedia 3/2018, S. 37-40.
- [4] Peter Habermehl: [Handbuch für startIDE](#). v1.5, GitHub, 2018.
- [5] [startIDE-Thread](#) im ftCommunity Forum.
- [6] Till Harbaum: [TX-Pi, der fischertechnik Raspberry Pi](#). GitHub, 2018.

Messen

startIDE (5): Temperaturmessung an der Community-Akku-Box

Peter Habermehl

Versuch macht kluch...¹

Im Lauf der Entwicklung der community Akku-Box [1] trat bei einigen Testern das Problem auf, dass sich die Akku-Box beim Laden so stark erwärmte, dass sie sich bis zur Unbrauchbarkeit verformte. Daraus resultierte die Empfehlung, erstens die Akkuzellen nicht in der Box zu laden und zweitens als Druckmaterial für das Gehäuse ABS zu verwenden.

Doch ich wollte der Sache auf den Grund gehen und führte daher eine kleine Versuchsreihe durch, um das Temperaturverhalten der Akkus zu untersuchen. Dabei erhebe ich keinen Anspruch auf Vollständigkeit oder Allgemeingültigkeit der Ergebnisse.

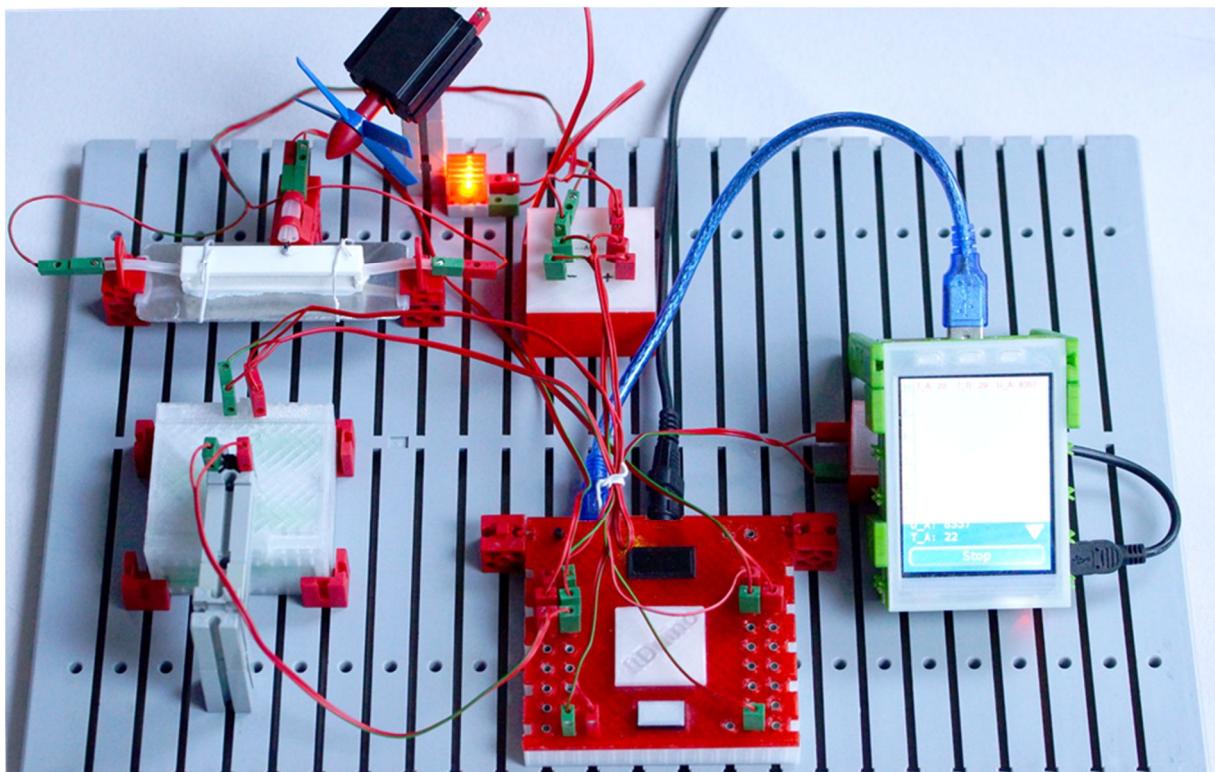


Abb. 1: Versuchsaufbau

¹ Ich bitte von Hinweisen auf die korrekte Orthographie des Wortes „klug“ abzusehen; die inkorrekte Schreibweise wird hier als erheiterndes Stilelement im Rahmen eines relativ trockenen Berichts verwendet.

Versuchsaufbau

Abb. 1 zeigt den Versuchsaufbau. Über ein Relais wird der Ausgang der Akku-Box mit einem 12-Ohm-Widerstand verbunden.

Der Deckel der Box ist mit einem Temperatursensor NTC 10k versehen, dessen Kontakte nach außen geführt wurden (Abb. 2). Dieser, sowie ein zweiter NTC 10k, der am Lastwiderstand angebracht ist, werden während des Tests überwacht. Bei Überschreitung von 50°C am Lastwiderstand wird ein Kühlgebläse zugeschaltet.

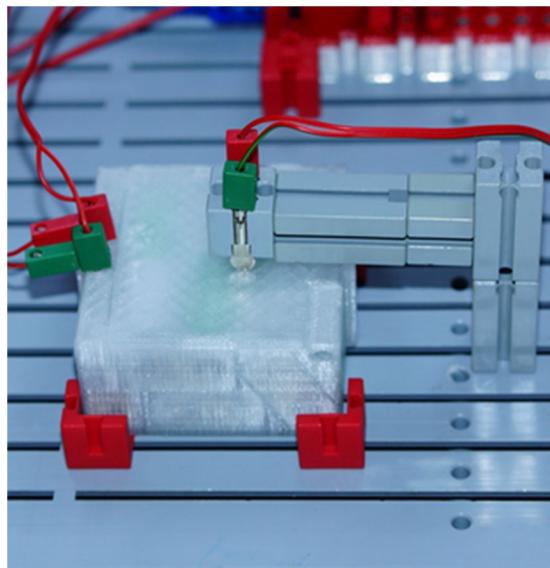


Abb. 2: Akku-Box mit NTC 10k

Die Lade- und Entladesteuerung obliegt einer Kombination aus TX-Pi, ftDuino und startIDE.

Zunächst wird der Testakku definiert auf 6300 mV (gemessen unter Last) über den 12-Ohm-Widerstand entladen. Das entspricht einer Entladeschlussspannung von 0,9V/Zelle. Dabei beträgt der rechnerische Entladestrom zu Beginn, bei 8,4V, ca. 0,7A, bei Entladeende (6,3V) ca. 0,53A, was gemessen an den 0,1A Stromaufnahme einer einzelnen fischertechnik-Glühlampe einer durchaus moderaten Belastung entspricht.



Abb. 3: Überwachung des Entladevorgangs mit startIDE

Abb. 3 zeigt die grafische Darstellung des Entladevorganges in startIDE auf dem TX-Pi. Da die x-Achse 55 Minuten in 15-Sekunden-Auflösung darstellen kann, wird alle 55 Minuten und am Ende des Entladevorganges (Abb. 4) ein Screenshot abgespeichert, sodass der gesamte Entladevorgang dokumentiert ist.

