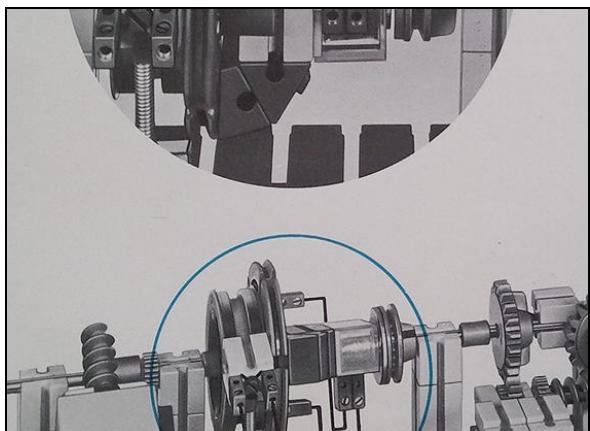
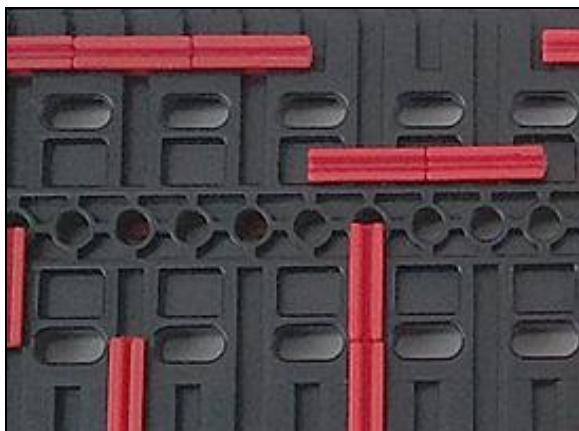
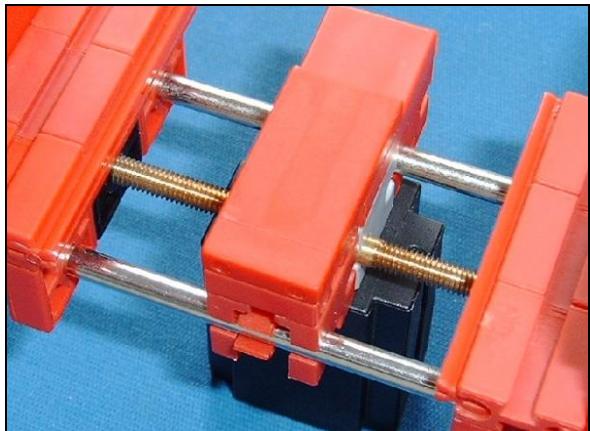
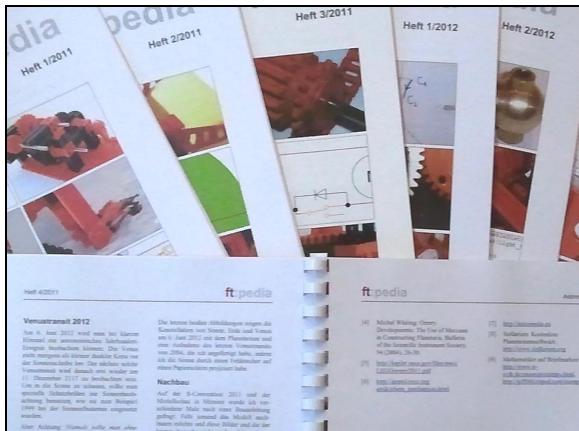
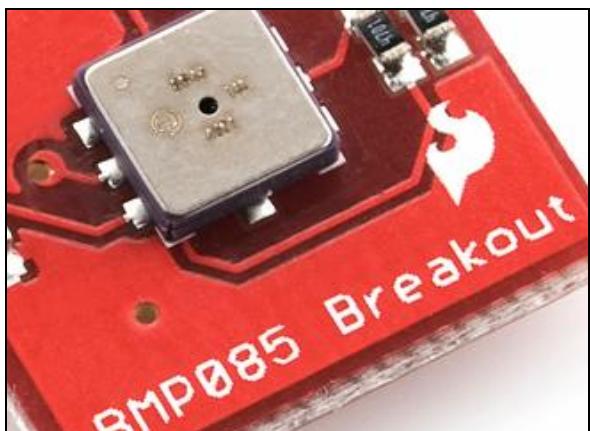


ft:pedia

Heft 1/2013



$$\sum_{i=1}^n x_i y_i - \left(\sum_{i=1}^n x_i \right) \left(\sum_{i=1}^n y_i \right)$$
$$- \left(\sum_{i=1}^n x_i \right)^2 \cdot \left(n \sum_{i=1}^n y_i^2 \right)$$



Herausgegeben von
Dirk Fox und Stefan Falk

ISSN 2192-5879

Editorial

Konkurrenz belebt das Geschäft. Hoffentlich.

Dirk Fox, Stefan Falk

Wer einmal mit beiden Bausätzen Roboter konstruiert hat, weiß: Der 2008 erschienene Robo TX Controller ist dem Lego-Mindstorms-Pendant NXT von 2006 nicht nur ebenbürtig: flacher, besser verbaubar, kein proprietäres Steckersystem, acht statt vier Sensoreingänge und vier statt drei Motorausgänge. Das nötige Kleingeld vorausgesetzt lassen sich zudem bis zu neun TXe koppeln – und so 36 Motoren steuern und 72 Sensoren auswerten. Auch technisch hat der TX dem Lego-Controller manches voraus: Mit acht MByte RAM verfügt er über den 128-fachen Speicherplatz (NXT: 64 kByte), der Flash-Speicher ist mit zwei MByte achtmal so groß (NXT: 256 kByte) und der ARM9 des TX ist mit 200 MHz viermal schneller getaktet als der ARM7 des NXT (48 MHz).

Beste Startbedingungen für einen Markterfolg. Dennoch hat der TX geschwächelt. Denn zu einem erfolgreichen Robotik-Baukasten gehören nicht nur Hardware und Mechanik. So unterstützt die Firmware erst seit April 2012 das I²C-Protokoll – mit der Folge, dass bis dahin nur ausgewählte fischertechnik-Sensoren (Spursensor, Fotowiderstand/-transistor, NTC, Farbsensor, Reed-Kontakt und seit 2010 ein Ultraschall-Abstandssensor) angeschlossen werden konnten. Und der CAMERA-Anschluss ist bis heute nicht nutzbar.

Damit waren fischertechnik-Robotern zahlreiche Möglichkeiten wie die Nutzung von Kompass-, Gyro- oder Beschleunigungssensoren verwehrt. Der NXT punktete derweil mit Sensoren der Firma HiTechnic, die nach und nach ins Lego-Programm

aufgenommen wurden. Vor allem aber bei der Integration in verbreitete Entwicklungsumgebungen wie .NET, [Microsoft Robotics](#), [LabView](#) oder [ROS](#) und der Unterstützung [verschiedenster Programmiersprachen](#) hat der NXT die Nase vorn. Denn ungeachtet der uneingeschränkt empfehlenswerten Robo Pro-Umgebung ziehen nicht nur Lehrer meist Standard-Programmiersprachen einer proprietären Umgebung vor. Die Folge: Trotz der Überlegenheit von fischertechnik bei der Hardware werden Roboter-Wettbewerbe von Lego-Modellen dominiert.

Am 04.01.2013 hat Lego [den Nachfolger des NXT angekündigt](#): Der EV3 genannte „Brick“ wird mit einem ARM9 und 64 MByte RAM (plus 16 MByte Flash-Speicher) ausgestattet, kann via WLAN angesteuert werden und erhält einen SD-Reader. Die Zahl der Motorausgänge steigt auf vier, die der Sensoreingänge ebenfalls.

Dank I²C und der (von Fans entwickelten) Microsoft Robotics Treiber könnte der TX dennoch Boden gut machen – gäbe es mehr (I²C-)Sensoren, eine Treiber-Unterstützung von Java, ROS, LabView und .NET, oder wäre die Plattform wenigstens offen, so dass die Community bestehende Entwicklungsumgebungen daran anpassen könnte.

Jetzt ist fischertechnik am Zug.

Beste Grüße,
Euer ft:pedia-Team

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

Inhalt

Konkurrenz belebt das Geschäft. Hoffentlich.....	2
Programmsteuerungen	4
ft-Spezialteile made by TST (Teil 3).....	20
Perlentauchen (Teil 3)	22
I ² C mit dem TX – Teil 3: Luftdruckmessung.....	32
„Einmessen“ eines digitalen Messgeräts	39
ft:pedia gedruckt	49

Termine

Was?	Wann?	Wo?
Fan Club Tag	14.07.2013	Waldachtal/ Tumlingen
c't hardware hacks Maker Faire	03.08.2013	Hannover
fischertechnik Convention 2013	28.09.2013	Erbes-Büdesheim

Hinweise

Seit dem 24.01.2013 gibt es die (kostenlose) **Android-App ROBOdroid 1.0**, mit der der Gabelstabler des TX Training Lab und das Grundmodell des TX Explorers via Bluetooth gesteuert werden können.

Am 03.08.2013 findet die erste [Maker Faire](#) der [c't hardware hacks](#) in Hannover statt – eine Messe für Bastler und Hacker (15 € Standgebühr, [Anmelde](#)schluss: 15.06.2013).

Seit dem 25.03.2013 steht **Robo Pro 3.2.3** auf der Webseite von fischertechnik zum [Download](#) bereit.

Impressum

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

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

Autoren: Stefan Falk (steffalk), Dirk Fox (Dirk Fox), Georg Stiegler (fantogerch), Andreas Tacke (TST).

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.

Elektromechanik

Programmsteuerungen

Stefan Falk

Als Fortsetzung der Motorsteuerungen-Artikelserie schauen wir uns heute an, wie man nur mit elektromechanischen Mitteln programmgesteuerte Maschinen bauen kann. Den Ablauf der einzelnen Vorgänge in der Maschine kann man dann durch Austausch des „Programms“ beliebig neuen Anforderungen anpassen.

Unsere in der ft:pedia bisher gebauten elektromechanischen Maschinen lösten einen auszuführenden Televorgang dadurch aus, dass ein vorheriger fertiggestellt wurde. Das wurde z. B. dadurch realisiert, dass am Ende eines Vorgangs ein Taster von der Maschine selbst betätigt wurde, der dann einen anderen Vorgang – etwa durch Einschalten eines Motors – auslöste. Für manche Aufgaben führt das aber zu sehr komplizierten Schaltungen und einem hohen Aufwand an Bauteilen.

Für komplexe Abläufe gibt es *programmgesteuerte* Maschinen, die eine gestellte Aufgabe mit relativ einfacher Schaltung und wenig Materialaufwand lösen. Zwei solcher Modelle wollen wir heute vorstellen.

Der Klassiker: Verkehrsampeln

Schon 1968 kam der erste Elektromechanik-Kasten^{[1], [2]} von fischertechnik heraus, der *e-m* bzw. *e-m1* (etwas später erschien nämlich mit dem *e-m2* sein kleiner Bruder). Er enthielt unter anderem zwei Schleifringe nebst darauf anzubringenden Unterbrecherstücken, Feder-gelenksteine und Kontaktstücke (siehe den Artikel „Perlentauchen“ in dieser ft:pedia-Ausgabe). Damit konnten recht elegant programmgesteuerte Maschinen gebaut werden. Ein echter Klassiker ist die Verkehrsampel:



Abb. 1: Elektromechanisch gesteuerte Ampel auf der Titelseite des e-m1-Handbuchs

Durch die auf den umlaufenden Metallbahnen der Schleifringe anzubringenden Unterbrecherstücke konnte so die wirklichkeitsgetreue Ansteuerung der roten, gelben und grünen Lampe realisiert werden:

Phase	Rot	Gelb	Grün
Rot	X	-	-
Rot-Grün-Übergang	X	X	-
Grün	-	-	X
Grün-Rot-Übergang	-	X	-

Natürlich lässt sich eine Ampel auf unzählige Arten realisieren – mechanisch,

elektromechanisch, elektronisch und computergesteuert. Die unterschiedlichen Lösungen gleichen sich aber alle in einem: Sie brauchen nichts weiter zu tun, als die Lampen zum richtigen Zeitpunkt ein und aus zu schalten. Das wiederholt sich dann einfach ständig, und schon arbeitet die Ampel endlos nach dem vorgegebenen „Programm“.

Um das praktisch nachzuvollziehen, bauen wir uns selber eine kleine Verkehrsampel:

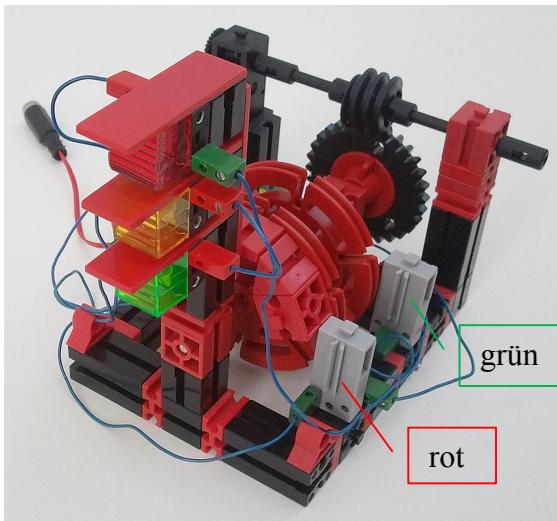


Abb. 2: Ampel aus aktuellen Teilen gebaut

Ein Motor dreht eine selbstgebaute Schaltwalze, die drei Taster betätigt, die wiederum je eine der Lampen der Verkehrsampel steuern.

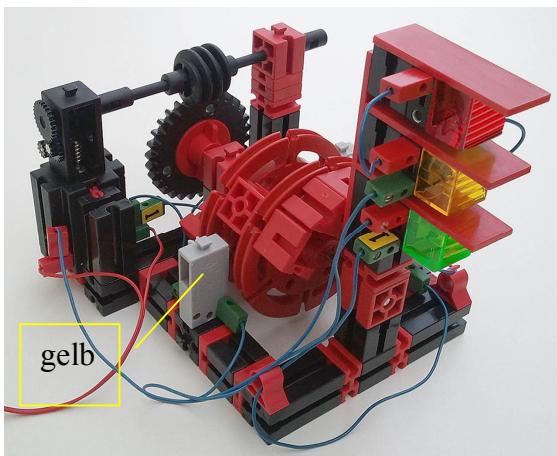


Abb. 3: Ansicht von links vorne

Anstelle der Schleifkontakte des 1968er Modells verwenden wir also drei Minitaster (für rot und grün auf der einen, für gelb auf der anderen Seite), und anstelle von Schleifringen geeignet mit Bausteinen bestückte Drehscheiben.

Das ganze kommt ohne eine Grundplatte aus, denn alles ist auf folgendem Gestell aufgebaut:

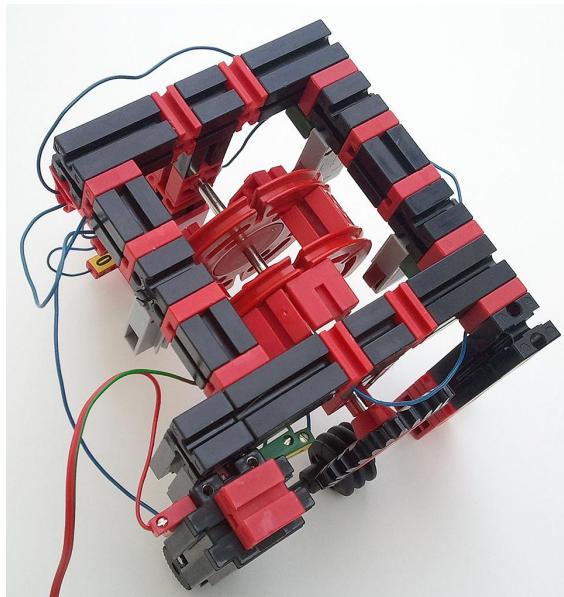


Abb. 4: Das Bodengestell der Ampel

Die vielen Bausteine 5 und 7,5 erleichtern den rasterkonformen Aufbau. Die ft-Drehscheibe ist nämlich genau 5 mm dick. Auch der Motor ist genau justierbar: Unter ihm sitzen zwei Bausteine 5, die genau den richtigen Abstand vom Boden bilden.

Auf der folgenden Abbildung sieht man auch gut, wie die „Sonnenblenden“ der Lampen gebaut sind: Nur die unterste, grüne Lampe sitzt direkt auf dem Baustein 30. Gelb und grün sitzen stattdessen auf je einem Baustein 15, und jede Lampe hat eine Bauplatte 15 • 45 als Blende. Die Bausteine 15 werden nur durch eine Bauplatte 15 • 60 auf der Rückseite zusammengehalten:

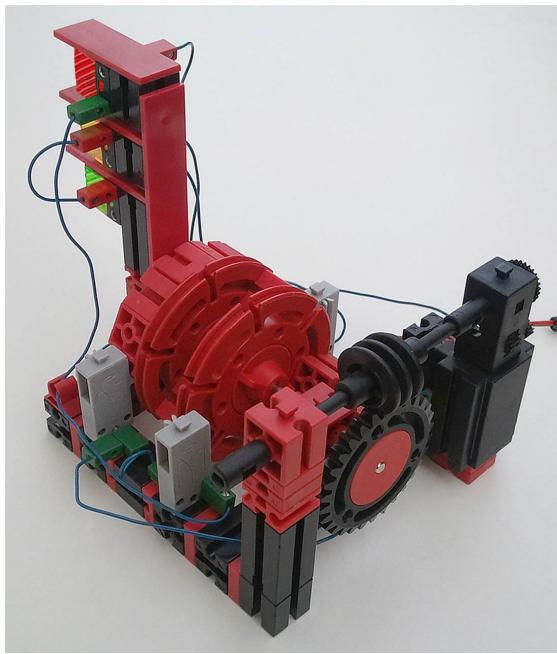


Abb. 5: Rückseite

Die elektrische Schaltung für die Ampel ist recht einfach: Der Motor läuft einfach immer durch, und jeder Taster hängt im Stromkreis einer Lampe – wird er betätigt, leuchtet „seine“ Lampe auf.