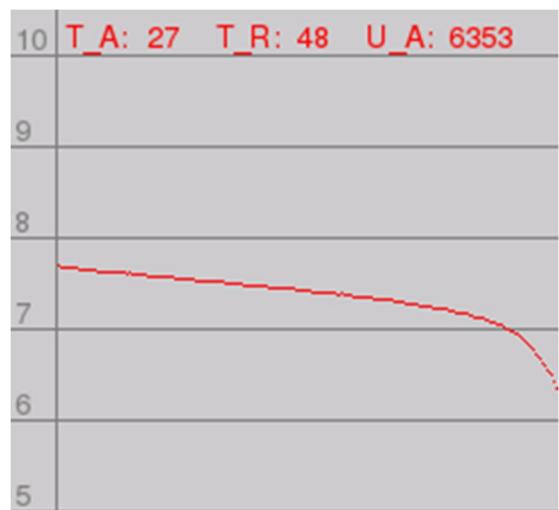


Abb. 4: Entladeschluss: startIDE-Grafik-Log

Angemerkt sei, dass während aller Entladevorgänge die Temperatur in der Akku-Box im Bereich von 25 bis 27°C lag. Die Umgebungstemperatur betrug bei allen Versuchen 21°C.



Abb. 5: Verwendetes ft-Akku-Ladegerät

Nach einer Ruhezeit von einer Stunde wird statt des Widerstands ein ft-Akku-Ladegerät angeschlossen, die Adaptierung ist in Abb. 5 dargestellt. Dadurch kann der Ladevorgang über das Relais gesteuert werden.

Während des Ladevorgangs wird lediglich die Temperatur in der Akku-Box gemessen. Bei Überschreitung von 55°C würde der Ladevorgang unterbrochen und die Box durch das Relais vom Ladegerät getrennt. Geladen wurde jeweils bis zum Erreichen des durch das Ladegerät indizierten Ladeendes.

Für diese Versuchsreihe wurde die Akku-Box mit drei verschiedenen Akkusätzen getestet:

- Satz 1, forthin als „Neu“ referenziert, ist mit sieben neuen IKEA Ladda 2000 mAh Zellen bestückt.

- Satz 2, als „Gebraucht“ referenziert, ist mit sieben Eneloop-Zellen des Typs „HR-3UTGA“, 1900 mAh bestückt, die gemeinsam ca. 200 Lade/Entladzyklen erfahren haben.
- Satz 3, als „Gemischt“ bezeichnet, besteht aus fünf Zellen des „Gebraucht“-Typs sowie zwei Zellen, die ein unbekanntes Vorleben haben und deutlichen Kapazitätsverlust aufweisen bzw. von einem Ladegerät bereits als defekt erkannt wurden.

Mit dem neuen Akku-Satz wurden drei Entlade-Lade-Zyklen beobachtet, mit dem gebrauchten und gemischten Satz jeweils zwei. Aufgrund der geringen Stichprobengröße wird auf weitere statistische Betrachtungen verzichtet.

Die Ergebnisse sind in der untenstehenden Tabelle (Tab. 1) dokumentiert.

Die Ergebnisse decken sich sehr schön mit den bereits bekannten Beobachtungen. Die Temperaturrentwicklung bei neuen und gebrauchten, intakten Zellen hält sich in Grenzen, die Abweichung liegt in einem Streubereich, der nicht nennenswert größer als die mit dem vorliegenden System zu erreichende Messgenauigkeit zu sein scheint.

Der mit quasi-defekten Zellen kontaminierte gemischte Zellsatz zeigt eine deutliche Tendenz hin zu höheren Temperaturen, auch wenn die hier gemessenen maximalen Temperaturen mit 43 bzw. 46°C für die Stabilität des Gehäuses nicht kritisch waren. Es ist ohne weiteres vorstellbar, dass mit anderen Zellen, die womöglich auch untereinander starke Unterschiede aufweisen, und anderen Ladegeräten weitaus höhere Temperaturen auftreten können, so wie im Diskussions-Thread des ftc-Forums berichtet [2].

Zellsatz	neu	neu	neu	gebraucht	gebraucht	gemischt	gemischt
T max [°C]	29	31	28	34	31	43	46

Tab. 1: Ergebnisse Temperaturmessung Akku-Ladegerät

Ich möchte nach dieser Testreihe schon aus Haftungsgründen keine Empfehlung für die Handhabung der ftc-Akkubox aussprechen.

Da ich die bei uns in Betrieb befindlichen Boxen, die auch unbeaufsichtigt in Kinderhand sind, jeweils mit neuen, gleichartigen Zellen bestückt habe und die ersten Ladevorgänge, die beaufsichtigt stattfanden, absolut unproblematisch verliefen, ziehe ich für mich selbst aber das Fazit, dass gepflegte Zellensätze durchaus risikoarm in der Box geladen werden können. Dies sollte jedoch nicht völlig unkontrolliert geschehen; ein Überprüfen der Temperatur durch gelegentliches Handauflegen während des Ladevorganges sollte jedoch ausreichend sein, um eine Verschlechterung des Zellzustands rechtzeitig zu bemerken.

Abschließend noch ein Wort zum Druckmaterial für die Boxen. Da ABS, das deutlich höhere Temperaturstabilität aufweist als PLA, relativ schwierig zu verarbeiten ist und dabei auch gesundheitliche Bedenken bezüglich der Ausdünstungen während des Druckes bestehen, habe ich die

abgebildete Box mit NTC aus PETG gedruckt. Hierzu waren keinerlei Veränderungen an der Konstruktion erforderlich, lediglich die Materialparameter, insbesondere die Temperaturen, wurden für den Druck angepasst.

Auch das sogenannte PLA-plus, das deutlich verbesserte mechanische und Temperatureigenschaften gegenüber PLA aufweist, sollte eine interessante Alternative sein.

Bei uns sind außer einer ABS-Box nichtsdestotrotz mehrere PLA-Boxen im Einsatz, ohne dass es bislang zu Auffälligkeiten gekommen wäre.

Quellen

- [1] Die [ftc-Akkubox](#) bei Thingiverse
- [2] [Diskussions-Thread](#) im ftc-Forum
- [3] [Einzelteilbestellung](#) der ftc-Akkubox bei fischerfriendsman

Messen

Kurz gemessen – Experimente im Mikrosekundenbereich mit dem ftDuino

Till Harbaum

Der ftDuino [1] ist technisch viel einfacher gestrickt, er taktet und rechnet langsamer als der TXT und erst recht als jeder PC. Trotzdem gibt es viele Dinge, die nur der ftDuino schnell genug erledigt. Warum das so ist und wofür sich das nutzen lässt zeigen wir in diesem Beitrag.

Hintergrund

Je höher der Mikroprozessor eines Computers getaktet ist, desto schneller reagiert der Computer. Das kennt man vom heimischen Windows-PC. Wenn's da ruckelt muss ein schnellerer Prozessor her, dann läuft es wieder flüssig. Und wenn der PC mit seinen Gigahertz zuckelt und ein TXT mit 600 MHz gemächlich zu Werke geht, was kann man da wohl von einem ftDuino mit seinen 16 MHz erwarten?

Ganz so einfach ist es aber nicht. Auf dem TXT (und erst recht auf dem PC) treiben diverse Systemprogramme, Treiber und Betriebssystemkomponenten im Hintergrund ihr Unwesen. Ständig will irgendeine Komponente die Aufmerksamkeit des Prozessors, und der muss seine Hauptaufgabe immer wieder unterbrechen und sich für ein paar Millisekunden irgendeiner Nebensache widmen, schließlich soll ja z. B. das WLAN auch dann funktionieren, wenn der TXT gerade einen Roboter steuert.

All das gibt es auf ftDinos (und den artverwandten Arduinos) nicht. Dort stört kein Betriebssystem, und es passiert neben dem Sketch des Nutzers kaum etwas. Ein paar Routinen erledigen im Hintergrund die USB-Kommunikation, und je nach Nutzung werden auch die Ein- und Ausgänge des ftDuino im Hintergrund bedient. Aber im

Vergleich mit dem PC oder dem TXT sind dies kleine Nebensächlichkeiten.

Der ftDuino ist daher in der Lage, nahezu seine gesamte Aufmerksamkeit einer ganz bestimmten Aufgabe zu widmen und beispielsweise den Zustand eines Eingangs kontinuierlich mit sehr hoher Frequenz abzufragen. Die meisten internen Operationen des im ftDuino verwendeten AVR-Prozessors benötigen einen Systemtakt zur Ausführung. Eine einfache Warteschleife „verbraucht“ weniger als zehn Operationen. Der Prozessor im ftDuino kann daher mit seinen gemächlichen 16 MHz den Zustand eines einzelnen Eingangs problemlos eine Million Male pro Sekunde abfragen.

Zusätzlich verfügt der AVR-Prozessor über interne Zähler, die ebenfalls mit Frequenzen in diesem Bereich betrieben werden können. Schon der für heutige Verhältnisse sehr geringe Takt von 16 MHz erlaubt es dem ftDuino daher, auf externe Hardwareereignisse im Mikrosekundenbereich zu reagieren.

Aber wozu braucht man das? Welche Ereignisse im Mikrosekundenbereich kommen in der realen fischertechnik-Welt überhaupt vor?

Nehmen wir als Beispiel eine einfache Lichtschranke aus Fototransistor und Lampe und ein Objekt von 1 cm Größe, das durch diese Lichtschranke bewegt wird. Bewegt sich

das Objekt mit einem Meter pro Sekunde (also 3,6 km/h), dann dauert die Unterbrechung der Lichtschranke nur 1/100tel Sekunde bzw. 10.000 Mikrosekunden. Bei kleineren Objekten und höheren Geschwindigkeiten reduziert sich dieser Wert entsprechend.

Wichtig ist die hohe zeitliche Auflösung auch bei langsameren Ereignissen, da sie die Messgenauigkeit bestimmt. Beträgt die zeitliche Auflösung der Messung beispielsweise 10 Mikrosekunden und die Länge des zu messenden Ereignisses eine Millisekunde (= 1000 Mikrosekunden), so beträgt die Messgenauigkeit 1%.

Der Arduino und damit auch der ftDuino nutzen einen Prozessor-internen sogenannten Hardware-Timer, um eine Zeitbasis mit einer Auflösung von vier Mikrosekunden zu erzeugen. Die so gewonnene Zeit kann mit Hilfe der Standard-Arduino-Funktion `micros()` jederzeit abgefragt werden kann.

Geschwindigkeitsmessung

Als erstes praktisches Beispiel soll eine doppelte Lichtschranke aus je zwei Fototransistoren und Leuchtdioden zur Geschwindigkeitsmessung realisiert werden. Die Verkabelung ist in Abb. 1 zu sehen, das komplette Modell in Abb. 2.

Die beiden Lichtschranken sind genau 30 mm voneinander entfernt. In diesem Fall

ist es unerheblich, wie weit eine Lichtschranke unterbrochen sein muss, bis sie als unterbrochen erkannt wird, da wir davon ausgehen können, dass sich beide Lichtschranken gleich verhalten.

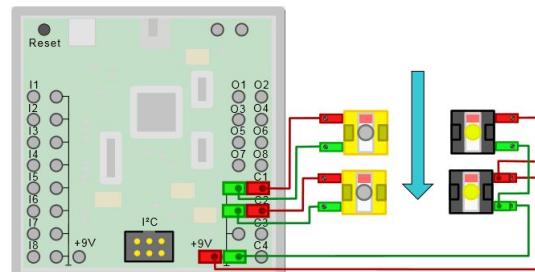


Abb. 1: Verkabelung der Geschwindigkeitsmessung

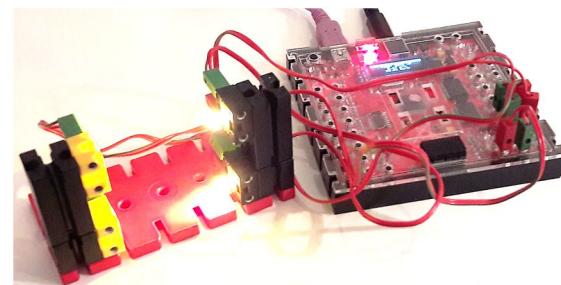


Abb. 2: Modell zur Geschwindigkeitsmessung mit zwei Lichtschranken

Die nötige Software bildet der folgende Sketch. Mit Erscheinen dieser ft:pedia-Ausgabe wird er als Teil der normalen ftDuino-Installation in der Arduino-IDE bei den Beispielen im Verzeichnis

`Beispiele/FtduinoSimple/Speed`
zu finden sein.

```

1 // 
2 // Speed.ino
3 //
4 // Geschwindigkeit mit Hilfe zweier Lichtschranken messen
5 //
6 // (c) 2018 by Till Harbaum <till@harbaum.org>
7 //
8
9 #include <FtduinoSimple.h>
10
11 void setup() {
12   Serial.begin(115200);
13   while(!Serial);
14   Serial.println(">> Speed measurement <<");
15
16   if(!ftduino.counter_get_state(Ftduino::C1))

```

```

17     Serial.println("Warning: Input C1 not low");
18
19     if(!ftduino.counter_get_state(Ftduino::C2))
20         Serial.println("Warning: Input C2 not low");
21 }
22
23 // Abstand der beiden Lichtschranken
24 #define DIST      30                      // 30mm
25
26 void loop() {
27     // beide Lichtschranken einlesen
28     uint8_t c1 = ftduino.counter_get_state(Ftduino::C1);
29     uint8_t c2 = ftduino.counter_get_state(Ftduino::C2);
30
31     // Warten auf die Unterbrechnung der ersten Lichtschranke
32     static uint32_t start;
33     if(!c1 && c2) start = micros();
34     else if(!c2) {
35         // zweite Lichtschranke unterbrochen -> Ausgabe
36         uint32_t t = micros() - start;
37         char str[32], n[10];
38
39         Serial.print("Time: ");
40         Serial.print(t, DEC);
41         Serial.print("µs -> ");
42         dtostrf(1000000.0/((float)t*(1000/DIST)), 4, 2, str);
43         Serial.print(str);
44         Serial.print("m/s -> ");
45         dtostrf(3600000.0/((float)t*(1000/DIST)), 4, 2, str);
46         Serial.print(str);
47         Serial.println("km/h");
48
49     // Warte bis Objekt wieder die Lichtschranken verlassen hat
50     while(!ftduino.counter_get_state(Ftduino::C1) ||
51           !ftduino.counter_get_state(Ftduino::C2));
52
53     delay(1000);
54 }
55 }
```

Listing 1: Sketch Speed.ino

Der in Listing 1 abgedruckte Sketch gibt seine Messergebnisse per USB an den PC aus. Die im Download verfügbare Version kann zusätzlich so konfiguriert werden, dass die Ausgabe auf dem internen OLED-Display eines entsprechenden ausgerüsteten ftDuino erfolgt.

Der Sketch basiert auf der FtduinoSimple-Bibliothek. Diese Bibliothek stellt nur die absolut minimalen Routinen zum Auslesen der Ein- und Ausgänge bereit. In diesem Fall hat das den Vorteil, dass es keine zusätzlichen Dienste im Hintergrund gibt,

die durch das aufwändige Auswerten analoger Eingänge oder ähnlich die eigentliche Messung beeinflussen. Wie oben beschrieben steht damit die volle Aufmerksamkeit des ftDuino für die eigentliche Messung zur Verfügung.

Die Messung selbst gestaltet sich sehr einfach. Wird in Zeile 33 des Sketches erkannt, dass die erste Lichtschranke unterbrochen wurde, so wird der genaue Zeitpunkt mit einer Genauigkeit von vier Mikrosekunden von der `micros()`-Funktion erfasst. Sobald dann die zweite Lichtschranke unterbrochen wird, wird in

Zeile 36 erneut die Zeit erfassst und die vorige Zeit davon abgezogen. Die Differenz ist genau die Zeit, die zwischen dem Unterbrechen der beiden Lichtschranken vergangen ist. In den Zeilen 42 bzw. 45 wird die Zeit dann mit Hilfe des Abstandes `DIST` der beiden Lichtschranken in eine Geschwindigkeit in Meter pro Sekunde (m/s) und Kilometer pro Stunde (km/h) umgerechnet.