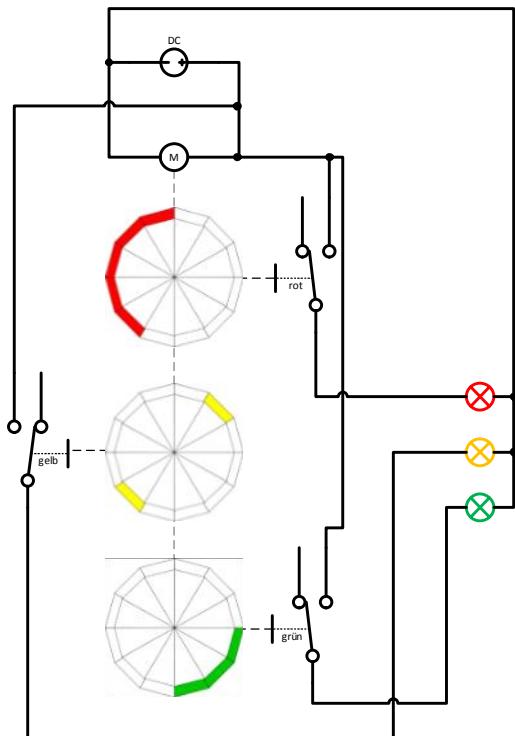


Abb. 6: Die Schaltung der Ampel

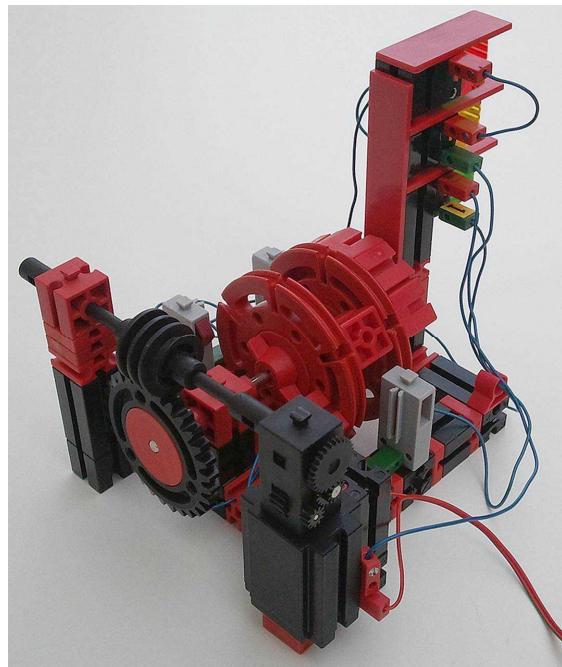


Abb. 7: Motorseite

Jetzt brauchen wir nur noch das geeignete „Programm“, nachdem die Lampen angesteuert werden. Dazu bestücken wir zwei Drehscheiben wie folgt:

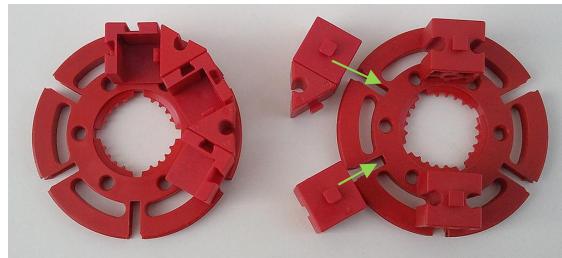


Abb. 8: Bestückung der Drehscheiben

Die linke Scheibe in Abb. 8 trägt die Bestückung für die rote Lampe, die rechte die für die gelbe (die ja nur zwei Mal im gesamten Zyklus kurz aufleuchten muss). Die Bauteile, die neben der rechten Drehscheibe liegen, gehören auf deren Unterseite – das sind die für die grüne Lampe.

Mit unseren „Ersatz-Schleifringen“ wie abgebildet bestückt gibt es Phasen, in denen keine der Lampen leuchtet. Bestückt man aber ohne „Pausen“, leuchten zeitweise Lampen gemeinsam, die nicht gemeinsam leuchten sollten. Das lässt sich in dieser Bauart mit den Minitastern kaum

vermeiden – probiert am besten selbst, was euch besser gefällt.

Der Motor muss dann noch richtig herum angeschlossen werden, damit die Ampel auch den richtigen Ablauf hat. Die rote und gelbe Lampe müssen im Ablauf zusammen vor der grünen aufleuchten, nicht danach. Wer noch keinen Führerschein hat, schaue sich das mal an einer echten Ampel an.

Damit ist unsere erste programmgesteuerte Maschine fertig! Viel Spaß beim Ausprobieren, Tüfteln, Verbessern, Ergänzen und Weiterentwickeln...

Aber natürlich können wir noch viel spannendere Sachen programmgesteuert bauen als Ampeln. Deshalb stellen wir als nächstes Modell eine wirklich „frei programmierbare“ Maschine vor.

Initialen-Schreiber

Nach der Ampel als relativ einfaches Beispiel für eine programmgesteuerte Anlage wollen wir uns nun einer größeren

Herausforderung stellen: Zu bauen ist eine Maschine, die rein elektromechanisch die Initialen von jemandem, also die Anfangsbuchstaben von dessen Vor- und Nachnamen, auf ein Blatt Papier schreiben kann. Was da genau geschrieben wird, soll von einem Programm gesteuert werden, das auf einem austauschbaren Datenträger gespeichert ist. Je nach eingelegtem Programm schreibt die Maschine dann beliebige Zeichen aufs Papier.

Das klingt ganz schön nach Computerei, oder? Dennoch werden wir das schaffen. Allerdings benötigt das Modell schon eine recht komfortable Ausstattung an elektromechanischen Teilen. Insgesamt werden wir verbauen (siehe Bauanleitung^[3]):

- Drei Motoren,
- sechs Taster,
- einen Elektromagneten sowie
- ein Relais.

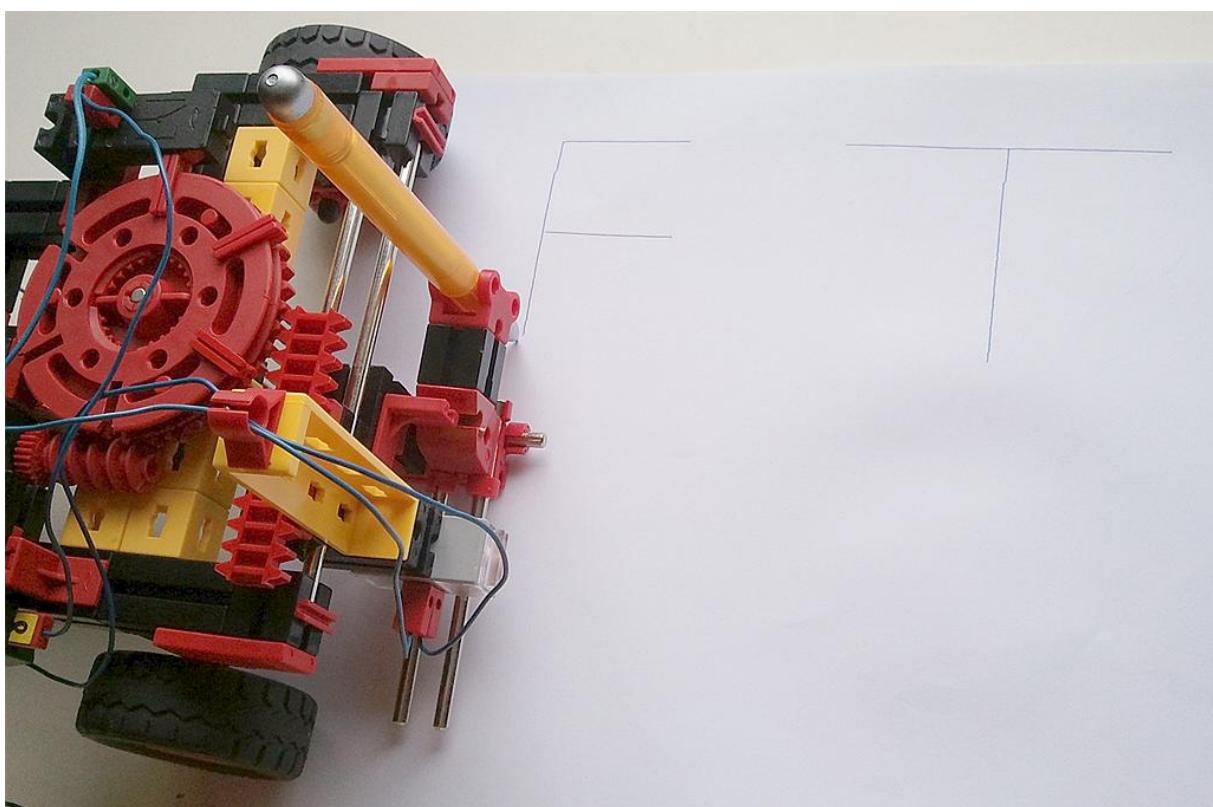


Abb. 9: Der Initialen-Schreiber hat „FT“ geschrieben

Das Relais kann eines der fertigen fischertechnik-Relaisbausteine sein oder alternativ (dazu brauchen wir aber einen zweiten Elektromagneten) ein ft:pedia-Selbstbaurelais, wie es in [Ausgabe 4/2011](#) ausführlich beschrieben ist.

Die Funktionsweise des Schreibers

Zunächst müssen wir aber festlegen, wie die Buchstaben zu Papier kommen sollen. Unser Modellvorschlag enthält einen Wagen mit folgenden Eigenschaften:

1. Er wird einfach auf das zu beschreiben-de Blatt Papier gestellt und kann darauf

Schritte mit einer bestimmten Länge fahren. Das Papier bleibt also einfach auf der Tischplatte liegen, und der Wagen fährt in einer Richtung darauf hin und her.

2. Quer zur Fahrtrichtung lässt sich ein Stift-Träger verschieben. Auf diese Weise können wir den Schreibstift auch senkrecht zur Fahrtrichtung verschieben. Auch das muss in definierten Schritten geschehen, damit die Maschine immer an einem bekannten Ort steht und also den nächsten korrekt anfahren kann.

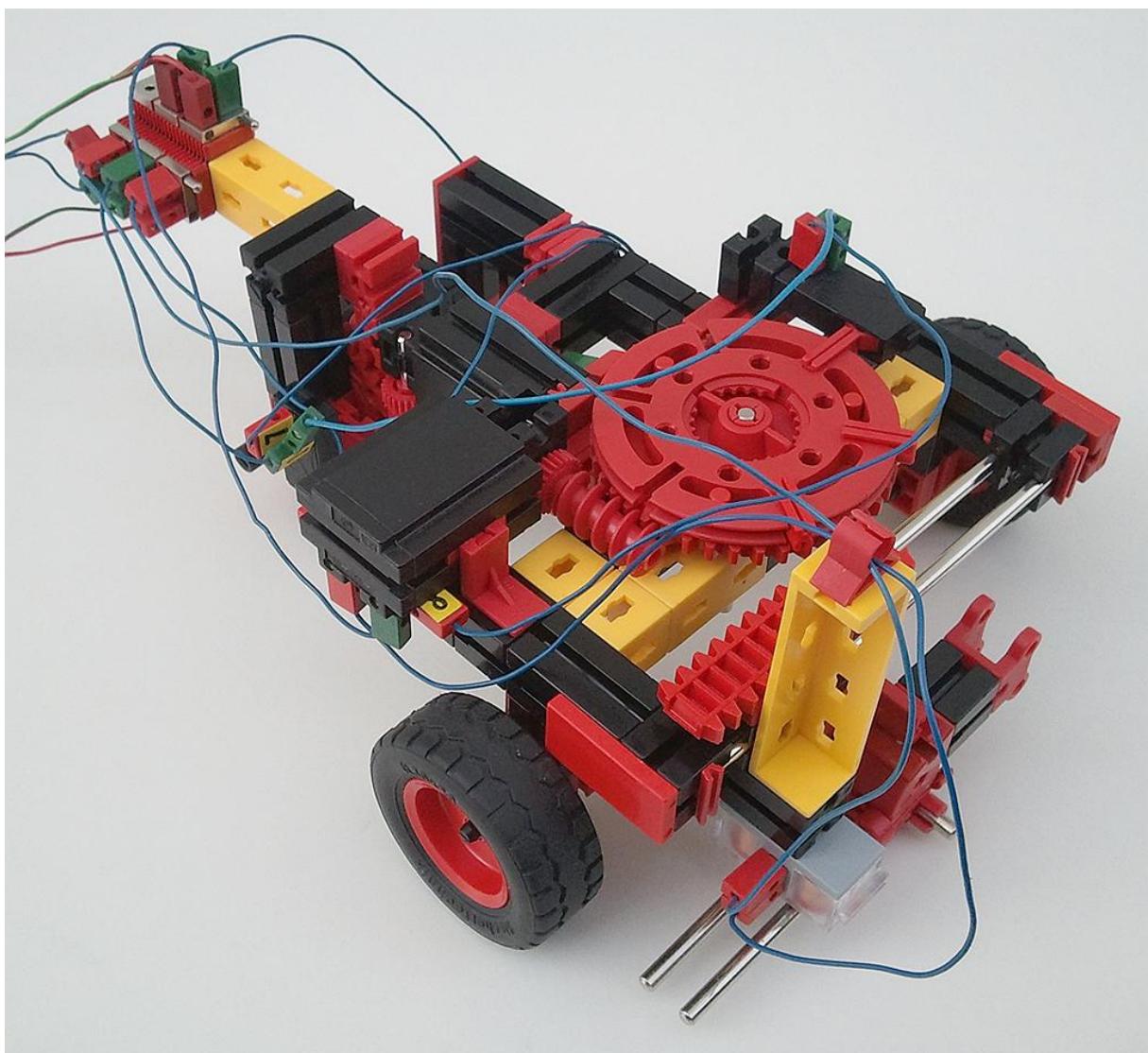


Abb. 10: Der Schreibwagen

3. Schließlich kann der Stift abgehoben oder aufs Papier niedergedrückt werden.

Dazu dient der eine Teil unseres Gesamtmodells, der Schreibwagen. Es ist ein Dreiradfahrzeug mit angetriebenem Hinterrad.

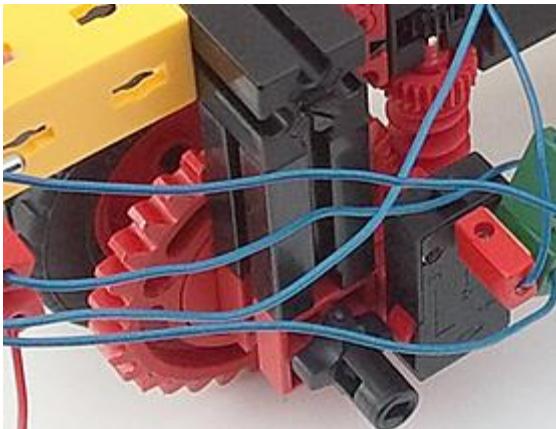


Abb. 11: Hinterradantrieb

Vorne kann ein Stifträger auf einem querliegenden Achsenpaar per Zahnstangenantrieb quer zur Fahrtrichtung verschoben werden.

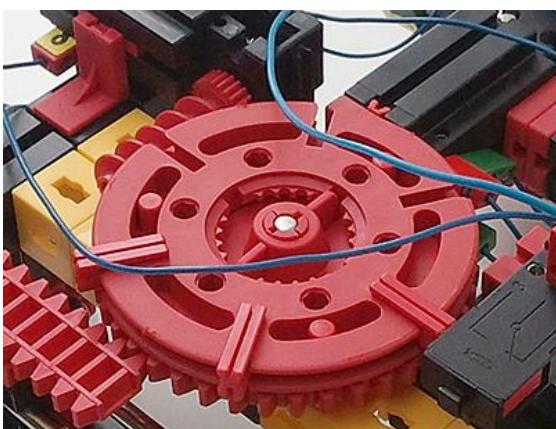


Abb. 12: Verschiebung des Stifträgers

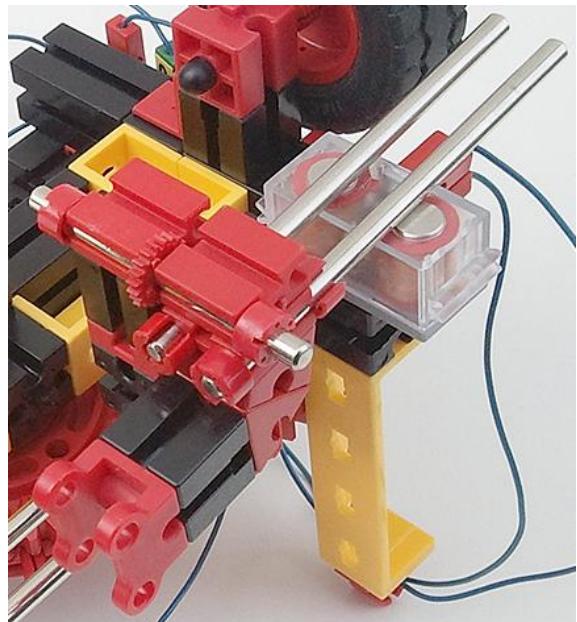


Abb. 13: Detailansichten des Stifträgers

In den Stifträger kann ein Schreibstift eingeklemmt werden (je nach Dicke des Stifts müsst ihr die Konstruktion hierfür vielleicht etwas anpassen).

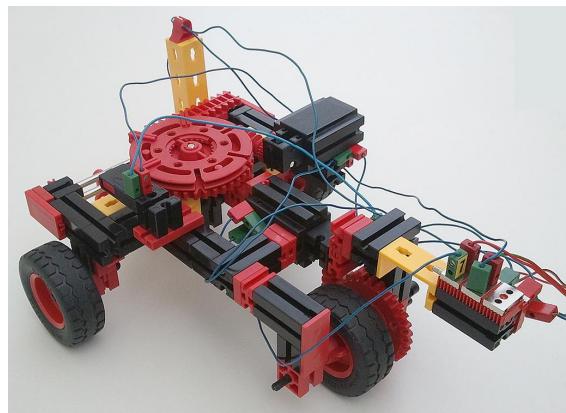


Abb. 14: Gesamtansicht Rückseite

Um die Kabel zum noch ausstehenden Steuergerät komfortabel anschließen zu können, sind im Modellvorschlag zwei dreipolige [Verteilerplatten](#) aus dem älteren ft-Elektromechanik-Programm angebaut. Wer keine hat, kann dasselbe auch mit drei leeren [Leuchtsteinsockeln](#) erreichen oder eben gleich entsprechend lange Kabel verwenden. Insgesamt werden zwischen Schreibwagen und Steuergerät fünf Leitungen benötigt werden.