Die Funktion `dtostrf()` wandelt das Ergebnis von der Fließkommadarstellung in eine Zeichenkette zur Textausgabe um. Der Parameter „4“ gibt dabei an, dass die Ausgabe maximal vier Zeichen lang sein soll und der Parameter „2“, dass maximal zwei Nachkommastellen angezeigt werden und die Anzeige entsprechend gerundet werden soll.

Nun wird es spannend: Ist das Modell aufgebaut und der Sketch gestartet kann man im seriellen Monitor der Arduino-IDE die Ausgaben des ftDuino sehen. Hier sollte zunächst nur „>> Speed measurement <<“ stehen. Der ftDuino wartet auf ein Ereignis. Ein beherzter Handkantenschlag zwischen die beiden Lichtschranken liefert dieses wie in Abb. 3 zu sehen. Mit etwas Schwung schafft man es, die Lichtschranken in weniger als 2000 Mikrosekunden zu unterbrechen, was einer Geschwindigkeit von über 50 km/h und damit der eines innerorts fahrenden Autos entspricht.

```
>> Speed measurement <<
Time: 3708µs -> 8.17m/s -> 29.42km/h
```

Abb. 3: Anzeige im seriellen Monitor

Besitzer eines OLED-ftDinos können die Ausgabe mit Hilfe des Beispiel-Sketches aus der Arduino-IDE auch auf das lokale Display umleiten wie Abb. 4 zeigt. Hier wurde mit moderaten 22,21 km/h auf die Bauplatte gehauen.



Abb. 4: Anzeige im OLED-Display

Die Geschwindigkeitsmessung ist natürlich nicht auf exakt diese Anordnung beschränkt. Es lassen sich auch Lichtschranken zum Durchfahren mit Modellautos bauen (wie schnell ist eigentlich der Bluetooth Racer?), und auch der Abstand der beiden Lichtschranken lässt sich variieren, wenn man die Konstante im Sketch entsprechend anpasst.

Wer mag, kann auch ganz auf Lichtschranken verzichten und beispielsweise zwei Taster oder Reed-Kontakte nacheinander überfahren.

Erdbeschleunigung

Etwas wissenschaftlicher wird es mit unserem zweiten Versuchsaufbau. Er ist nur unwesentlich komplizierter; sein Schaltplan ist in Abb. 5 dargestellt.

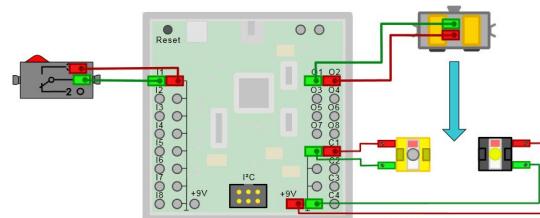


Abb. 5: Schaltplan zur Messung der Erdbeschleunigung

Diesmal kommt nur eine Lichtschranke zum Einsatz, dafür aber zusätzlich ein Elektromagnet und ein Taster. Mit diesem Aufbau lässt sich die durch Gravitation verursachte Erdbeschleunigung messen, indem ein metallischer Gegenstand vom Elektromagneten gehalten wird [2]. Sobald der Nutzer die Taste drückt wird der Elektromagnet stromlos und verliert damit

seine magnetische Wirkung; der Gegenstand löst sich und fällt durch die Lichtschranke.

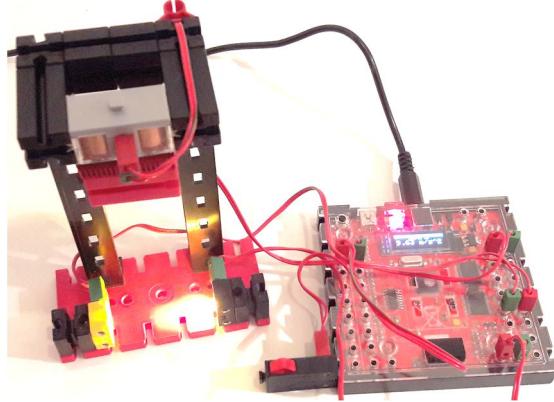


Abb. 6: Modell zur Messung der Erdbeschleunigung

```

1 // 
2 // Gravity.ino
3 //
4 // (c) 2018 by Till Harbaum <till@harbaum.org>
5 //
6
7 #include <FtdduinoSimple.h>
8
9 void setup() {
10   Serial.begin(115200);
11   while(!Serial);
12   Serial.println("Gravity");
13
14   if(!ftduino.counter_get_state(Ftdduino::C1))
15     Serial.println("Warning: Input C1 not low");
16
17   if(ftduino.input_get(Ftdduino::I1))
18     Serial.println("Warning: Input I1 not high");
19
20   ftdduino.motor_set(Ftdduino::M1, Ftdduino::LEFT);
21 }
22
23 // Fallhöhe in Metern
24 #define DIST 0.09 // 9 cm
25
26 void loop() {
27   static uint32_t start = 0;
28   uint8_t c1 = ftdduino.counter_get_state(Ftdduino::C1);
29   uint8_t i1 = ftdduino.input_get(Ftdduino::I1);
30
31   // warte bis Taste gedrückt
32   if(!start && i1) {
33     start = micros();
34     ftdduino.motor_set(Ftdduino::M1, Ftdduino::OFF);
35   }
36
37   // Taste wurde gedrückt, warte auf Unterbrechung der Lichtschranke
38   if(start && !c1) {

```

Aus der Zeit, die vom Abschalten des Magneten bis zur Unterbrechung der Lichtschranke vergeht, lässt sich die Erdbeschleunigung bestimmen.

Abb. 6 zeigt einen entsprechenden Aufbau. Wichtigster Parameter ist hier die Strecke, die der Gegenstand fallen muss, bevor er die Lichtschranke unterbricht. Im abgebildeten Modell muss der rote 30er-Stein mit der im Bild nur schwer zu erkennenden Rückslussplatte (31325) 90 mm fallen bevor die Lichtschranke unterbrochen wird. Dieser Wert ist auch in Zeile 24 des folgenden Sketches hinterlegt.

```

39     uint32_t t = micros() - start;
40     float ts = (float)t/1000000; // us time in seconds
41     char str[32], n[10];
42
43     // g = 2*s/t2
44     dtostrf(2*DIST/(ts*ts), 4, 2, n);
45
46     // Ausgabe der Zeit in Mikrosekunden und der daraus errechneten
47     // Erdbeschleunigung
48     Serial.print(t, DEC);   Serial.print("µs -> ");
49     Serial.print(n);       Serial.println("m/s2");
50
51     delay(1000);
52     ftduino.motor_set(Ftduino::M1, Ftduino::LEFT);
53
54     // Warte bis Baustein wieder entfernt wurde
55     while(!ftduino.counter_get_state(Ftduino::C1));
56     start = 0;
57 }
58 }
```

Listing 2: Sketch Gravity.ino

Der Sketch funktioniert wieder sehr einfach. Wird in Zeile 32 erkannt, dass die Taste gedrückt ist, wird der Magnet spannungslos gemacht und die aktuelle Zeit in Mikrosekunden gespeichert. Sobald dann in Zeile 38 erkannt wird, dass die Lichtschranke unterbrochen ist, wird – wie schon bei der Geschwindigkeitsmessung – die Differenz zur Startzeit gebildet. Mit Hilfe der Gleichung

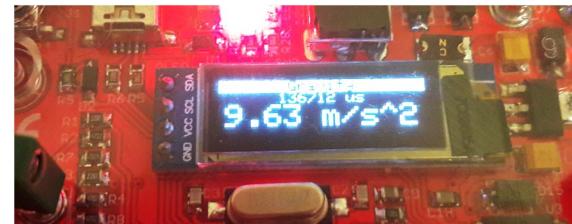
$$g = 2 \cdot \frac{s}{t^2}$$

lässt sich in Zeile 44 des Sketches aus der Strecke in Metern und der Fallzeit in Sekunden die Erdbeschleunigung in m/s² errechnen. Danach wartet der Sketch eine Sekunde, bevor er den Magneten wieder einschaltet. Die nächste Messung kann starten, sobald das Objekt aus der Lichtschranke genommen wurde.

Auch hier gibt es in der Arduino-IDE im Verzeichnis

Beispiele/FtduinoSimple/Gravity

eine erweiterte Version, die einen OLED-ftDuino nutzen kann. Ein mögliches Ergebnis ist in Abb. 7 zu sehen. Die Ergebnisse dürften zunächst deutlich vom korrekten Wert (9.81 m/s²) abweichen.

*Abb. 7: Anzeige des Ergebnisses im OLED-Display*

Tatsächlich ist es sogar wahrscheinlich, dass erstmal gar nichts funktioniert, weil die Rückschlussplatte einfach am spannungslosen Magneten kleben bleibt. Das ist ein bekanntes Phänomen, das ein Streifen Tesa-film auf der Metallfläche der Rückschlussplatte löst (wie von fischertechnik bereits in Hobby-3 Band 2 [3] auf Seite 16 beschrieben).

Selbst wenn die Last nun abfällt, wird die gemessene Zeit immer noch unerwartet lang und die errechnete Beschleunigung unerwartet gering sein. Das liegt daran, dass der Tesa-filmstreifen das Problem nicht vollständig löst. Es dauert immer noch einige Zeit, bis sich das Magnetfeld soweit abgebaut hat, dass das Objekt schließlich fällt.

Weitere Tesa-film-Schichten lösen auch dieses Problem. Erst wenn der Magnet das Objekt noch so gerade eben halten kann,

erhält man Messwerte, die im erwarteten Bereich liegen.

Theoretisch könnte man auch die Tatsache nutzen, dass man die Kraft des Magneten durch die Analogausgänge des ftDuino reduzieren kann. Das erfordert aber den Einsatz der „großen“ ftDuino-Bibliothek, die unter anderem die Erzeugung der analogen Ausgangswerte mit Hilfe der sogenannten PWM im Hintergrund erledigt. Aber genau diese Hintergrundprozesse stören, wie eingangs erläutert, unsere Messung, da sie den Prozessor des ftDuino regelmäßig „ablenken“. Daher ist die Tesa-film-Schicht die bevorzugte Lösung.

Fazit

Gerade im Bereich der so genannten Echtzeitverarbeitung sind kleine und einfache Mikrocontrollerlösungen wie die Arduinos und der ftDuino jedem komplexeren System überlegen. Das schlanke Design erlaubt es, das Verhalten des Systems exakt vorherzusagen und präzise Messungen auszuführen. Das ermöglicht einfache, aber interessante Experimente. Mehr zum ftDuino erfährt man unter [1].

Referenzen

- [1] [ftDuino](http://ftduino.de), <http://ftduino.de>
- [2] Wikipedia: [Schwerefeld](#).
- [3] Fischer Werke: [Hobby-3 Band 2](#).

Computing

Das Universal-Interface am LPT-Port unter Windows 10/64 bit

Helmut Jawtusch

Dieser Artikel richtet sich an Besitzer des legendären Universal-Interfaces (UI), deren PC noch einen LPT-Port besitzt. Möglicherweise wurde inzwischen ein 64-bit-System installiert. Ich beschreibe hier, wie man dennoch das Universal-Interface mit den Sprachen Terrapin Logo oder Java verwenden kann.

In meinem Artikel *Das Universal-Interface am LPT-Port* [1] hatte ich vorgeschlagen, auf einer Partition ein 32-bit-System (z. B. Windows XP) zu installieren. Dazu müsste man zuerst eine Installations-CD mit Windows XP erstellen, auf der ein SATA AHCI-Treiber integriert wurde. Nach der Installation von Windows XP muss man nach geeigneten Treibern suchen. Da das recht aufwändig ist, habe ich nach einer

einfacheren Lösung gesucht. Ulrich Müller hatte im Jahre 2000 beschrieben, wie man seine LPT-Treibersoftware unter Windows 98 verwenden kann:

1. *WRTdev0.VXD* nach *Windows\System\VMM32* kopieren
2. *WinRTVXD.REG*: Eintrag in die Registry durch Doppelklick

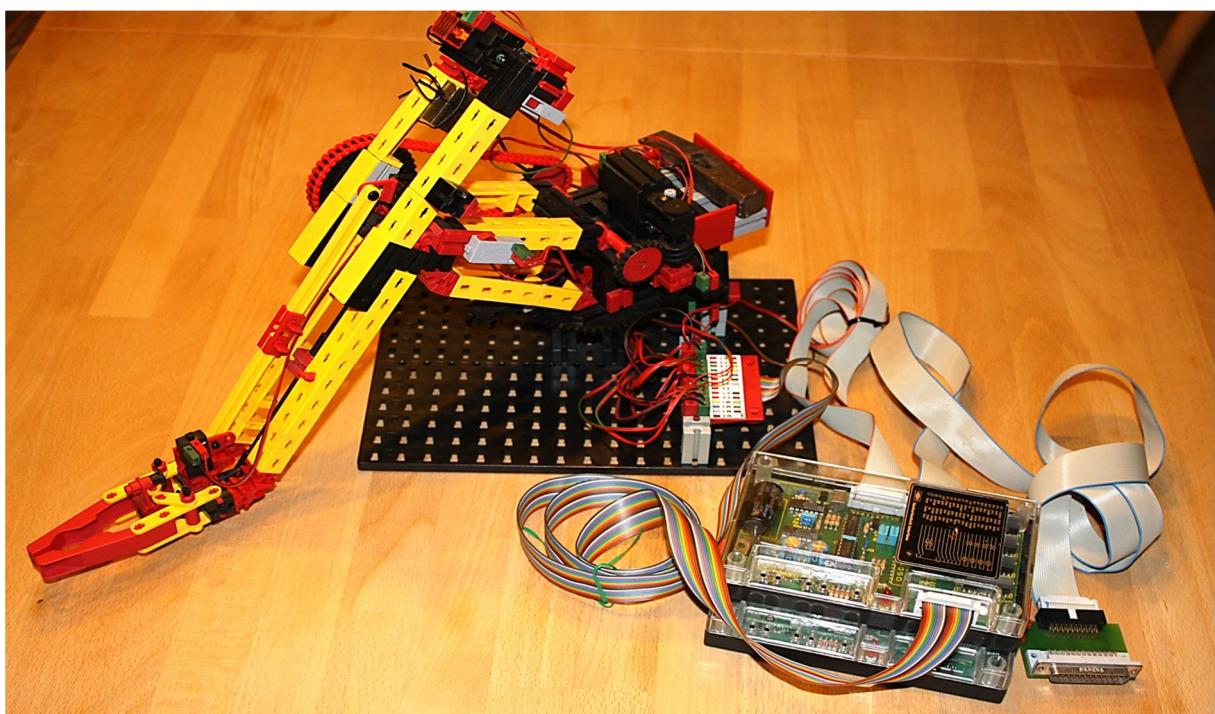


Abb. 1: Trainings-Roboter am Universal-Interface

Um diesen VXD-Treiber unter Windows 10 64 bit nutzen zu können, habe ich ein virtuelles Windows 98-System eingerichtet. Dazu habe ich den VMware Player [2] (freeware) installiert. Windows 98 befindet sich in der Datei *Microsoft Windows 98 Second Edition [VMware VM].7z* [3]. Man braucht damit keine Windows 98-CD!