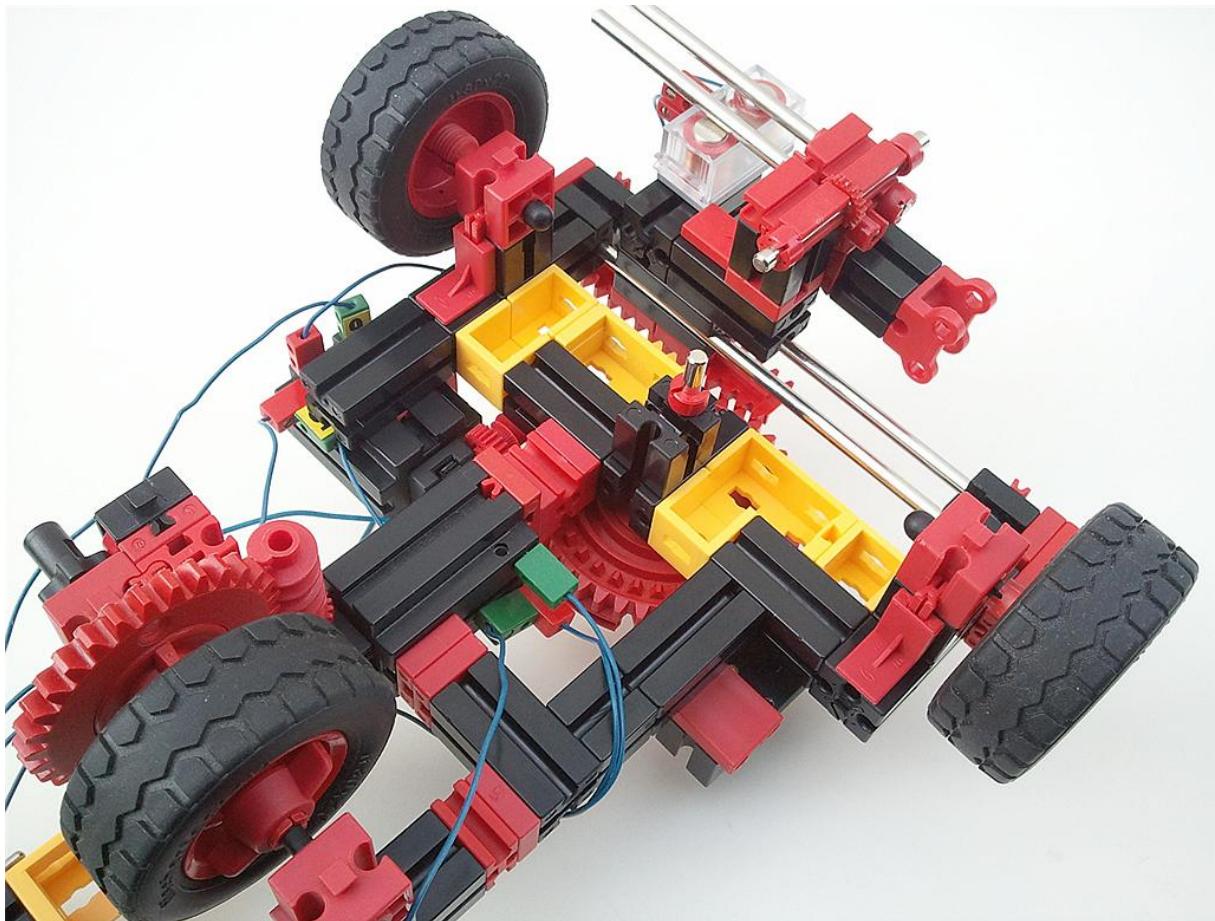


Abb. 15: Die Unterseite des Schreibwagens

Damit das Schreiben der Buchstaben richtig funktioniert, definieren wir die Operationen, die die Maschine ausführen können muss:

1. Sie muss einen Schritt auf dem Papier fahren können, z. B. für die waagerechten Linien des „F“.
2. Sie muss den Stift quer zur Fahrrichtung einen Schritt versetzen können, z. B. für die Distanz zwischen den beiden waagerechten Linien des „F“.
3. Sie muss diese beiden Bewegungen in beide Richtungen ausführen können: nach rechts und auch zurück nach links fahren, sowie den Stift in Richtung Ober- und Unterkante der Buchstaben verschieben.
4. Sie muss all diese Operationen sowohl mit abgehobenem Stift durchführen

können (wenn Leerraum überfahren werden muss) als auch mit abgesenktem (wenn tatsächlich eine Linie zu zeichnen ist).

Die Maschine arbeitet also ähnlich wie ein computergesteuerter Plotter. Wir brauchen zum Zeichnen von Buchstaben oder Ziffern aber nur eine ganz geringe „Auflösung“: Es genügen wenige Schritte in waagerechter Richtung bis zum Papierende, und es genügen zwei Schritt-längen quer zur Fahrtrichtung. Das ermöglicht es uns, gänzlich ohne „richtigen“ Computer auszukommen, wie wir noch sehen werden.

Allerdings müssen wir die Schritt-längen schon hinreichend genau einhalten können, damit beim Zurückfahren auch wieder genau genug die Ausgangsposition angefahren werden kann. Das sehen wir, wenn

wir uns anschauen, wie das „F“ zu zeichnen ist:

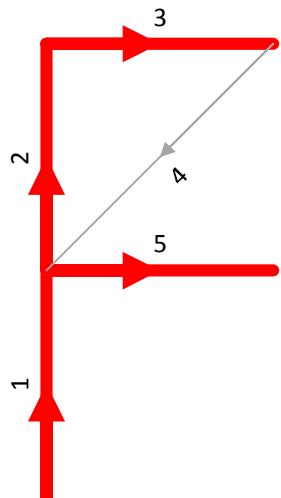


Abb. 16: Zeichnen eines „F“

Wir fahren also in folgender Reihenfolge von der Ausgangsposition (in der der Stift am linken unteren Eck stehen soll):

1. Zwei Schritte mit abgesenktem Stift quer zur Fahrtrichtung nach oben.
2. Mit immer noch abgesenktem Stift einen Schritt nach rechts fahren.
3. Mit angehobenem Stift einen Schritt zurück nach links unten. Hier ist es wichtig, die Position hinreichend genau zu treffen, damit der nächst Querstrich sauber beginnt.
4. Schließlich einen Schritt nach rechts fahren bei abgesenktem Stift, um das „F“ zu vollenden.

Damit wir die Schritte in definierter Länge zeichnen, rüsten wir die beiden Motoren zum Fahren des Wagens und zum Verschieben des Stifts quer dazu mit je einem Positionstaster aus. Einer sitzt beim Heckmotor und wird von einem Impulszahnrad direkt auf der Antriebsachse betätigt. Der andere sitzt oben auf dem Fahrzeug und wird von der Z40/Drehscheiben-Kombination mit eingesteckten Verbindern 15 betätigt.

Die Grundschaltung jedes Motors sieht dann so aus:

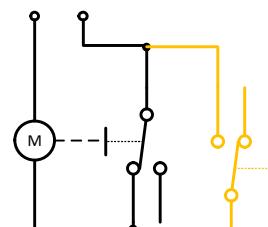


Abb. 17: Die Grundschaltung je Motor

Die beiden Anschlüsse oben sollen Strom liefern – für die jeweils benötigte Drehrichtung richtigerum gepolt. Der Motor läuft, solange sein Impulstaster nicht betätigt wird. Sobald er soweit gedreht hat, dass der Taster gedrückt wird, unterbricht dieser seine Stromzufuhr.

Damit erreichen wir, dass der Motor immer ganze, definiert lange Schritte ausführt. Der in der Schaltung orangefarbig eingezeichnete zweite Taster kann den Impulstaster überbrücken, um den Motor zum Loslaufen zu bewegen. Dieser braucht also nur kurz gedrückt zu werden (bis der Impulstaster losgelassen wird nämlich), um einen Schritt zu fahren. Diesen Taster müssen wir später noch geeignet ergänzen; er wird nicht auf dem Fahrzeug sitzen, sondern in einem externen Steuergerät.

Am besten baut ihr den Schreibwagen anhand der Abbildungen auf und erprobt diese Schaltung ausführlich in beiden Drehrichtungen für beide Motoren. Erst wenn alles gut funktioniert, wenden wir uns dem nächsten Schritt zu.

Die Steuerung

Ganz getrennt vom Schreibwagen müssen wir noch eine geeignete Steuerung bauen. Die muss folgendes können:

- Sie muss den Stift nach Bedarf absenken oder anheben können.

- Sie muss die Taster zur Überbrückung der Impulstaster im Wagen geeignet ansteuern, um die Motoren einen Schritt fahren zu lassen.
- Schließlich muss sie die Stromversorgung der Motoren in der jeweils gerade benötigten Polung bereitstellen, damit die Motoren jeweils in die richtige Richtung fahren.
- Das Steuergerät muss seine Befehle von den Datenträgern mit dem jeweils auszuführenden Programm lesen und korrekt umsetzen können.

Überlegen wir zunächst mal, wie wir überhaupt die „Datenträger“ mit ihrem „Programm“ darauf realisieren können. Dazu spicken wir ein wenig in der Geschichte der Datenverarbeitung und schauen uns eine *Lochkarte*^[4] der ersten Computer an:

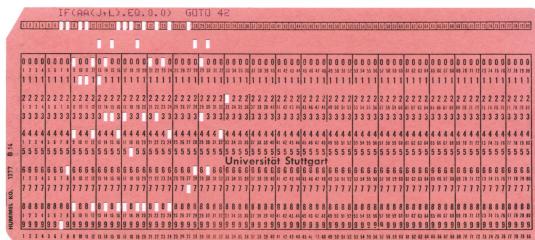


Abb. 18: Lochkarte (Quelle: Wikipedia)

Solche Karten wurden in den 1960er und 1970er Jahren verwendet, um Daten und Programme einem Computer bereitzustellen zu können. Ein geeignetes Gerät, der *Lochkartenleser*, tastete die Lochkarte Spalte für Spalte ab. Jedes in die Karte gestanzte Loch stand für ein bestimmtes Signal (ein „Bit“), welches nur zwei Zustände annehmen konnte: Loch gestanzt oder Loch nicht gestanzt. Durch die Kombination mehrerer solcher Bits (nämlich alle in einer Spalte der Lochkarte) konnten auch mehr Informationen als nur zwei verschiedene dargestellt werden.

Na, das können wir doch auch! Nehmen wir doch, um nur eine Möglichkeit vorzu-

stellen, eine schwarze Bauplatte 90 • 120, und stecken da in einem bestimmten Muster Teile drauf. Wie wir im Folgenden noch entwickeln werden, benötigen wir für das Programm zum Schreiben von „FT“ mindestens zwei Platten, von denen wir sowohl die Vorder- als auch die Rückseite verwenden.

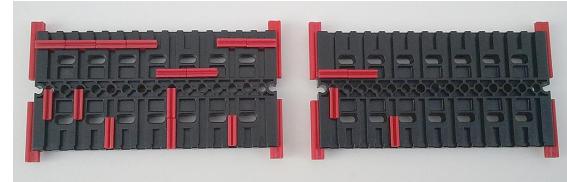


Abb. 19: Vorderseite der Karten: Das „F“

Für unsere Programm-Karten verwenden wir, wie ihr seht, Federnocken. Warum wir für unser Modell die oberen beiden Reihen waagerecht und die unteren beiden senkrecht ausgerichtet bestücken, wird später noch klar werden.

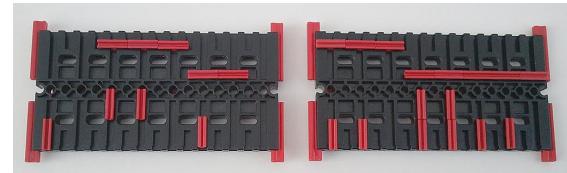


Abb. 20: Rückseite der Karten: Das „T“ und die Rückkehr in die Ausgangsposition

„Teil vorhanden“ entspricht dann bei der Lochkarte „Loch gestanzt“, „Teil nicht vorhanden“ entspricht „Loch nicht gestanzt“. Und das Beste: Durch einfaches Umstecken können wir – anders als bei echten Lochkarten – unser „Programm“ jederzeit ändern, ohne eine neue Karte zu benötigen.

Die Karten müssen von einem noch zu bauenden Lesegerät von links nach rechts spaltenweise gelesen werden. Immer vier übereinander liegende Positionen (eine Spalte) legen zusammen einen Befehl fest, den die Maschine ausführen soll. Entsprechend der Aufgabenstellung, die das Steuergerät erfüllen muss, verwenden wir also vier „Bits“. Jede Spalte unserer Karten enthält vier Positionen, an denen

eine Federnocke eingeschoben oder eben nicht eingeschoben sein kann.

Was genau bedeuten nun die Teile auf der Karte? Wir definieren:

1. Bit 1: Die von oben gesehen erste Position sagt aus, ob der Stift aufs Papier gedrückt (Federnocke vorhanden) oder abgehoben (keine Federnocke) sein soll.
2. Bit 2: Die zweite Position von oben gibt die Polung für die Stromversorgung und damit die Drehrichtung der beiden Motoren im Schreibwagen an. Ist dort keine Federnocke eingeschoben, sollen das Fahrzeug nach rechts und/oder der Stifträger nach oben fahren. Steckt eine

Federnocke in dieser Position, soll die Bewegung in die andere Richtung gehen, also zurück nach links bzw. unten.

3. Bit 3: Die dritte Position von oben, also die zweite von unten gesehen, gibt bei Bestückung mit einer Federnocke ein Signal, das den Impulstaster des Stifträgers überbrückt. Wenn das Lesegerät hier eine Federnocke erkennt, wird es also bewirken, dass der Stift genau einen Schritt nach oben bzw. (je nach Polung durch Bit 2) nach unten bewegt wird – es überbrückt kurz den Impulstaster, bis der Motor von alleine weiter läuft.

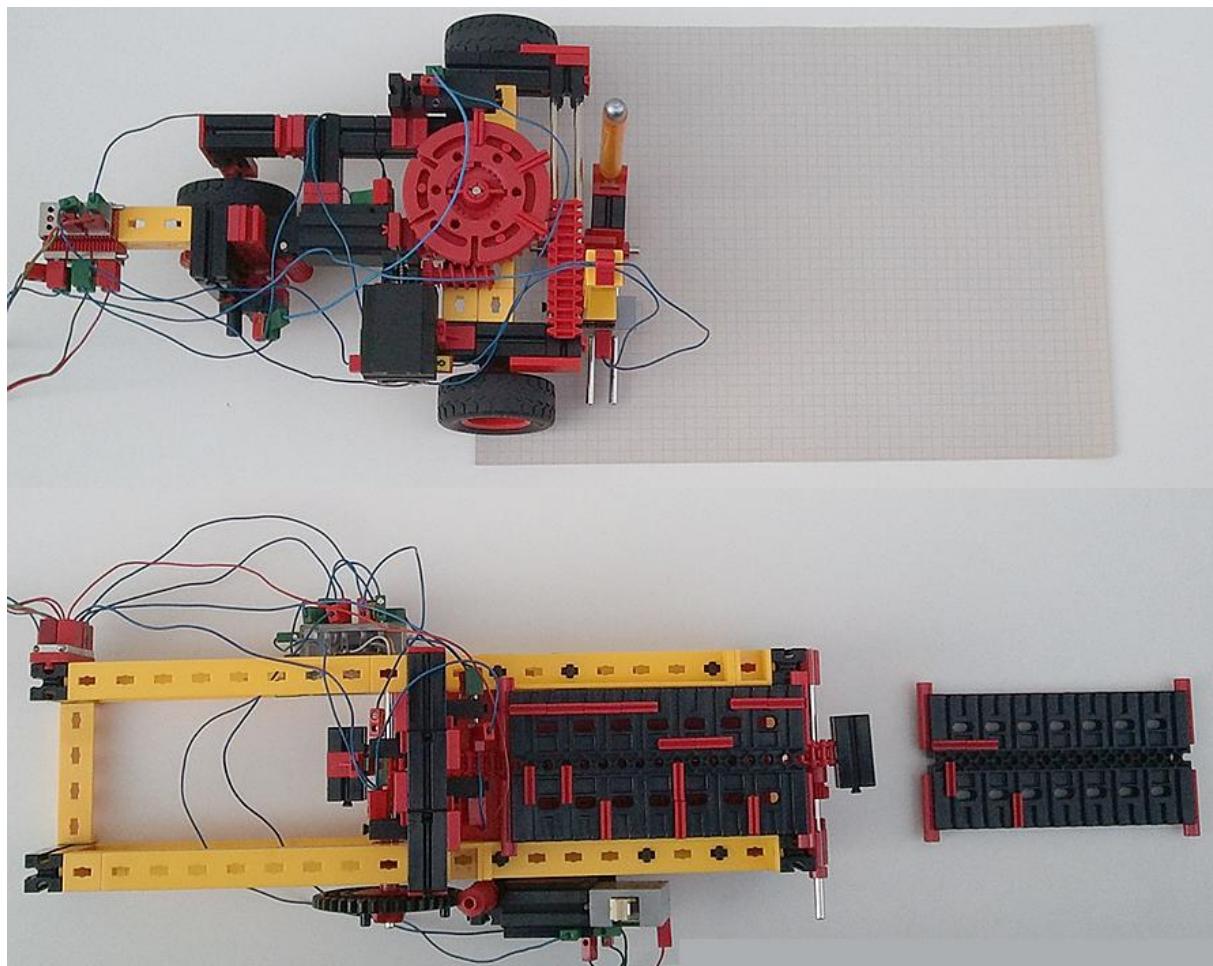


Abb. 21: Gesamtansicht auf Schreibwagen, Steuergerät und Programmkarten

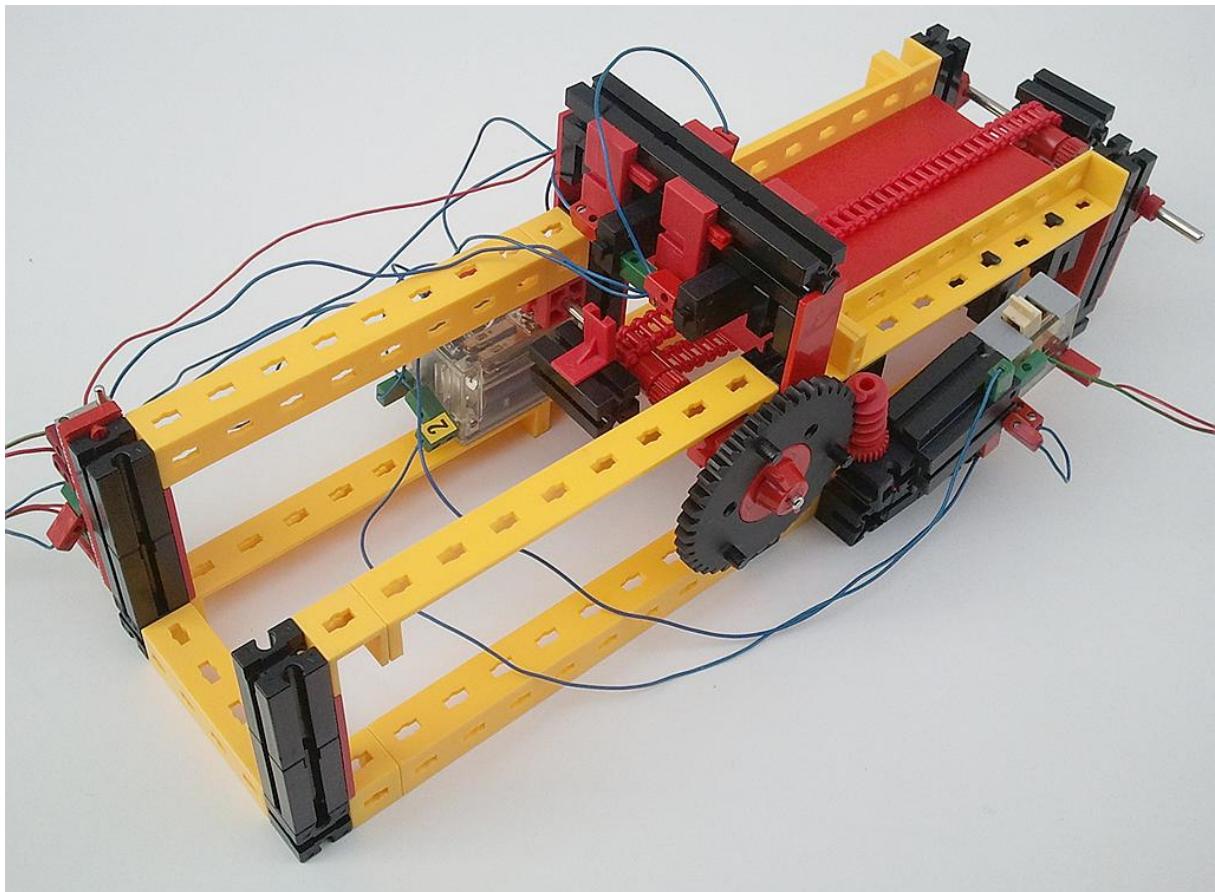


Abb. 22: Das Steuergerät mit dem Programmkarten-Leser

4. Bit 4: Das unterste Bit schließlich wirkt wie Bit 3, nur für den Fahrmotor des ganzen Wagens. Es stößt mit Federnocken also das Verfahren des ganzen Wagens um genau einen Schritt nach rechts bzw. (je nach Polung durch Bit 2) nach links an.

Das Steuergerät

Damit wir unsere Programmkarten automatisch lesen können, legen wir sie in ein Steuergerät ein, welches sie per Kettenzug langsam durch eine Leseeinheit schiebt. Auf diese Weise wird Spalte für Spalte abgetastet, an welchen Positionen Federnocken stecken und wo nicht. Insgesamt sieht die Anlage dann so aus wie in Abb. 21 dargestellt:

- Der Schreibwagen steht auf einem DIN-A4-Blatt.

- Das Steuergerät steht separat und ist mit dem Schreibwagen nur über eine Reihe von Kabeln verbunden.
- Im Steuergerät ist die erste Programm-karte so eingelegt, dass ihre Oberseite von links nach rechts abgetastet und damit abgearbeitet wird.

Die Programmkarten werden von einem Motor durch die Leseeinheit geschoben, wie die nächsten Bilder zeigen. Die Leseeinheit selber besteht aus vier Tastern und ist in der folgenden Abb. 23 auf dem Kopf liegend dargestellt.

Die Taster sind mit je einem [Baustein 5 • 15 • 30](#) und einem Verbinder 15 an einem Querbalken hängend befestigt. Die Taster zeigen also im Betrieb nach unten und lesen den durch die Federnocken gebildeten Programmcode von der Oberseite der durchs Lesegerät geschobenen

Programmkarte. Alle vier Taster sind genau im Raster angeordnet, passend zu den Federnocken auf den Programmkarten.

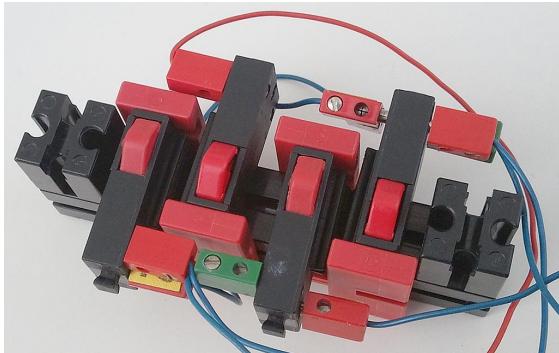


Abb. 23: Die Leseeinheit

Die Leseeinheit ist von oben in zwei seitlich am Steuergerät hoch ragenden Bauplatten $15 \cdot 60$ eingeschoben. Das erlaubt die genaue Justierung, denn die Taster müssen von den Federnocken der Programmkarten zuverlässig gedrückt werden, dürfen sich aber nicht daran verhaken und den Transport der Programmkarten behindern.

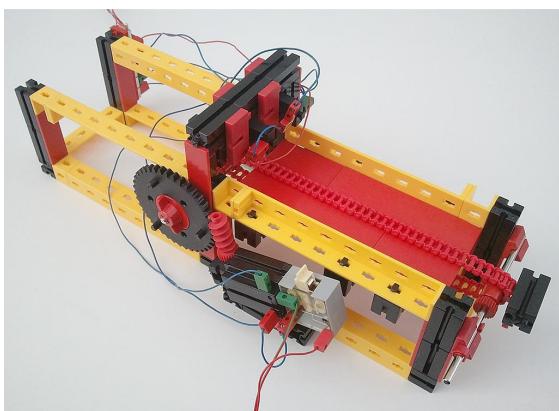


Abb. 24: Vorderseite des Steuergerätes

Der Motor treibt eine Kette an, auf der an zwei Stellen (eine würde zur Not auch genügen) Bausteine 30 und zur Verstärkung ein [Winkelstein 10 • 15 • 15](#) auf einem [Förderkettenglied](#) angebracht sind. Zur Befestigung des Motors dienen ein Winkelstein 15° , zwei Winkelsteine $7,5^\circ$ und zwei Federnocken.

Im Modellvorschlag ist direkt am Motor ein Ein-/Ausschalter für die gesamte Anlage angebracht. Das könnt ihr natürlich

ganz nach Eurem Bestand an Schaltern auch anders lösen.

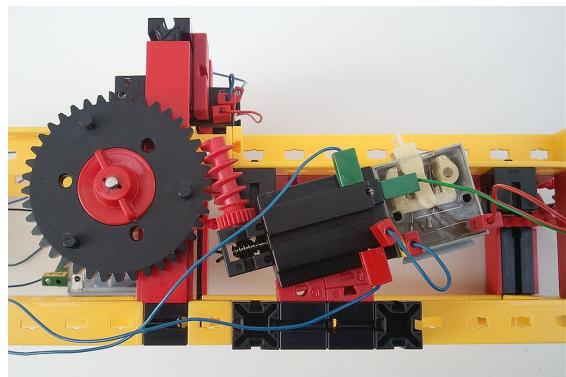


Abb. 25: Detailaufnahme der Motoraufhängung

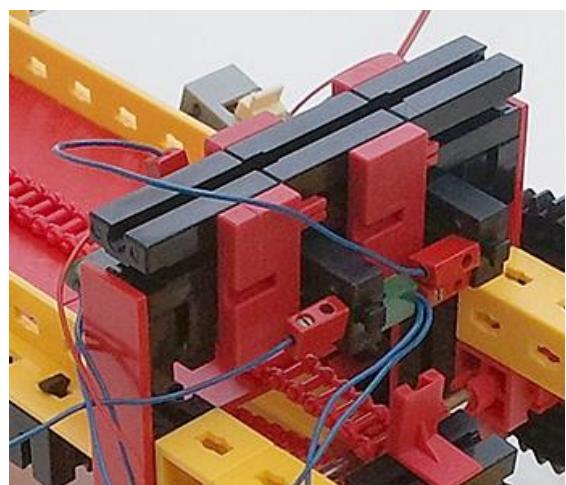
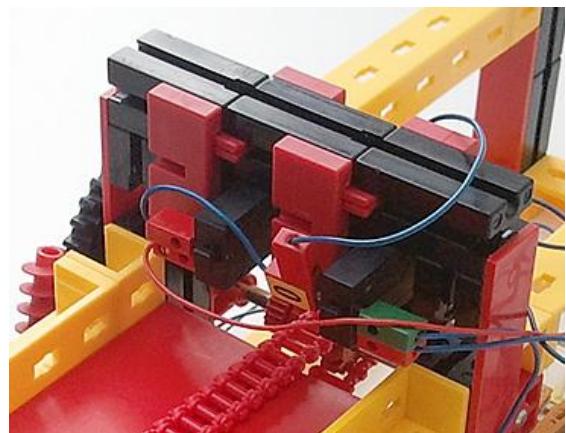


Abb. 26: Detailansicht des Steuergerätes

Auf der Rückseite des Steuergerätes haben wir auch gleich ein [em-5-Relais](#) angebracht, welches wir für die elektrische Schaltung brauchen werden. Jeder andere fischertechnik-Relaisbaustein, jedes andere Relais mit Betätigung durch 9 V Gleich-

spannung und mit zwei Umschaltkontakte oder auch das ft:pedia-Selbstbaurelais aus [ft:pedia 4/2011](#) können natürlich genauso verwendet werden.

Die Programmkartens werden also einfach oben auf das Steuergerät gelegt und automatisch durch die Leseeinheit geschoben. Damit die Antriebskette nicht durchhängt, wird sie von Bauplatten 30 • 90 und 30 • 45 von unten unterstützt (Abb. 28).

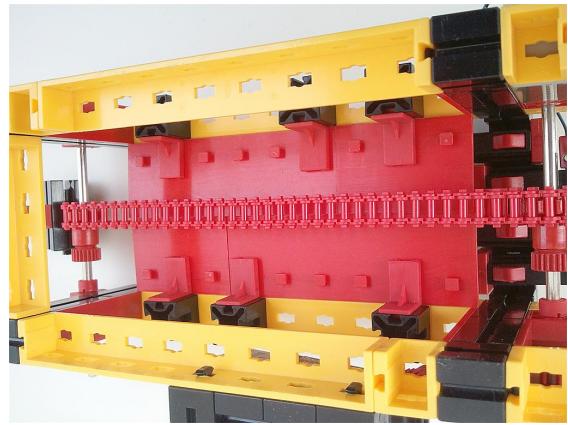


Abb. 28: Befestigung der Stützplatten für die Antriebskette auf der Unterseite

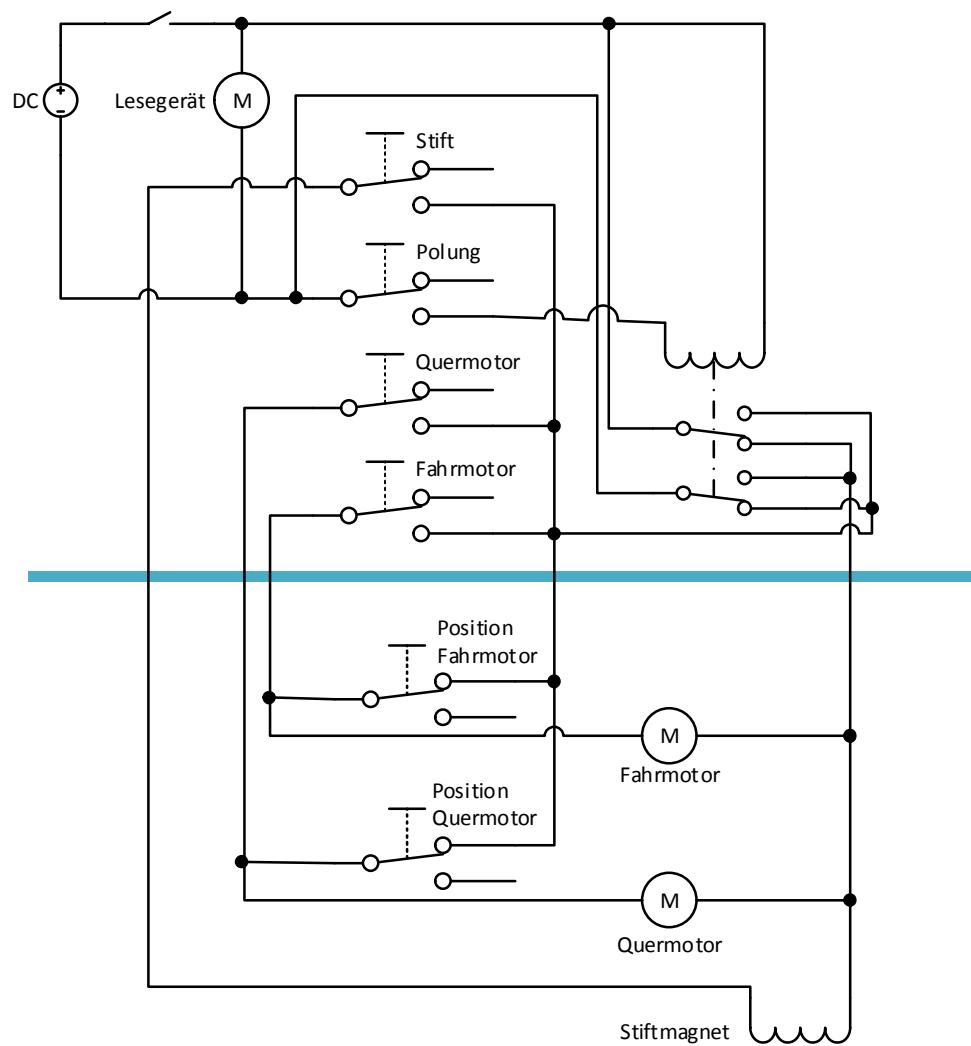


Abb. 27: Vollständiger Schaltplan des Initialenschreibers

Die Schaltung des Initialenschreibers

Abb. 27 zeigt die elektrische Schaltung der gesamten Anlage. Alles oberhalb des blauen Balkens gehört zum Steuergerät, alles darunter zum Schreibwagen. Wie man sieht, benötigen wir fünf lange Leitungen zwischen den beiden Komponenten. Aber der Reihe nach:

Wenn am Schalter der Strom eingeschaltet wird, läuft der Antriebsmotor für die Kette des Lesegerätes die ganze Zeit. Nun legen wir die erste Programmcarte mit der abzutastenden Seite nach oben auf den Kettenbereich des Lesegerätes. Sie wird von den an der Kette angebrachten Mitnehmern durchs Lesegerät mit den vier Tastern getrieben.

Fangen wir mit dem Polungstaster (Bit 2) an: Sobald der gedrückt ist, fließt Strom durch das Relais, welches also anzieht. Durch die Beschriftung der beiden Wechselkontakte des Relais wirkt es wie ein Polwenderschalter (siehe die vorherigen Ausgaben der ft:pedia). Das Relais stellt also Strom in einer oder anderen Polung bereit, je nach Betätigung des Polungstasters.

Der so in seiner Richtung festgelegte Strom kann nun zu den Fahr- und Quermotoren des Schreibwagens und wieder zurück fließen, und zwar genau dann, wenn der zugehörige Taster für Fahr- bzw. Quermotor (Bit 3 bzw. 4 der Leseeinheit) betätigt sind.

Es genügt aber schon ein kurzer Impuls (durch einen quer zur Leserichtung eingesetzten Federnocken nämlich), um die Motoren zu einem ganzen Schritt zu bewegen. Sobald der Fahr- oder Quermotor sich nämlich etwas bewegt, geben sie ja ihren jeweiligen Impulstaster frei. Sobald der aber losgelassen wurde, überbrückt er über seinen Ruhkontakt den Lesetaster des jeweiligen Motors. Deshalb läuft der auch bei einem kurzen Impuls am

Lesetaster einen vollständigen Schritt weiter, bis er seinen Impulstaster wieder betätigt. Dann bekommt der Motor wieder keinen Strom mehr und bleibt stehen.

Bleibt uns nur noch der Stift. Damit wir möglichst wenige Leitungen zwischen Steuergerät und Schreibwagen brauchen, lassen wir den Elektromagneten für den Stift vom selben Strom speisen wie die Fahr- und Quermotoren. Durch die Wiederverwendung der evtl. umgepolten Spannung am Ausgang des Relais sparen wir zwei Stromzuführleitungen. Dem Elektromagneten ist es ja auch egal, wie herum der Strom durch ihn durchfließt. Die Metallachsen zur Bewegung des Stiftes zieht er in jedem Fall an.

Und das ist schon die gesamte Schaltung! Für die komplexen Operationen, die unsere Maschine ausführen kann, ist das wahrlich wenig Aufwand. Wir haben nicht mal Elektronik dafür benötigt, ganz zu schweigen von einem Computer.

Das F und das T

Jetzt können wir auch das Programm besprechen, welches durch Stecken von Federnocken auf die Bauplatten gespeichert wird, und das uns ja – in unserem Beispiel – ein „F“ und ein „T“ zeichnen soll. Selbstverständlich könnt ihr auch andere Buchstaben oder auch Ziffern stecken. Probiert, testet, tüftelt!

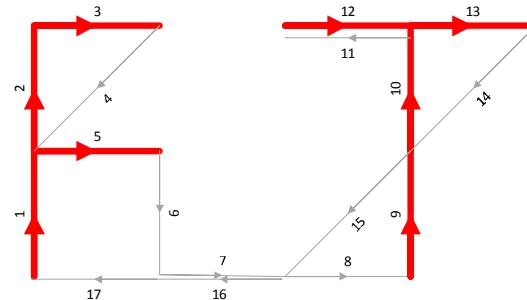


Abb. 29: Der komplette Zeichenvorgang für „FT“ und Rückkehr zum Ursprung

Insgesamt muss der Schreibstift für das Schreiben von „FT“ die Bewegungen laut

Abb. 29 in der angegebenen Reihenfolge ausführen.

Wir fangen also links unten an. Achtet darauf, dass der Quermotor den Stift wirklich auf der unteren Position stehen hat, bevor ihr die Maschine anschaltet!

Eine Vergrößerung von Abb. 19 zeigt die ersten Schritte:

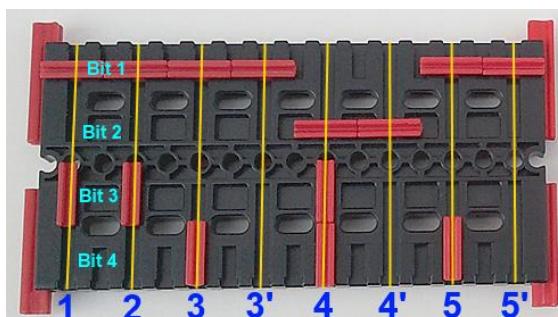


Abb. 30: Die erste Programm-Karte des „F“

In Schritt 1 – die Karten werden ja von rechts ins Lesegerät geschoben und also von links nach rechts abgetastet – hat also Bit 1 und 3 gesetzt. Es wird also der Stift abgesenkt (Bit 1), und ein Fahrimpuls geht an den Quermotor. Da Bit 2 nicht gesetzt ist, bleibt das Relais unbetätigt, und der Quermotor schiebt den abgesenkten Stift um einen Schritt nach oben. Linie 1 des „F“ ist gezeichnet!