Als Treiber für das UI habe ich wieder *umfish.dll* (von Ulrich Müller [4]) und meinen darauf aufbauenden Treiber *umfishcount.dll* verwendet.

Unter Terrapin Logo hat die Ansteuerung des UIs damit sofort geklappt. Für Java musste ich noch die Wrapper-Datei *javafish.dll* und die Jar-Datei *ftcomputing.jar* überarbeiten.

Ich habe für die Verwendung von Java (in der Windows 98-VM) *jdk-6-windows-i586* und *Eclipse-SDK-3.3-win32* installiert. Damit funktioniert auch die Ansteuerung des UI mit Java.

Abb. 2 zeigt das Fenster, in dem (unter Windows 10 64 bit) Windows 98 emuliert wird.

Damit der vorhandene LPT-Port von Windows 98 genutzt werden kann, musste ich im VMware Player noch *Edit virtual machine settings* aufrufen und dort explizit die Ressource *LPT-Port* hinzufügen: *Add → Parallelport → Use physical parallel port on the host*.

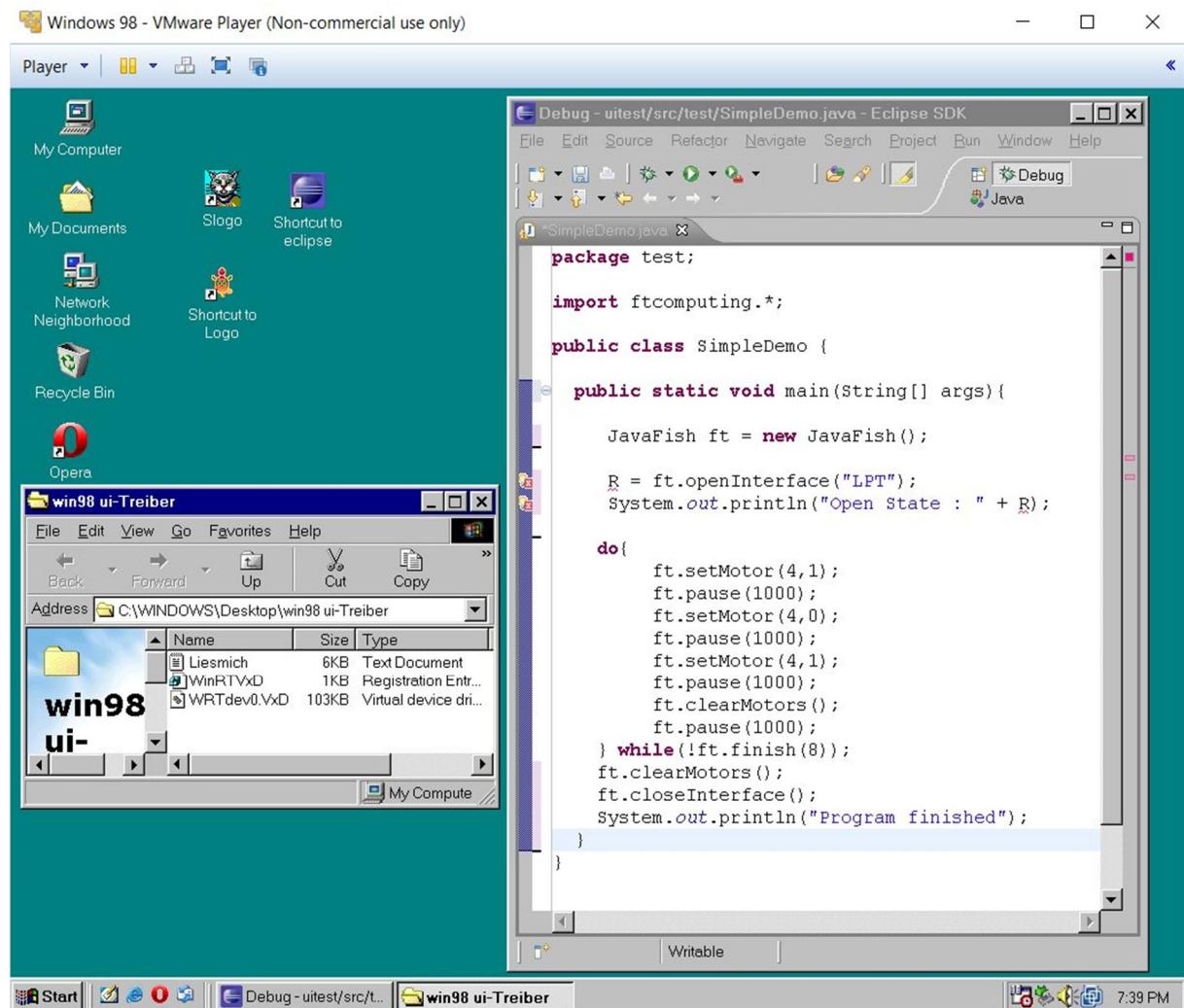


Abb. 2: Windows 98 als virtuelle Maschine unter Windows 10

In *C:\Windows\System* habe ich die Treiber *umfish.dll*, *umfishcount.dll*, *javafish.dll*, *ftcomputing.jar* abgelegt und die Datei *ftComputing.jar* wie folgt eingebunden:

1. Eclipse: Projekt anlegen (Eclipse in Java Perspektive mit Standard Einstellungen; Arbeiten im Package Explorer: Kontext-Menu über Rechtsklick).
2. *New, Java Project*, Project Name: *uitest*, *Finish*.
3. *New, Package*, Name: *test*.
4. Package *NeuesProjekt* aufklappen, *New, Class*. Name: *SimpleDemo*, Checkbox *public static void main* aktivieren.
5. NeuesProjekt: *Build Path, Add External Archives*. Verzeichnis *ftComputing* im Workspace suchen und *ftcomputing.jar* wählen.
6. In *class SimpleDemo* (im Editor) hinter package *NeuesProjekt* *import ftcomputing.*;* einfügen.
7. Weiter im Editor mit Programmdetails für reine Konsolen- bzw. einfache Swing-Programme.

Bei Interesse kann ich die Windows 98-Treiberdateien *umfish.dll*, *umfishcount.dll*,

javafish.dll, *ftcomputing.jar* sowie die Java-Installationsdateien *jdk-6-windows-i586* und *Eclipse-SDK-3.3-win32* in einem Software-Paket zum UI auf der [ftcommunity.de](#) unter Downloads/Software hochladen, sobald das wieder möglich ist.

Anmerkung: Sollte das Motherboard kein LPT-Port besitzen, kann die Karte Longshine LCS-6320 PCI Express mit Parallel-Port (Preis ca. 10 €) eingesetzt werden. Beachtet dazu auch meinen Beitrag in [ft:pedia 4/2017 \[1\]](#). Ich habe das mit dem virtuellen Windows 98 nicht erprobt, vermute aber, dass es wie unter winxp funktioniert. Über eine Rückmeldung, z. B. im [Forum der fischertechnik-Community](#), würde ich mich freuen.

Quellen

- [1] Helmut Jawtusch: [Das Universal-Interface am LPT-Port](#). [ft:pedia](#) 4/2017, S. 34-35.
- [2] VMware: [Workstation Player](#).
- [3] VMware-Image einer [Windows 98-VM](#).
- [4] Ulrich Müller: [ftComputing](#). [ft-Community-Website](#).