Jetzt leuchtet auch ein, warum die oberen beiden Bits mit Federnocken in Leserichtung versehen werden, während die für die Motoren (Bit 3 und 4) quer dazu gesteckt werden: Wenn wir einen Fahrimpuls auf die Motoren geben, müssen ja sowohl die Polung für die Fahrtrichtung bereits korrekt eingestellt und auch der Stift bereits abgesenkt oder angehoben sein. Deshalb lassen wir Stift und Polung einfach zeitlich etwas vorher korrekt schalten, bevor die Fahrmotoren ihren Fahrimpuls bekommen.

Schritt 2 läuft wie Schritt 1: Der immer noch abgesenkten Stift wird einen weiteren Schritt nach oben bewegt. Damit ist die linke Seite des „F“ fertig. Noch weiter nach oben kann die Mechanik nicht fahren,

der Motor würde rattern – das wäre dann wohl ein „Programmfehler“, wenn wir drei Schritte nach oben stecken würden.

Für die nächste Linie muss der Stift immer noch abgesenkt bleiben, aber der Fahrmotor muss einen Schritt nach rechts machen. Das erledigt die dritte „Anweisung“ auf unserer Programmkkarte.

Die nächste Bewegung (4) ist die um einen Schritt nach links unten. Damit der Fahrmotor aber genügend Zeit hat, seinen Schritt ganz bestimmt zu vollenden, fügen wir einen Schritt 3' ein, in dem einfach nur der Stift noch eine Zeitlang abgesenkt bleibt.

Erst jetzt kommt Schritt 4: Der Stift wird abgehoben, weil der Magnet keinen Strom mehr bekommt. Das Relais wird durch das erstmals gesetzte Bit 2 umgepolzt. Und beide Motoren bekommen gleichzeitig einen Impuls zum Losfahren. Das bewirkt, dass der Schreibstift in abgehobenem Zustand gleichzeitig sowohl einen vollständigen Schritt zurück nach links als auch zurück nach unten fährt.

Damit die Motoren sicher Zeit haben, diesen Schritt zu beenden, ist wieder ein „Leerschritt“ eingefügt, in dem die Polung des Relais noch hinreichend lange umgeschaltet bleibt.

Nun kommt die letzte Linie des „F“ dran: Schritt 5 aktiviert den Stift, fährt einen Schritt nach rechts und gibt dem Motor wieder genügend Zeit, seine Bewegung zu vollenden, bevor der Stift wieder angehoben wird.

Jetzt könnt ihr den Rest unseres Programms bestimmt alleine „lesen“. Hier nochmal das gesamte Programm zum Schreiben von „FT“:

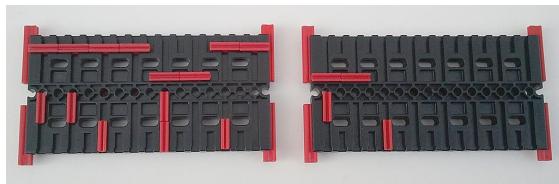


Abb. 31: Vorderseite der Karten: Das „F“

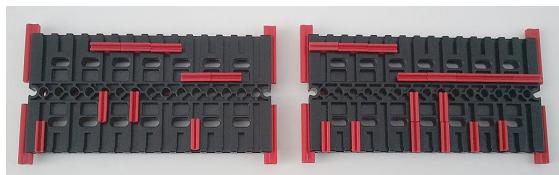


Abb. 32: Rückseite der Karten: Das „T“ und die Rückkehr in die Ausgangsposition

Vergleicht das Programm bitte mit dem Zeichenweg in Abb. 29. An mehreren Stellen sind wieder Pausen eingebaut, damit eine gestartete Bewegung korrekt vollendet werden kann, bevor der Stift oder die Polung der Motoren umgeschaltet wird. Ganz am Ende schließlich sind Programschritte eingefügt, die den Wagen und den Stift wieder zurück in seine Ausgangsposition bewegen.

Der gesamte Ablauf

Wenn ihr die Programmkarten wie gezeigt bestückt habt, muss die Maschine also wie folgt bedient werden:

1. Stellt sicher, dass der Quermotor den Stift in der unteren Endlage stehen hat.
2. Stellt sicher, dass der Stift so eingespannt ist, dass er das Papier in abgehobenem Zustand nicht berührt, in abgesenktem Zustand aber zuverlässig zeichnet. Zur Justage dient auch der kleine Winkelstein $10 \cdot 15 \cdot 15$ beim Stifträger. Mit dem könnt ihr genau einstellen, wie weit der Stift beim Abschalten des Magneten zurück fallen soll – aber nur so weit, dass der Magnet die Metallachsen noch anziehen kann!
3. Schaltet die Maschine ein. Die Transportkette muss in die Leseeinheit hinein loslaufen (sonst polt den Antriebsmotor um).

4. Wenn gerade kein Mitnehmer stört, legt die Programmplatte mit dem ersten Teil des „F“ auf die Förderstrecke rechts vom Lesegerät ein (vgl. Abb. 21).

5. Die erste Programmplatte wird abgearbeitet. Wenn sie durchgezogen wurde, könnt ihr gleich die zweite auflegen, um das „F“ vollenden zu lassen.

6. Dreht die erste Platte um und legt sie in die Maschine, um das „T“ zu beginnen. Tipp: Steckt die Verbinder 30 mit abgeflachter Seite in den Bauplatten so ein, dass sie immer in eine Richtung zeigen. Dann steckt man die Karten nicht so leicht verkehrt herum hinein.

7. Wenn sie durch ist, legt die zweite ebenfalls umgedreht ein, um das „T“ fertig zu stellen und die Maschine in die Ausgangsposition fahren zu lassen.

Wenn irgendetwas schief gehen sollte, könnt ihr die Maschine ja schnell mit dem Hauptschalter abschalten. Wenn alles richtig eingestellt, verdrahtet und justiert ist, läuft die Maschine fehlerfrei und zuverlässig.

Dann habt ihr natürlich ein weites Feld der „Programmierung“ vor euch! Vielleicht legt ihr euch eine „Programmbibliothek“ mit den zehn Ziffern an? Oder lasst die Anfangsbuchstaben eures Vor- und Nachnamens zeichnen? Wir wünschen euch jedenfalls viele spannende Experimente und recht viel Spaß beim Herstellen und Betrieb eures selbstgebauten „Schreibcomputers“!

Quellen

- [1] [Anleitung zum Ur-Elektromechanik-Kasten](#), fischerwerke 1968
- [2] [hobby 3 Begleitbuch Band 1](#), fischerwerke 1973
- [3] Thomas Kaiser, Sven Engelke: [Bauanleitung Initialenschreiber](#).
- [4] Wikipedia: [Lochkarte](#).

Tipps & Tricks

ft-Spezialteile made by TST (Teil 3)

Andreas Tacke

In einer lockeren Reihe stellt TST einige von ihm entwickelte Spezialteile vor, die so manche Lücke beim Bauen mit fischertechnik schließen. Nach seinen Antriebswellen mit Motor-Adaptoren ([ft:pedia 2/2012](#)) und seiner Lösung für die stabile Montage von Power-Motoren ([ft:pedia 3/2012](#)) folgt ein Spezialteil, das die Herzen von Roboter-Fans höher schlagen lassen dürfte.

In meinem heutigen Beitrag geht es um eine Alternative zum Greifzangengetriebe 32342. In den 80er Jahren gab es den Trainingsroboter-Baukasten (30572, Abb. 1).



Abb. 1: Trainingsroboter 30572

In diesem Baukasten gab es ein spezielles Getriebe zum Bau der Greifzange (Abb. 2). Dieses Getriebe gab es nur in diesem Baukasten, daher ist es heute sehr selten. Wer eines besitzt, gibt es auch nicht wieder her.

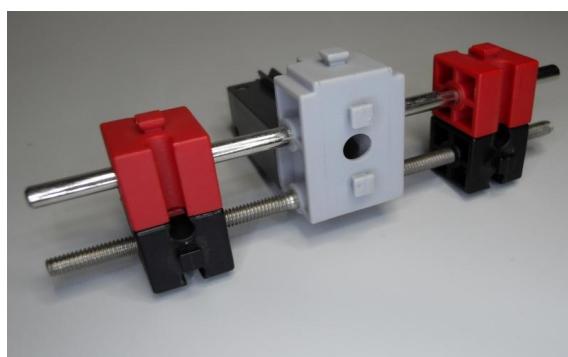


Abb. 2: Greifzangengetriebe 32342

Die Spindel besitzt auf der einen Seite ein Rechts- und auf der anderen ein Linksgewinde. Somit werden die Muttern zusammen oder auseinander gefahren. Der Baustein 15 mit Bohrung dient zur Führung und zur Verdrehssicherung.

Ich habe einen Minigetriebehalter hergenommen und für diesen eine Spindel mit Rechts- und Linksgewinde M4 angefertigt.



Abb. 3: Getriebehalter mit Spindel

Weiter stellte sich das Problem, dass die Bausteine mit den Gewindegelenken schwer zu bekommen waren – also selber machen. Dazu nahm ich den Baustein 32321 und versah diesen mit Gewindegelenken, einmal Rechts- und einmal Linksgewinde. Die mit Linksgewinde kennzeichnete ich durch einen blauen Punkt auf einem Zapfen.

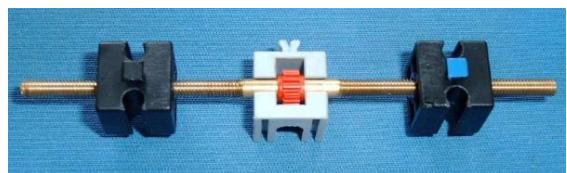


Abb. 4: Getriebehalter mit Muttern

Mit dem so modifiziertem Getriebebehalter lässt sich nun ähnlich wie mit dem Greifzangengetriebe 32342 eine Greifzange aufbauen.

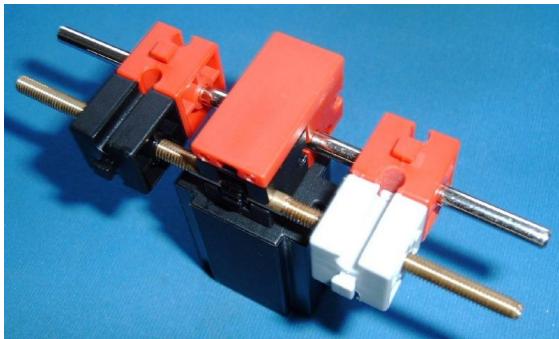


Abb. 5: Getriebebehalter montiert am Mini Motor

Durch den Anbau einer weiteren Führungsstange lassen sich auch wirklich große Greifzangen realisieren, in Abb. 6 z. B. mit zwei V-Grundplatten 45 x 45 x 5,5 zu sehen.

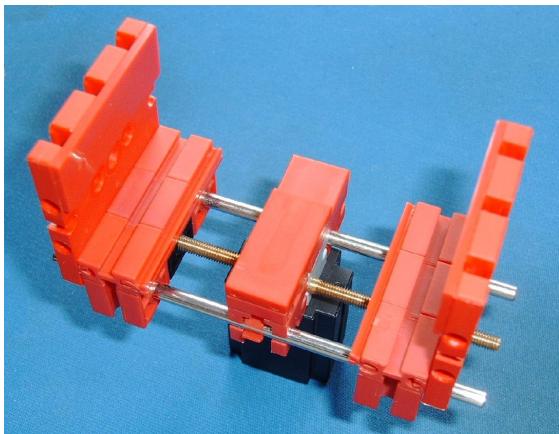


Abb. 6: Greifzange mit zwei V-Grundplatten 45x45x5,5.

Eine weitere sehr interessante Anbauvariante hat Jan Werner aufgezeigt. Er hat an seinem [5-Achs-Roboter](#) auch meinen Getriebebehalter für seine Greifzange benutzt. Diese Variante ist sehr kompakt und für diese Anwendung, finde ich, die perfekte Lösung (Abb. 7).

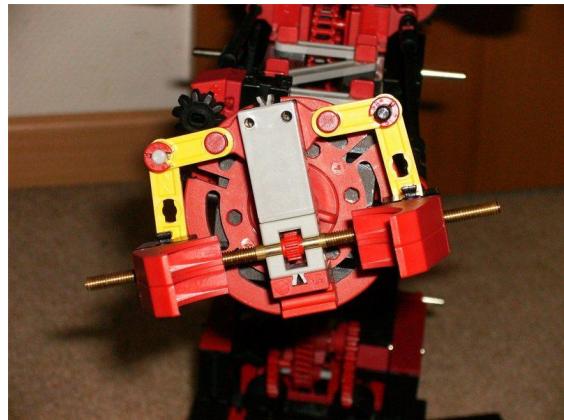


Abb. 7: Greifzange am 5-Achs-Roboter von J. Werner

Ansonsten kann man dieses Getriebe für verschiedene Anwendungen nutzen: immer da wo Teile linear auf und zu bewegt werden sollen, wie z. B. bei Schiebetüren.

Da sieht man mal wieder, wenn es etwas nicht mehr gibt, findet sich meistens eine Möglichkeit, diese Lücke durch einen Nachbau zu schließen...

fischertechnik-Basiswissen

Perlentauchen (Teil 3)

Stefan Falk

Nach Standardbauteilen und Elektromotoren setzen wir unsere fischertechnik-Zeitreise mit einem Querschnitt durch die vielfältigen Teile aus dem Bereich Elektromechanik fort.

Schon 1968 gab es die ersten Elektromechanik-Kästen von fischertechnik, und sie enthielten eine ganze Reihe spezieller Bauteile:

Schalter und Taster

Der als Einzelteil vorliegende Schalter war der [31331 Polwendeschalter](#) in grau (später gab es ihn auch in schwarz), der in jüngerer Zeit durch einen in ganz anderer Bauart (36708) abgelöst wurde:



Abb. 1: Alter und neuer Polwendeschalter

Der neuere (im ersten [Motor+Getriebe-Kasten](#) auch in grau) hat drei Schalterstellungen und dient der manuellen Steuerung eines Motors (aus/links/rechts). Der ältere hat nur zwei Schalterstellungen (links/rechts). Aber er hat drei besonders nützliche Eigenschaften:

Erstens kann er leichtgängig von einer Maschine umgeschaltet werden. Ein Beispiel gibt der [Elektromechanik-Kasten](#)

[von 1981](#) mit einem Fahrzeug, das automatisch umkehrt, sobald es an eine Wand stößt:

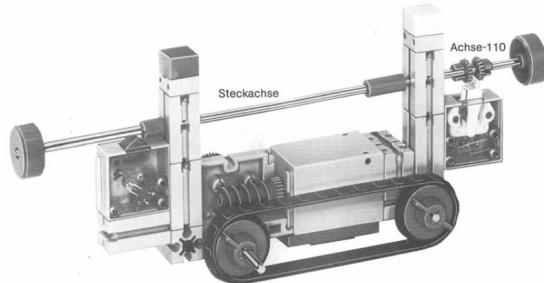


Abb. 2: Automatisch umkehrendes Fahrzeug

Eine andere Anwendung ist ein ständig hin- und her drehendes Rührwerk aus dem Begleitbuch [hobby 3 Band 2](#):

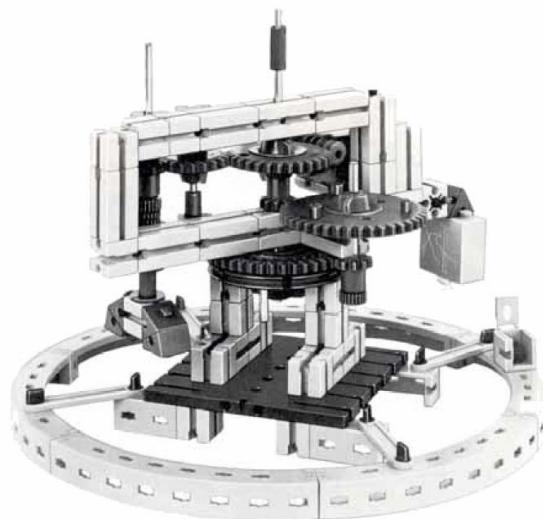


Abb. 3: Automatisch umpolendes Rührwerk

Am sich drehenden Rührwerk sitzt der Polwendeschalter, und immer wenn er aus einer Richtung an die senkrecht hoch-

stehende Lasche anstößt, wechselt der Motor des Rührwerks die Richtung.

Zweitens kann man durch das kleine Loch im Schalthebel (siehe den Pfeil in Abb. 1) ein dünnes Seil ziehen. Mit geschickt platzierten Knoten kann man so auch durch Seilzüge ein Umpolen bewirken. Das wurde z. B. bei einem Garagentor aus dem [Elektronik-Kasten ec](#) von 1972 ausgenutzt:

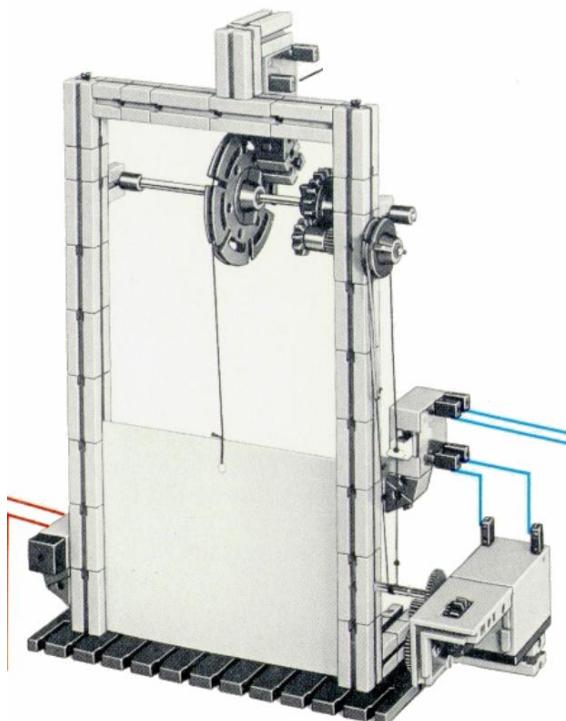


Abb. 4: Garagentor mit Umschaltung per Knoten im Seilzug

Und einen dritten praktischen Vorteil hat er gegenüber seinem Nachfolger: Die Stecker blockieren nur die untere Hälfte, daher kann man den Schalter z. B. neben einem BS30 in Längsrichtung verbauen.

Die Taster – die sich im Gegensatz zu Schaltern nach jeder Betätigung in den Grundzustand zurückstellen – haben eine interessante Entwicklung hinter sich. Der Ur-Taster war der 31332 von 1968. Etwas später wurde der ansonsten gleiche Taster mit einem dreieckigen Stößel hergestellt. Das hatte den Vorteil, dass er durch Maschinen von der Seite her in beide Richtungen leicht betätigt werden kann.