Computing

Der ftPwrDrive-Controller für Schrittmotoren und Servos – Teil 1

Dirk Wölffel, Björn Gundermann, Christian Lauff,
Christian Bergschneider, Stefan Fuss

Das fischertechnik-Universum basiert auf normalen Elektromotoren; Schrittmotoren gibt es nur beim 3D-Drucker und Servos lassen sich ausschließlich „analog“ mit dem ControlSet benutzen. Höchste Zeit, beide Typen in das fischertechnik-System zu integrieren.

Aus ein paar chinesischen Platinen und der Frage „Bekommt Ihr das unter ROBO Pro ans Laufen?“ entstand auf der diesjährigen Südconvention die Idee zu einem fischertechnik-Schrittmotorcontroller. Nach wenigen Stunden brummte bereits der erste Schrittmotor am Protoboard. Es zeigte sich jedoch sehr schnell, dass die Tücke wie immer im Detail liegt und die Integration ins fischertechnik-System aus weit mehr als ein paar fernöstlichen Platinen besteht.

Bei der Vorstellung des Projektes in der ftCommunity wurden zusätzlich PWM-bzw. Servo-Ausgänge gewünscht. Der Vorteil ist, dass bei einer Servo-Ansteuerung keine Endschalter benötigt werden. Über die PWM-Ausgänge lassen sich jedoch nicht nur Servos ansteuern: Mit einer externen Leistungsendstufe² kann auch die Drehzahl einer Fräse eingestellt oder die Temperatur des Extruders bei einem 3D-Drucker geregelt werden.

Der ftPwrDrive-Controller

Der ftPwrDrive-Controller ist ein Community-Projekt, mit dem wir jedem fischertechniker eine Plug & Play-Lösung für

Schrittmotoren und Servos an den gängigen Controllern – TX, TXT und ftDuino – zur Verfügung stellen wollen. Der ROBOTICS BT Smart Controller kann leider nicht unterstützt werden, da er keine Möglichkeit zum Anschluss externer Hardware bietet. Den Controller wird es als fertiges Gerät geben, so dass man seine Motoren direkt anschließen und loslegen kann.

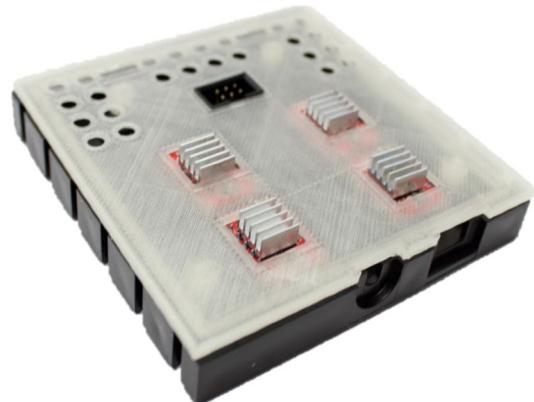


Abb. 1: Der Prototyp des ftPwrDrive

Fertige ftPwrDrive-Controller und passendes Zubehör können jetzt schon über [Björns WebShop](#) vorbestellt werden. Das Gehäuse (Abb. 1) wird wie der ftDuino [5] und die fischertechnik-Controller die Grundfläche

² Das PWM-Signal ist ein Logikpegel, mit dem die Servomotoren direkt geregelt werden können. Um damit einen leistungsfähigen Motor oder

eine Heizung zu steuern muss eine Endstufe die notwendige Leistung bereitstellen.

90x90 mm haben; die Höhe wird etwas über der des ftDuino liegen.

Am ftPwrDrive-Controller können bis zu vier Schrittmotoren angeschlossen werden. Hier bietet sich der Typ NEMA³ 14 an (Abb. 2), der u. a. auch beim fischertechnik-3D-Drucker zum Einsatz kommt. Durch eine Stromjustierung kann der Controller bei Bedarf auch auf andere Schrittmotoren eingestellt werden. Der Anschluss der Motoren erfolgt über die gewohnten ft-Bundhülsen.



Abb. 2: NEMA-14 Schrittmotor

Der Controller besitzt für die vier Schrittmotor-Ausgänge jeweils einen Tastereingang. Diese werden von der Firmware sowohl als Endschalter als auch als Referenzpunkt genutzt.

Wird ein Endschalter ausgelöst, so stoppt die Firmware den Motor. Durch die höheren Leistungsdaten könnten sonst die fischertechnik-Bausteine im Modell Schaden nehmen. Gleichzeitig kann die Firmware den Endschalter als Referenz- oder Nullpunkt nutzen. Die Firmware fährt den Motor so lange, so dass er gerade nicht den Endschalter auslöst. Man erhält so eine gut

reproduzierbare Referenzposition im Modell, auf der alle anderen Fahrkommandos aufsetzen können.

An die vier Servo-Ausgänge können fischertechnik-Servos aus dem Control-Set oder kompatible Servos direkt angeschlossen werden.

Schrittmotoren sind sehr hungrig und kommen mit der fischertechnik-9V-Versorgung nicht in Gang. Sie benötigen 18-24V Betriebsspannung mit für fischertechnik hohen Leistungsdaten. Da ein NEMA-14-Motor pro Spule bis zu 0,8A Strom zieht, werden Netzteile mit hoher Leistung benötigt. Als Stromversorgung eignen sich Notebooknetzteile mit 18-24V ab 100W Leistung.

Der Anschluss an den Controller erfolgt über den I²C-Bus. TX und ftDuino können direkt über den 6-poligen EXT-Stecker verbunden werden; der TXT benötigt zusätzlich einen Levelshifter wie zum Beispiel den ft-Extender (Abb.3) [1].



Abb. 3: der ft-Extender

Die Programmierung des ftPwrDrive kann in verschiedenen Umgebungen erfolgen. Eine ROBO Pro-Bibliothek gibt es bereits. Für die Community Firmware und python wird es eine Integration in ftrobopy geben; der ftDuino erhält eine Arduino-Bibliothek.

³ NEMA ist eine standardisierte Baureihe von Schrittmotoren, die von verschiedenen Herstellern angeboten werden. Der Typ NEMA-14

passt aufgrund seiner Leistungsdaten und Größe gut zu fischertechnik.

Schrittmotoren

Beim Schrittmotor liegen die Spulen im Gehäuse. Permanentmagnete sind auf der Welle des Motors angebracht. Fließt nacheinander durch die Spulen Strom, so dreht sich der Motor je Signal von Spule zu Spule. Der Motor führt je Steuersignal einen Schritt aus. Die Schrittweite eines NEMA-14-Motors beträgt $1,8^\circ$ und erlaubt somit eine für fischertechnik sehr präzise Positionssteuerung. Eine gute Beschreibung der Funktionsweise, Kennlinien und unterschiedlichen Typen ist in [4] zu finden.

Durch ihre Bauweise sind Schrittmotoren schwerer als fischertechnik-Elektromotoren und echte Stromfresser. Für batteriebetriebene Erkundungsroboter sind sie definitiv die falsche Wahl. Für stationäre Modelle haben Schrittmotoren aber viele Vorteile:

- Der Schrittmotor benötigt keinen Dekoder, er hat ihn im Antrieb integriert. Er fährt die angegebene Schrittzahl und bleibt dann ohne Nachlauf stehen. Ideal für Plotter, 3D-Drucker, Fräsen und Roboterarme.
- Schrittmotoren können sich extrem langsam bewegen. Getriebe werden in der Regel nicht benötigt. Ohne Getriebe entfällt das Umkehrspiel bei Plottern und Druckern. Uhren können einfach gebaut werden.
- Die NEMA-14-Baureihe ist stärker als XM- oder Encodermotoren. Im Stillstand und bei geringen Drehzahlen bringen sie sogar ihr größtes Drehmoment. Der Roboterarm bleibt somit im Stillstand stehen und sinkt nicht ab.
- Synchrones Verfahren von mehreren Achsen ist eine für Schrittmotor-Steuerungen relativ einfache Übung. Mit einem x/y-Antrieb und Encodermotoren echte Kreise in ROBO Pro zu zeichnen ist dagegen fast unmöglich.