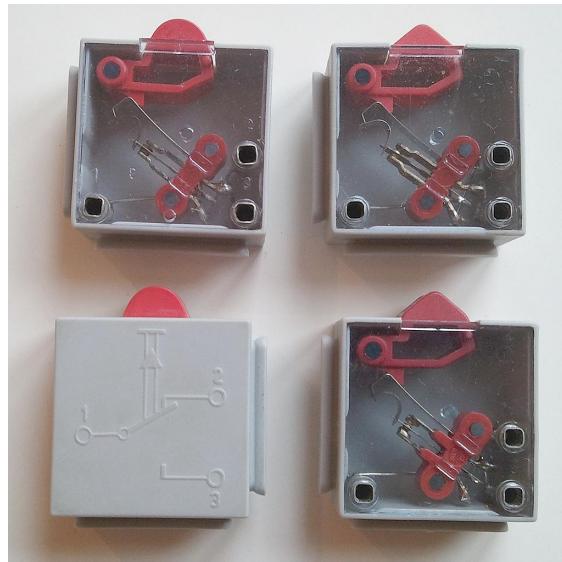


Abb. 5: Die großen Taster mit rundem und mit eckigem Stößel

Aber auch die Ur-Fassung kann man für etwas Besonderes ausnutzen, wie das [Clubheft 1972-1](#) mit einem Überrollschalter zeigt. Zieht man den Stab von rechts nach links durch die Mechanik, wird der Taster betätigt (Abb. 6).

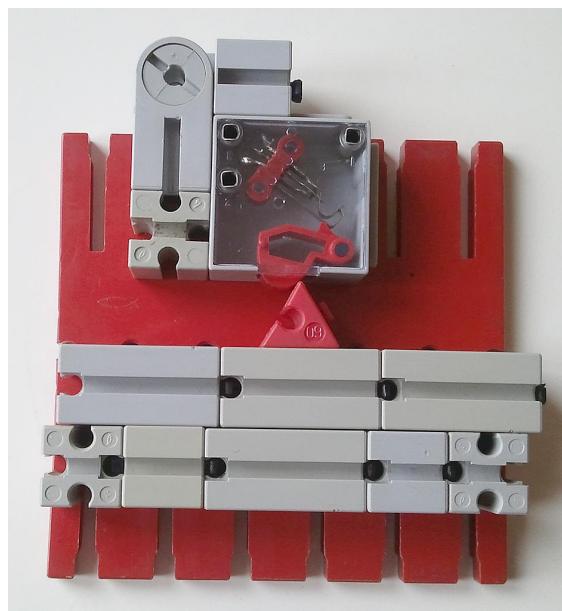


Abb. 6: Überrollschalter – von rechts nach links wird der Taster betätigt

Zieht man ihn dagegen von links nach rechts durch, wird der an einem Feder-gelenkstein (siehe [Perlentauchen Teil 2](#) in [ft:pedia 4/2012](#)) angebrachte Taster nur

nach oben gedrückt, ohne betätigt zu werden (Abb. 7).

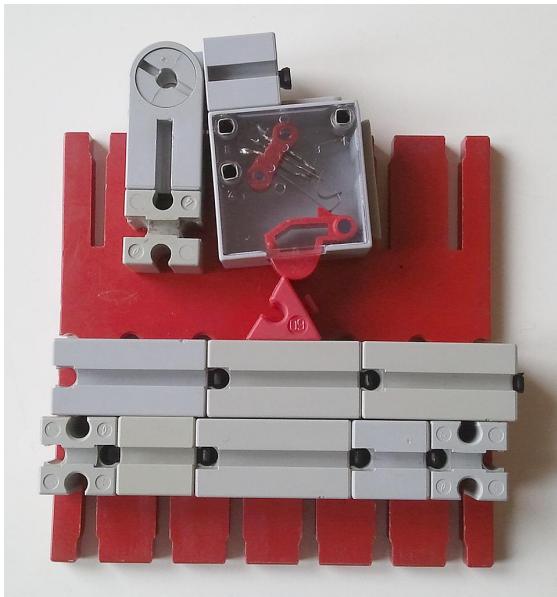


Abb. 7: Überrollschalter – von links nach rechts wird der Taster nur angehoben

Schließlich kam der heute bekannte [37780 Minitaster](#) heraus, zuerst mit Federn für Nuten, später mit Zapfen an den schmalen Seiten.

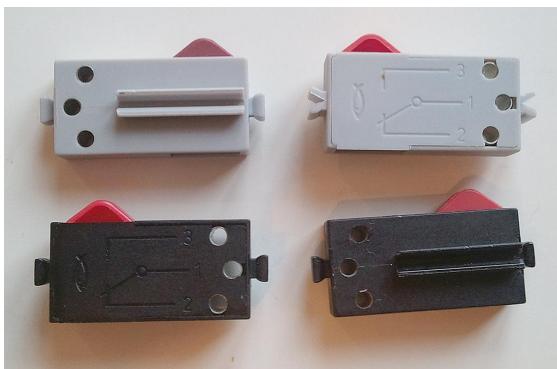


Abb. 8: Minitaster mit Zapfen und Federn

Die Minitaster sind zwar nicht mehr so leichtgängig wie die großen Taster, dafür aber viel kleiner. Zudem kann man mehrere davon übereinander setzen und alle gleichzeitig betätigen. Wir zitieren eine Abbildung aus ft:pedia 4/2012:

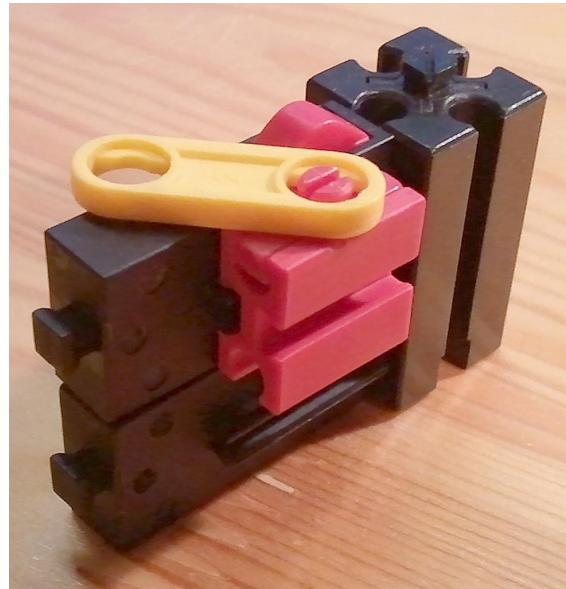


Abb. 9: Zwei Minitaster gemeinsam betätigt

Ein ganz besonders flexibel einsetzbarer Schalter ist der [Stufenschalter](#), der in Zusatzpackungen und im [hobby 3](#) enthalten war (Abb. 10).

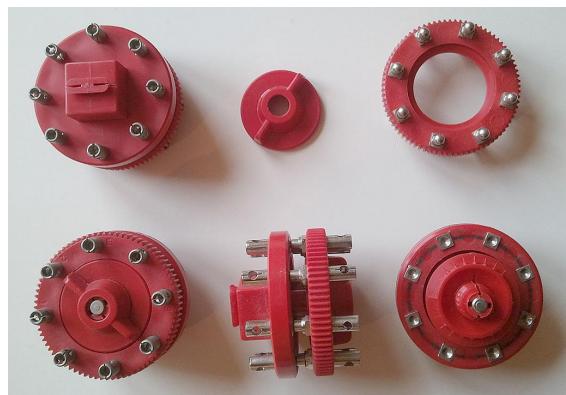


Abb. 10: Stufenschalter

Er trägt in seinem Unterteil acht Kontakte, auf die in seinem drehbaren Oberteil entsprechend acht Federkontakte drücken (die Federkontakte besprechen wir später noch). So konnte dieser Schalter beliebig verdreht werden, um eine von acht Stufen anzusteuern. Durch die Verkabelung der oberen und unteren Kontakte lassen sich eine Vielzahl von Steuerschaltungen aufbauen. Oftmals hat man auf dem Oberteil sogar nur kurze Kabelbrücken ohne Verbindung nach „außen“, sodass der Schalter beliebig oft gedreht werden kann.

Eine mögliche Schaltung kann einen Motor ausschalten, oder für Links- und Rechtsdrehung einschalten (hier werden also nur drei Stellungen benötigt), und gleichzeitig noch mit zwei Lampen die Drehrichtung anzeigen – alles mit einem einzigen Schalter:

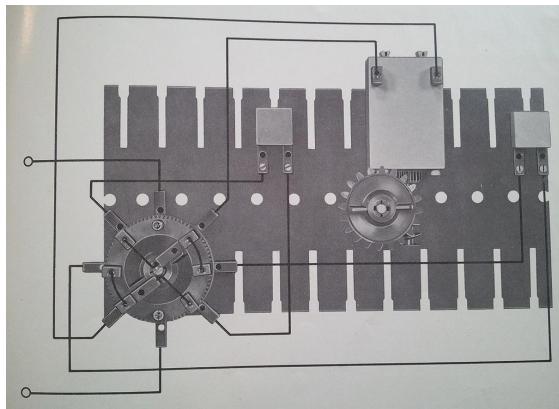


Abb. 11: Motorsteuerung mit dem Stufenschalter

Leuchtsteine

Die ersten Lampenbausteine hatten Glühbirnen für 6,8 V zum Einschrauben. Es gab bereits normale Birnen und Linsenlampen, in deren Glaskolben eine Linse integriert war.

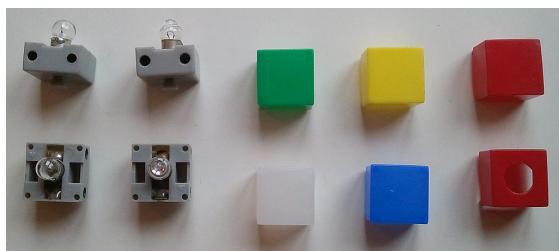


Abb. 12: Leuchtsteine mit schraubbaren Glühbirnen

Damit kann man den Lichtkegel konzentrieren und so z. B. Lichtschranken für größere Distanzen aufbauen. Für die Linsenlampen gab es extra eine Leuchtkappe rot mit Bohrung, damit das Licht auch ausgenutzt werden konnte und die Leuchtkappe nicht überhitzte (die für Fotozellen gedachten schwarzen Leuchtkappen sollte man aus eben diesem Grunde *nicht* für Linsenlampen verwenden). Eine kleine

Ergänzung war das kleine Abdeckplättchen, mit dem man die unbenutzten Kontakte der Leuchtsteine abdecken konnte.

In neuerer Zeit wurden die Glühbirnen gegen solche in einem Einstekteil ersetzt. Diese Lampen sind für die mittlerweile bei fischertechnik üblichen 9 V Betriebsspannung ausgelegt. Die Leuchtkappen bestehen jetzt aus härterem Plastik und sind eher transparent als matt:



Abb. 13: Leuchtsteine mit Stecklampen

Verteilerplatten

Für die Verlängerung elektrischer Leitungen kann man Leuchtsteine ohne Glühlampe verwenden, man kann aber auch die speziellen ein- oder dreipoligen Verteilerplatten benutzen:

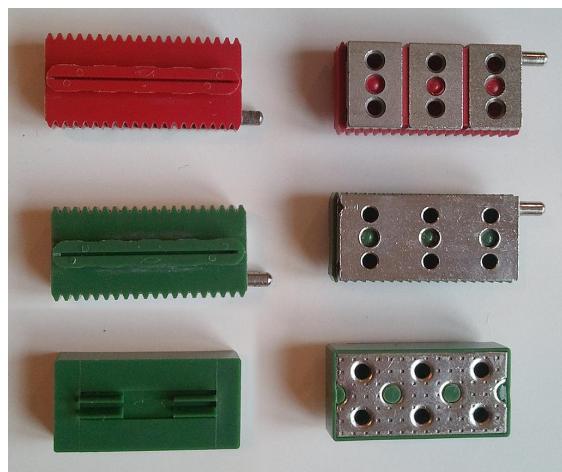


Abb. 14: Verteilerplatten

Sie werden mit den Stegen auf ihrer Rückseite in der Nut eines Bausteins befestigt; die spätere Version hat Clipfedern zum Aufschieben oder einfachen

Aufdrücken. An einer Ecke der Ur-Versionen ragt ein Metallstift heraus, der in die entsprechenden Löcher auf der anderen Seite passt. So kann man einen einfachen Schiebeschalter bauen oder mehrere Verteilerplatten ohne zusätzliche Leitung verbinden.

Magnete und Rückschlussplatten

Schon gleich zu Anfang der fischertechnik-Elektromechanik gab es die Dauermagnete mit roter und grüner Bodenplatte, die sich durch die magnetische Polung der aufgeklebten Magnete unterschieden. Mit Dauermagneten kann man eine Vielzahl an Experimenten und Modellen aufbauen – da lohnt sich ein Blick in die alten Elektromechanik-Anleitungen.

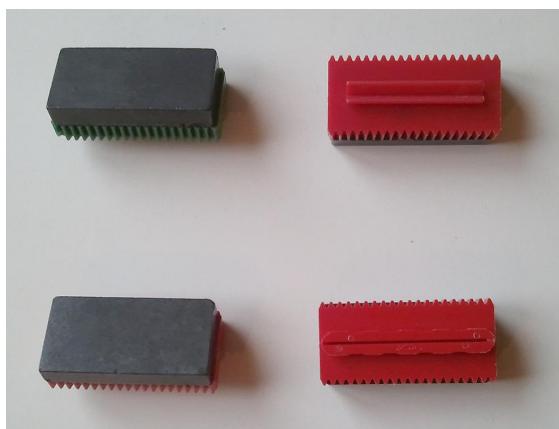


Abb. 15: Dauermagnete

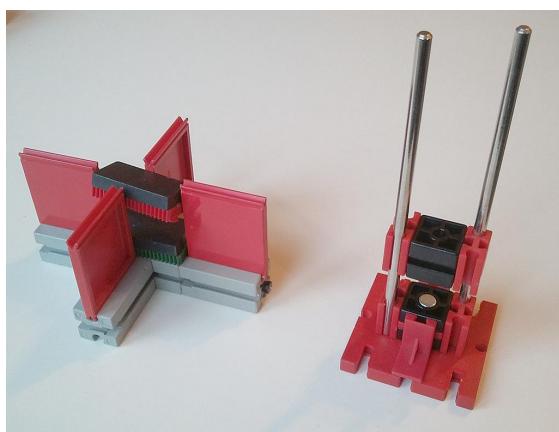


Abb. 16: Schwebemodelle aus je zwei Magnetbausteinen

Mehrere Generationen führten schließlich zum heute aktuellen [108278 Magnetbaustein 15](#). Eine [Suche nach „Magnet“](#) in der ft-Datenbank liefert einen noch größeren Überblick.

Mit den Elektromagneten – erst für 6,8 V mit Bügel, später für 9 V mit Zapfen und Federn an den Seiten – sind auch elektrisch steuerbare Magnetfelder verfügbar.

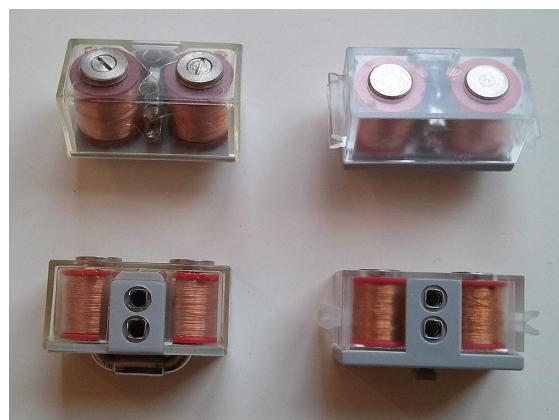


Abb. 17: Älterer und neuerer Elektromagnet

Die lassen sich für unzählige Zwecke nutzen, und damit man neben Metallachsen auch etwas zum magnetisch Anziehen hat, gab es die Rückschlussplatten – rund und eckig.



Abb. 18: Rückschlussplatten

Ein Anwendungsbild davon sehen wir noch im Verlauf dieses Artikels.

Federfüße

Ein wichtiges und oft verwendetes Bauteil war der [31307 Federfuß](#). Er ist ca. 30 mm lang und kann federnd verbogen werden. Das kann für federnde Teile in der Mechanik, Sperren, Rasten, das Andrücken von Kontakten und vieles mehr verwendet werden.

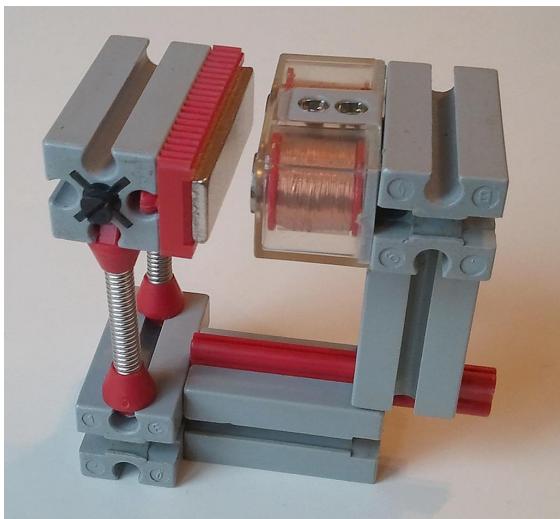


Abb. 19: Ein Elektromagnet zieht eine von Federfüßen gehaltenen Rückschlussplatte an
Anwendungsbilder sehen wir gleich noch.

Steckachsen

In den frühen Elektromechanik-Kästen waren auch lange Metallachsen enthalten. Die können zur Verlängerung sogar aneinander gesteckt werden und sind vor allem als lange Stromführungen gedacht.



Abb. 20: Verlängern von Stromführungen

Kontakte

Eine ganze Reihe verschiedener Kontaktstifte gab es in den verschiedenen Elektromechanik-Kästen und dem Kasten hobby 3.

Die wichtigsten waren der [31338 Klemmkontakt](#) für die damaligen 4,5 V-Flachbatterien und für Achsen, das [31305 Kontaktstück 20](#) und der [31306 Federkontakt](#) (rechts unten in Abb. 21). Der enthält im Inneren eine Feder, die den auf der Abbildung nach links herausragenden Stößel sanft herausdrückt.

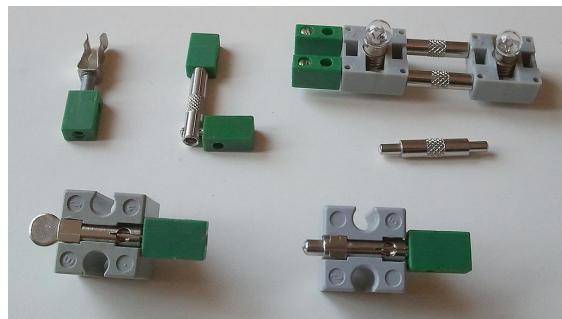


Abb. 21: Kontaktstifte

Ein Beispiel für die Anwendung sind schleifende Kontakte. Abb. 22 zeigt eine verschiebbare Bausteinreihe mit Stromführungsachsen, deren eines Ende direkt in einem Leuchtstein steckt. Auf einer Seite wird der Strom mit einem per Feder-Gelenkbaustein angedrückten Kontaktstück zugeführt, auf der anderen Seite von einem Federkontakt:

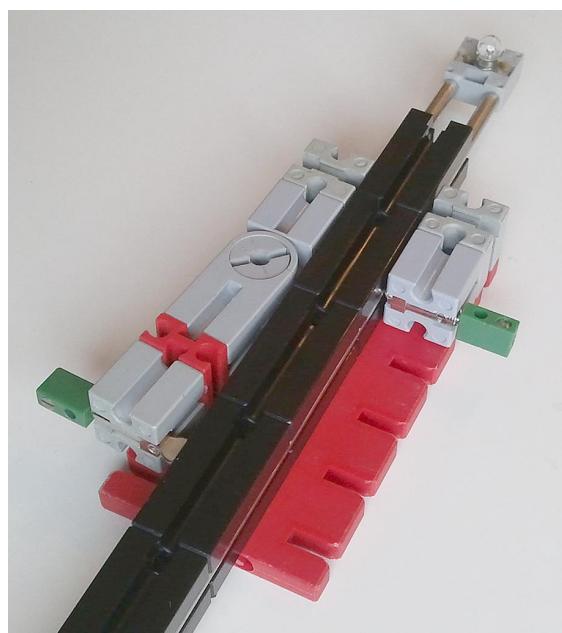


Abb. 22: Einsatzbeispiele von Kontakten

Schleifringe und Schalt-scheiben

Sehr auffällig waren die [Schleifringe](#), die zwei umlaufende Metallbahnen besitzen, die wiederum mit je einem Kontakt verbunden sind. Sie liegen in zwei Ausführungen vor: Mit Buchsen und mit Stiften (Abb. 23).

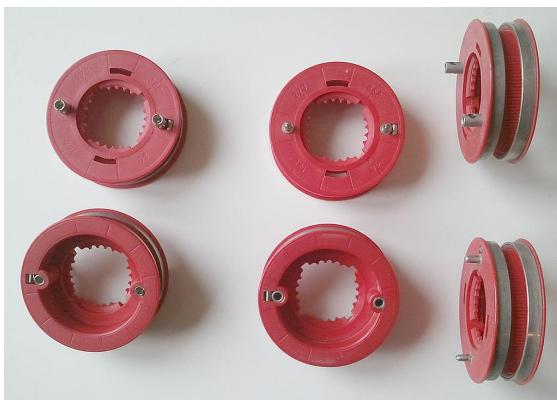


Abb. 23: Schleifringe mit Buchsen und mit Stiften

Die Kontakte passen in eine angesteckte Drehscheibe, von der ab man weiterbauen kann. Das dient zum einen der Stromübertragung auf drehende Teile eines Modells:

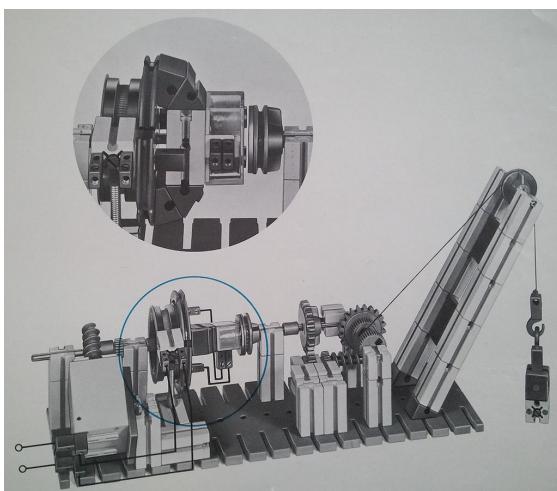


Abb. 24: Kranmodell mit elektromagnetischer Kupplung aus der e-m1-Anleitung

Die Variante mit Stiften kann direkt neben anderen Schleifringen auf einer Achse montiert werden. Dadurch kann man mehrere Schleifringe elektrisch verbinden. Mit einer kurzen Kabelbrücke am letzten Schleifring hat man so sämtliche Metallbahn-Paare miteinander verbunden.

Wozu dienen mehrere Schleifringe, bei denen alle Bahnen miteinander verbunden sind? Das wird klar, wenn wir die zu den Schleifringen gehörenden Einfach- und Doppel-Unterbrecherstücke in Abb. 25 betrachten.

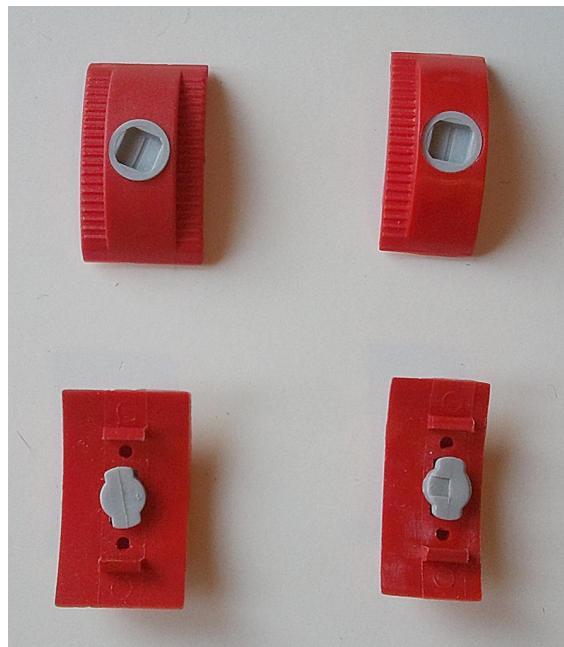


Abb. 25: Unterbrecherstücke

Sie können an beliebiger Position auf Schleifringe aufgesetzt und durch Aufsetzen und Verdrehen des Zapfens eines Bausteins verriegelt werden. Die Schleifringe besitzen dazu sogar eine Skala mit Grad-Einteilung auf der Seite. Und das ermöglicht den Bau programmgesteuerter Maschinen! Mit den schon in der letzten Ausgabe der ft:pedia gezeigten Feder-gelenksteinen oder mit den oben beschriebenen Kontakten und Federfüßen kann man einer Bahn Strom zuführen – auf dieser Bahn bringt man also keine Unterbrecher an. Die anderen Bahnen liefern dann Strom an die angedrückten Kontakte, wenn gerade kein Unterbrecherstück „im Weg“ ist.

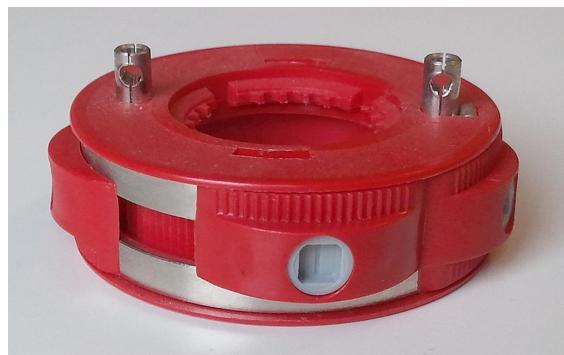


Abb. 26: Mit Unterbrechern bestückter Schleifring

Ein Bild dazu findet sich auch im Artikel [Programmsteuerungen](#) in dieser Ausgabe.

Da die Unterbrecherstücke eine sanfte Erhöhung besitzen, lassen sich auch Taster davon in beiden Drehrichtungen wunderbar betätigen.

Wenngleich man mit den später erschienenen [Schaltscheiben](#) keinen Strom auf drehende Teile übertragen kann, so dienen sie doch wenigstens der Betätigung von Tastern oder ähnlichem. Sie werden immer paarweise auf eine ft-Nabe gesetzt, und durch verdrehtes Anbringen kann man festlegen, wie groß die „Öffnung“ sein soll, in deren Winkelbereich der Taster nicht betätigt werden soll:

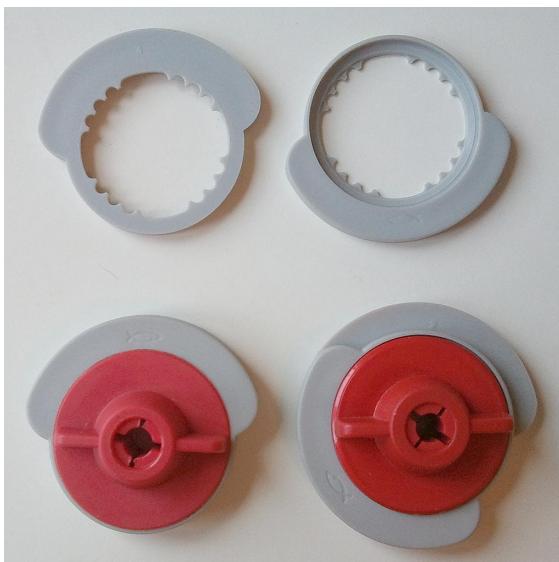


Abb. 27: Schaltscheiben

Schwingfedern

Die [31329 Schwingfeder](#) kann vielseitig eingesetzt werden: Zum Selbstbau eines Relais oder auch einfach als federndes Element, wie die folgenden Bilder zeigen.

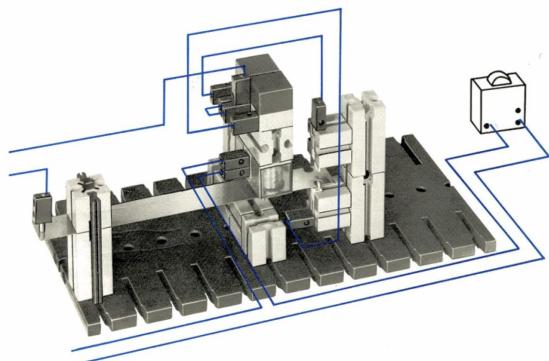


Abb. 28: Selbstbau-Relais mit einem Umschaltkontakt aus Schwingfeder, Elektromagnet und Kontaktstücken aus der Anleitung zum Kasten em1

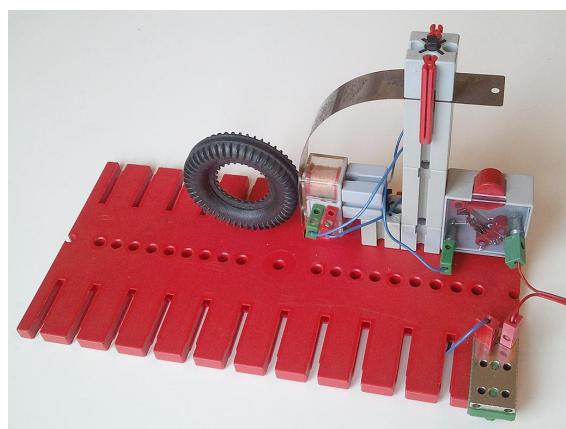


Abb. 29: Bausteinschleuder mit Elektromagnet und Schwingfeder

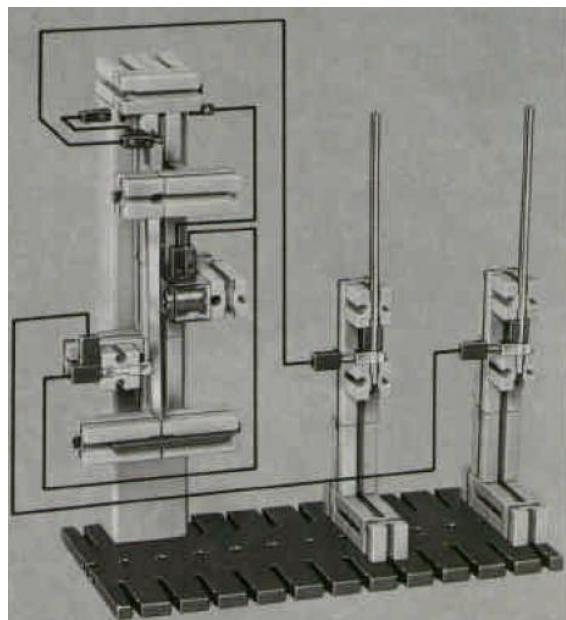


Abb. 30: Elektrisiermaschine

Ein bemerkenswertes Modell mit der Schwingfeder und einem Elektromagneten war die „Elektrisiermaschine“ aus dem [Clubheft 1972-3](#):

Aus der geringen Betriebsspannung entstehen durch Ausnutzung der Eigenarten des Elektromagneten bis zu 1000 V Spannung an den beiden Achsen! Aber keine Sorge: Da die verfügbare Stromstärke dabei winzig wird, bleibt die Sache völlig ungefährlich.

Bimetallstreifen

Nicht zu verwechseln mit der Schwingfeder ist der [Bimetallstreifen](#). Er besteht, wie sein Name schon sagt, aus zwei verschiedenen Metallstreifen, die fest aneinander geklebt sind. Da die beiden Metallarten sich bei Erwärmung unterschiedlich stark ausdehnen, kann man damit – rein elektromechanisch – z. B. eine Sicherung gegen Überhitzung bauen:

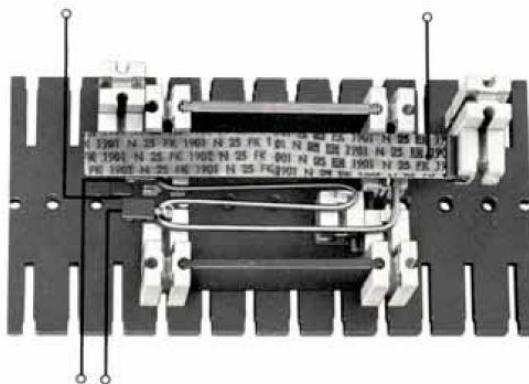


Abb. 31: Bimetall-Thermosicherung mit der ft-Antriebsfeder als Heizelement

Eine Heizung – mehrere ft-Lampen oder eine unter Strom gesetzte ft-Antriebsfeder etwa – erwärmen das Bimetall. Der elektrische Kontakt wird je nach Modellbauweise als Öffner oder Schließer zwischen dem Bimetall (in das ein ft-Stecker eingesteckt werden kann) und einem der Kontaktstücke hergestellt.

Relais

Neben den mit Elektromagneten und anderen Einzelteilen zu bauenden Relais gab es auch schon früh fertige Relais mit zwei Umschaltkontakten von fischer-technik. Die erste Version ist nur 30 • 30 • 45 mm groß, der spätere RB II steckt dagegen in einem „Silberling“-Gehäuse wie die 1970er Elektronikbausteine (die wir in einer späteren Folge dieser Artikelserie noch ausführlich besprechen werden).



Abb. 32: em5-Relais

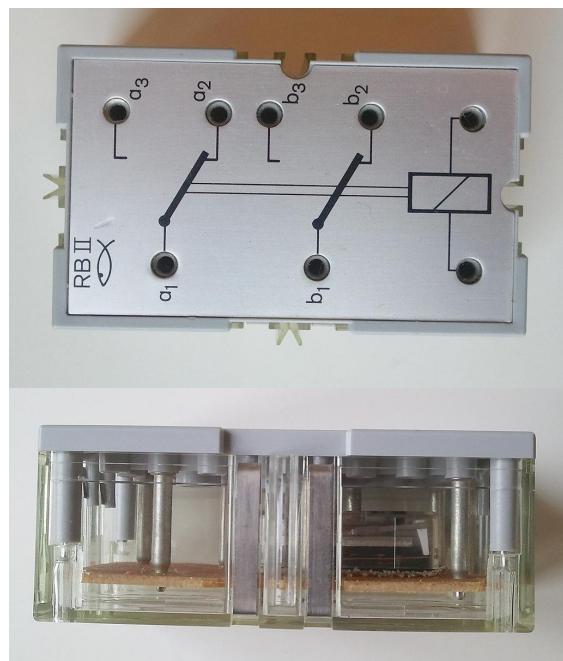


Abb. 33: Relaisbaustein RB II

Elektrisch verhalten sich die beiden Relais-Varianten völlig gleich. Die ähnlich aussehenden Relaisbausteine aus dem Elektronik-Programm haben zusätzlich noch einen Verstärker eingebaut – auch das besprechen wir in einer späteren Folge.

Zählwerk

Mit dem Zusatzkasten [30242 Zählwerk](#) bekommt man ein elektromagnetisch bis 20 Impulse zählendes und anzeigenendes Element.

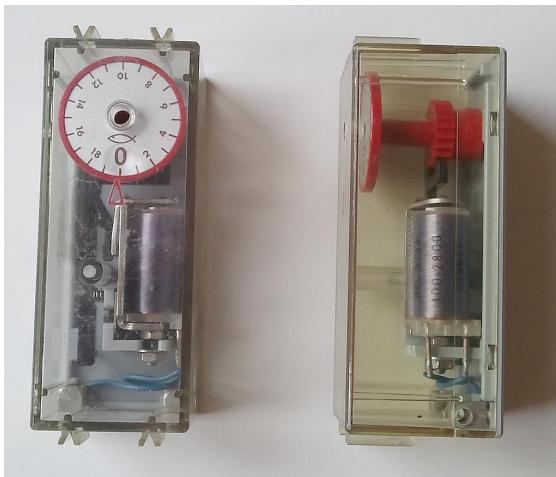


Abb. 34: Das ft-Zählwerk

Natürlich lassen sich Zählwerke auch in unzähligen Bauweisen aus Elektromechanik-Bauteilen selber herstellen – siehe etwa die Artikel *Vom Zählen und Abzählen* aus [ft:pedia 1/2012](#) und [2/2012](#).

Reed-Kontakt

Magnetisch schalten kann man mit [Reed-Kontakten](#). Es gibt sie in verschiedenen Ausführungen (Abb. 35).

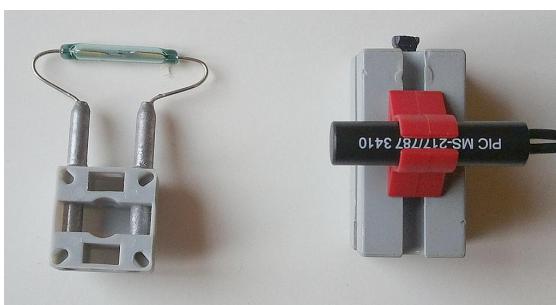


Abb. 35: Ältere und jüngere Reedkontakte



Abb. 36: Detailaufnahme des Reedkontakte

Kommt der Reedkontakt in die Nähe eines Magnetfeldes, schließt er elektrisch wie ein Taster. Damit lassen sich natürlich unzählige Maschinen bauen, bei denen eine Dreh- oder Schiebeposition ohne direkte Berührung festgestellt werden kann.