Servos

Servomotoren (Abb. 4) bestehen aus kleinen Elektromotoren mit einer integrierten Positionsregelung. Der Servo erhält über ein Steuersignal eine Winkelvorgabe, die Positionsregelung fährt diesen Winkel an und hält anschließend die Winkelposition ein. Im Modellbau wird als Steuersignal in der Regel ein PWM-Signal eingesetzt [3].



Abb. 4: Servomotor

Sie können sich so nur um weniger als 100 Grad bewegen, eignen sich aber für alle Stellmotoranwendungen, da sie wie der Schrittmotor die Endposition halten. Servos sind verhältnismäßig klein und leicht. Als Stromversorgung benötigen sie eine Spannung zwischen 5V und 6,8V.

Die klassische fischertechnik-Anwendung ist der Einsatz als Lenkung in Verbindung mit dem Control Set. Es lassen sich damit aber auch elektrisch angetriebene Kugelbahnweichen bauen oder Finger einer mechanischen Hand steuern.

In der ft:pedia wurden bereits Servo-Ansteuerungen auf Basis von Adafruit-Platinen vorgestellt [2, 3].

Das Zubehör

Ein Controller allein ist kein fertiges Modell, deshalb wird es zahlreiches, auf die NEMA-Serie und den Controller abgestimmtes Zubehör (Abb. 5) geben. Einiges davon könnt ihr bequem über Björns Shop bestellen. NEMA-14-Motoren wird es mit passenden Schrauben und fischertechnik-Halter geben.



- 1) Original Motorhalter für den Nema 14 Schrittmotor
 2) Schrittmotor (160528) Original ft-Antriebritz Nema 14 (159577)
 3) Original ft-Antriebshülse Nema 14 (160549)
 4) Schnecke für Powermotor (TST)

- 5) Zahnrad Z 10 (TST)
 6) Flexible Wellenkupplung Aluminium von 4 auf 5 mm
 7) Flexible Wellenkupplung Gummi von 4 auf 5 mm
 8) Starre Wellenkupplung Stahl von 4 auf 5 mm
 9) Original Mini-Taster (37783)

Abb. 5: Zubehör

Bei den Servos wird es kleine und große Servos geben. Idealerweise finden wir einen baugleichen zur fischertechnik-Fernsteuerung. Ein erster Prototyp für ein Servo-Gehäuse ist in Abb. 6 zu sehen. Gebrauchte fischertechnik-Stecker und -Taster für die Endlage werden ebenfalls angeboten werden. Wer kein altes Notebooknetzteil mehr hat, kann es sich im Shop mitbestellen.



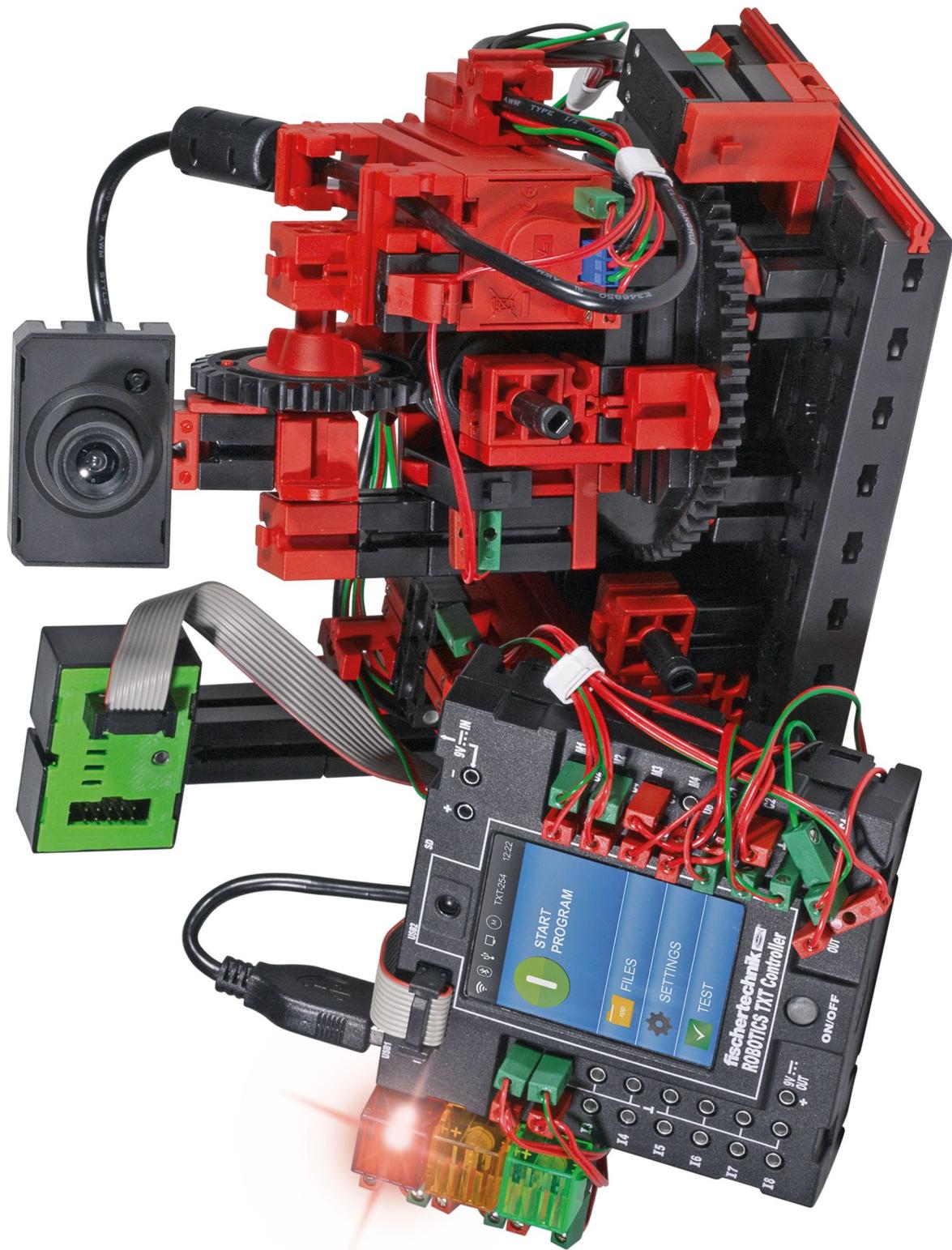
Abb. 6: Servo mit Gehäuse

Fehlt euch noch weiteres Zubehör? Dann meldet euch bei uns im Forum oder über Björns Shop und dessen Kontaktdaten.

Im zweiten Teil des Beitrags werden wir Euch den fertigen Controller und seine Programmierung in den unterschiedlichen Sprachen vorstellen.

Quellen

- [1] Stefan Fuss, Christian Bergschneider, Björn Gundermann, Christian Lauff, Rudenz Schulz: [Plug & Play am I²C-Bus mit dem ftExtender](#). ft:pedia 2/2018, S. 55-59.
- [2] Dirk Fox: [I²C mit dem TX\(T\) – Teil 16: Servo-Driver](#). ft:pedia 2/2017, S. 41-47.
- [3] Christian Bergschneider, Stefan Fuss: [Alternative Controller \(3\): Der ftPi – ein Motor Shield für den TX\(T\)](#). ft:pedia 2/2016, S. 68-72.
- [4] [Schrittmotoren](#). rn-wissen.de
- [5] Till Harbaum: [ftDuino – Open-Source trifft Konstruktionsbaukasten](#). ft:pedia 1/2018, S. 85-91.



Modell aus dem fischertechnik-Baukasten ROBOTICS TXT Smart Home (2018)