Summer

Eines der jüngsten elektromechanischen Bauteile ist schließlich der [Summer](#) aus dem Jahre 1991. Er ist zwar leider überhaupt nicht ans fischertechnik-Raster angepasst, aber er lässt sich doch recht einfach verbauen.

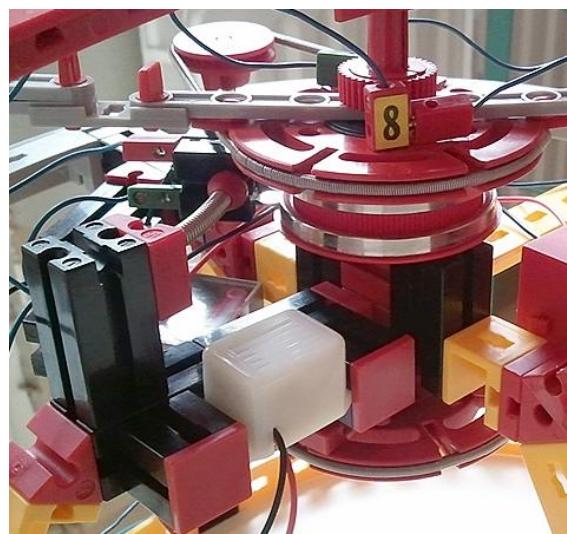


Abb. 37: Ausschnitt aus dem Modell „[Party-Hut](#)“ mit Antriebsfeder, Schleifring, Federfuß, Kontaktstücken und Summer

Wie geht's weiter?

In der nächsten Folge von „Perlentauchen“ gehen wir die fischertechnik-Elektronik an. Mehrere Elektronik-Generationen über mehr als drei Jahrzehnte zeigen nicht nur die Entwicklung der fischertechnik-Elektronik, sondern auch die des technischen Fortschritts auf diesem Gebiet.

Computing

I²C mit dem TX – Teil 3: Luftdruckmessung

Georg Stiegler

Seit dem Firmware-Update vom April 2012 auf Version 1.30 lassen sich am TX beliebige I²C-Komponenten anschließen und mit den Robo Pro Versionen ab 3.1.3 ansteuern. Nach einer Einführung in die Grundlagen des I²C-Protokolls [1] und in die Ansteuerung eines LED-Displays [2] wird in diesem Beitrag der Anschluss und die Nutzung eines I²C-Luftdrucksensors vorgestellt.

Hintergrund

Wer so wie mein Sohn und ich mit Begeisterung die RoboPro-Bibliothek „I²C“ durchforstet läuft Gefahr von den faszinierenden neuen Möglichkeiten infiziert zu werden – und diese vielleicht sogar selbst erweitern zu wollen. Mit ein wenig Mut und Ausdauer lassen sich dabei neue Welten entdecken oder – nicht ganz so hoch gegriffen – zumindest Gesetzmäßigkeiten unseres eigenen Planeten anschaulich erkunden. Der letzte Familienurlaub in den Bergen war Auslöser des Gedankens, den Zusammenhang zwischen Luftdruck und Höhe genauer zu durchleuchten und am besten sogar vor Ort Messungen mit dem Reisebegleiter „TX Controller“ vorzunehmen. Wieder zu Hause machten wir einen Luftdrucksensor mit ausführlicher Dokumentation (leider wie so oft nur in Englisch verfügbar) der Firma Bosch ausfindig:

Der BMP085

Dieser digitale Sensor [3] dient zur äußerst präzisen Messung des Druckes in einem Bereich von 300 hPa bis 1.100 hPa (hPa = Hektopascal = Millibar). Aus dem Luftdruck lässt sich die Höhe ableiten – dazu später mehr – und so gibt der Hersteller einen Messbereich von - 500 m bis

+ 9.000 m über dem Meeresspiegel (NN = Normalnull) an. Das Rauschen der Druckmesswerte soll im besten Fall etwa 0,03 hPa oder 0,25 m betragen. Dank seiner geringen Abmessungen und des geringen Stromverbrauchs kommt der Sensor beispielsweise in GPS-Navigationsgeräten, Höhenmessern, Wetterstationen oder auch PDAs und Handys zum Einsatz.

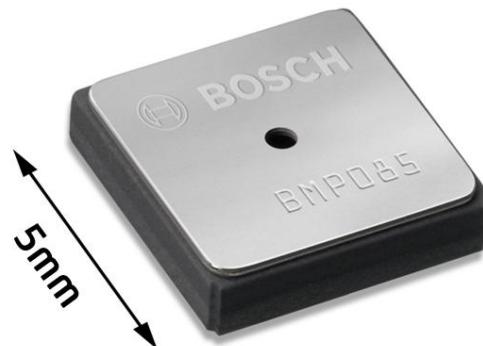


Abb. 1: Foto des Sensors

Der Messung des Druckes geht bei diesem Sensor immer erst eine Messung der Temperatur (Messbereich - 40°C bis + 85°C) voraus – schön für uns, denn dadurch benötigen wir z. B. bei Wetterbeobachtungen keinen weiteren Sensor. Zudem ist der Sensor bereits ab Werk kalibriert und kann somit ohne Umschweife per I²C-Bus mit dem TX Controller betrieben werden.

Aber eine Stolperstelle gibt's dann doch noch: die Versorgungsspannung des BMP085 liegt zwischen +1,8 und +3,6 Volt. Der I²C-Anschluss EXT 2 am TX Controller liefert jedoch +5 Volt. Eine große Anzahl an Sensoren und Aktoren ist direkt an die 5 V anschließbar – es gibt aber auch viele, die nur für 3,3 Volt ausgelegt sind.

Daher empfiehlt es sich, beim Kauf eines so genannten „Breakout Boards“ darauf zu achten, dass dieses direkt an die +5 V Versorgungsspannung des TX-I²C-Busses angeschlossen werden kann (Abb. 3 oben).

Andernfalls muss man mittels weiterer elektronischer Bauteile die Versorgungsspannung sowie die Pegel der Datenleitung (SDA) und der Takteleitung (SCL) konvertieren. Solche Bausteine nennt man „Level Shifter“ oder „Level Converter“. Damit lassen sich dann 5 V- und 3,3 V-Bausteine gleichzeitig an einem I²C-Bus betreiben

(Abb. 3 unten). Bei dem günstigen [Sensor-Board der Firma Drotek](#) (Abb. 2) kann man sogar zwischen beiden Spannungstypen wählen.

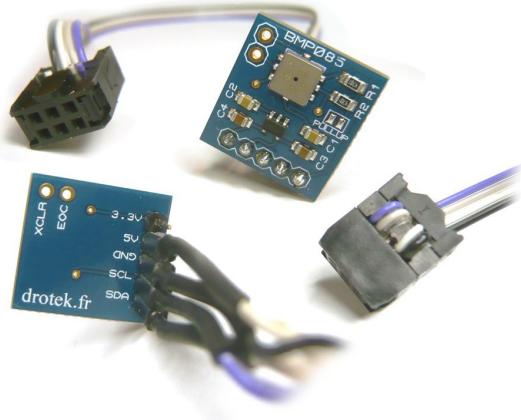


Abb. 2: Foto eines Breakout Boards

Für den im Folgenden beschriebenen RoboPro-Treiber spielt es keine Rolle welches Board zum Einsatz kommt.

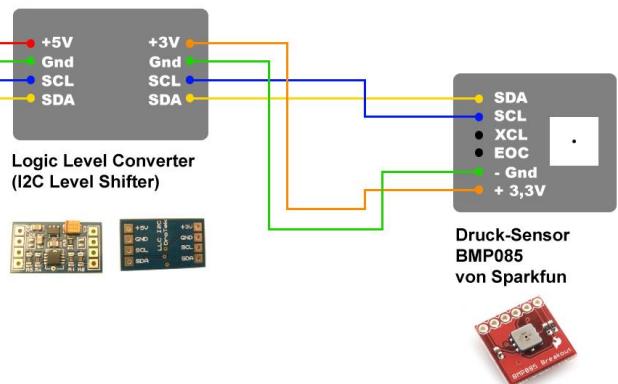
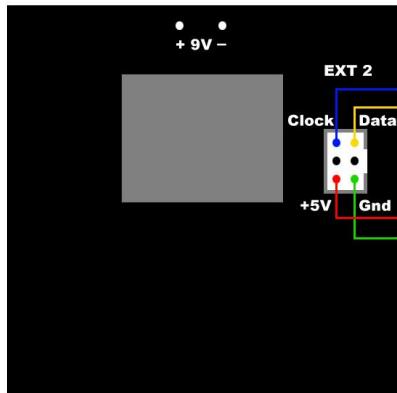
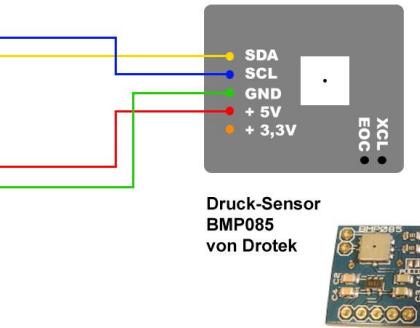
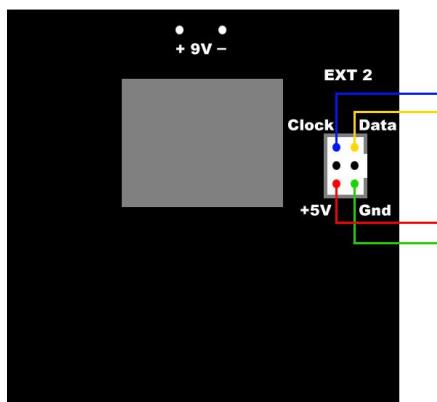


Abb. 3: Anschluss an den TX-Controller

Funktionsweise

Der BMP085 umfasst einen piezoresistiven Sensor – eine Membran biegt sich infolge eines mechanischen Druckunterschieds zwischen der einen Seite und der gegenüber liegenden Seite durch, wodurch aufgebrachte elektrische Widerstände verändert werden [4] – einen Analog-Digital-Wandler, eine Kontrolleinheit mit einem EEPROM [5] und die serielle I²C-Schnittstelle.

Jeder einzelne BMP085-Sensor enthält ab Werk in einem speziellen Speicherbereich (EEPROM) 11 Kalibrierungsdaten, die vor der eigentlichen Messung einmalig ausgeleren werden müssen (Abb. 4, 1.). Diese Daten brauchen wir, um später die gemessenen, unkompenzierten Werte für Temperatur und Druck individuell berechnen zu können.

Die Messung der Temperatur und des Drucks wird jeweils mit einem Startbefehl an den Sensor eingeleitet, worauf nach einer kurzen Wartezeit die Messwerte ausgeleren werden können. Anschließend wird die Temperatur in Grad Celsius und der absolute (also tatsächliche) Luftdruck in hPa berechnet.

I²C-Kommandos

Um unserem Sensor die Messwerte für die Temperatur und den Druck zu entlocken müssen wir sowohl beim Befehl I²C-Lesen als auch später beim I²C-Schreiben die Geräteadresse kennen – sie lautet **0x77**.

Da der BMP085 eine I²C-Datenübertragungsgeschwindigkeit von bis zu 3,4 MHz zulässt, können wir unter Geschwindigkeit den „Fast Mode“ mit 400 kHz auswählen.

Im ersten Schritt wollen wir die 11 Kalibrierungsdaten aus dem internen Speicherbereich mit der Unteradresse 0xAA lesen. Aus dem Datenblatt geht hervor, dass die Länge der Daten jeweils 16 bit beträgt (MSB zuerst).

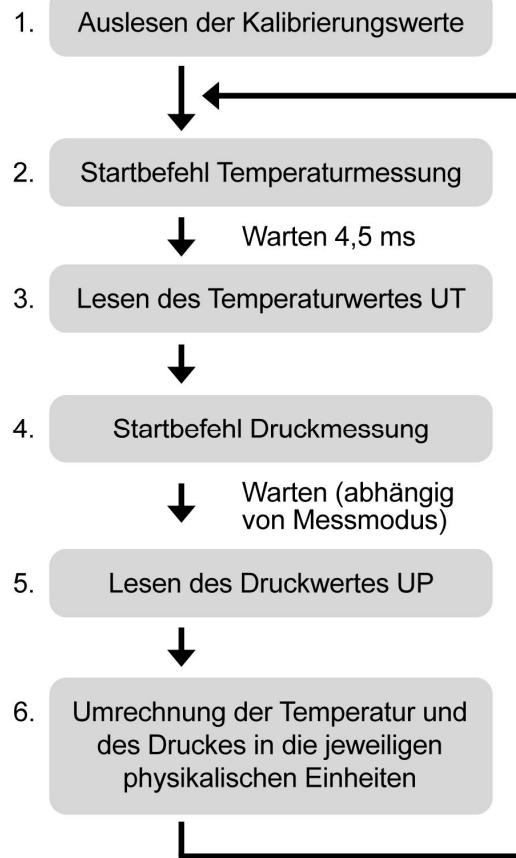


Abb. 4: Funktionsprinzip

Da die Daten alle aufeinander folgen (176 bit lang) bleiben die Lese-Befehle „offen“ bis der letzte Wert gelesen ist (Abb. 5).

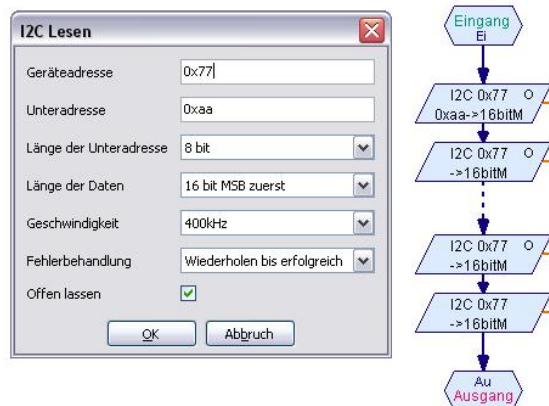


Abb. 5: Lesen der Kalibrierungsdaten

Ein kleines Problem (für mich im ersten Moment ein großes) beim Auslesen ergibt sich dadurch, dass mehrere Daten als „unsigned short“ interpretiert werden

müssen. Da ich bislang leider keiner Programmiersprache mächtig bin (im Bereich der Mikrocontroller ist „C“ und „C++“ sehr weit verbreitet), musste ich wohl oder übel erst mal ein wenig „Grundlagenforschung“ betreiben.

Es gibt mehrere Datentypen [6], die man in der Weiterverarbeitung – beispielsweise mit Bit-Operatoren wie SHR oder SHL – kennen und beachten muss (Tab. 1).

Typ	Speicherplatz	Wertebereich (dezimal)
<code>char</code>	1 Byte	-128 bis +127 bzw. 0 bis 255
<code>signed char</code>	1 Byte	-128 bis +127
<code>unsigned char</code>	1 Byte	0 bis 255
<code>short</code>	2 Byte	-32768 bis +32767
<code>unsigned short</code>	2 Byte	0 bis 65535
<code>int</code>	4 Byte	-2147483648 bis +2147483647
<code>unsigned int</code>	4 Byte	0 bis 4294967295
<code>long</code>	4 Byte	-2147483648 bis +2147483647
<code>unsigned long</code>	4 Byte	0 bis 4294967295

Tab. 1: Datentypen

So hat bei meinem Sensor der vierte Kalibrierungswert „AC4“ einen Wert von -31.439 (short). Dieser muss dann aber als vorzeichenlose Zahl (unsigned short) interpretiert werden [7, 8]: bei einer 16 bit (= 2 Byte) langen negativen Zahl muss man $2^{16} = 65.536$ dazu addieren (Abb. 6 oben).

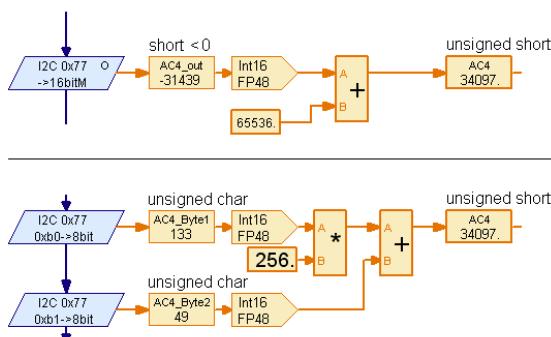


Abb. 6: Lesen einer vorzeichenlosen Zahl

Dabei ist zu beachten, dass bei den Variablen und Operatoren als Datentyp „Gleitkomma 48 bit“ ausgewählt wird.

Man kann aber die Konstanten des Formats „Unsigned Short“ auch byteweise auslesen, was den Vorteil bietet unabhängig davon

zu sein, ob die auszulesende Zahl negativ oder positiv ist (Abb. 6 unten).

Hat man die Kalibrierungsdaten einmalig ausgelesen, wird als nächster Schritt die Temperaturmessung eingeleitet: an die Unteradresse 0xF4 wird der Datenwert 0x2E gesendet, gefolgt von einer Wartezeit (4,5 ms oder auferundet 0,005 s). An der Unteradresse 0xF6 wird dann der 16 bit lange „Roh-Wert (UT)“ für die Temperatur ausgelesen (unsigned short) (Abb. 7).

Der „Pseudo-Code“ im Datenblatt lautet:

```
write 0x2E into reg 0xF4; wait 4,5ms
read reg 0xF6 (MSB), 0xF7 (LSB)
UT = MSB << 8 + LSB
```

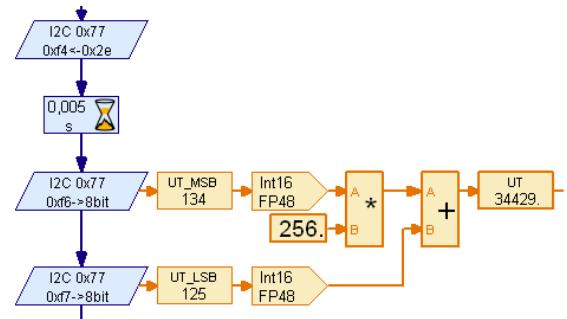


Abb. 7: Auslesen von UT

Abb. 7 zeigt zudem, dass das Zeichen „<<“ im Pseudo-Code einer Multiplikation mit einer entsprechenden 2er-Potenz gleichzusetzen ist (Achtung: Datentypen beachten):

„<< 8“ entspricht „* 2⁸“ entspricht „* 256“

Das Zeichen „>>“ im Pseudo-Code ist einer Division mit einer entsprechenden 2er-Potenz gleichzusetzen (Achtung: Datentypen beachten):

„>> 4“ entspricht „/ 2⁴“ entspricht „/ 16“

Das Auslesen des „unkompensierten“ Druckwertes und die folgenden Berechnungen gestalten sich ein wenig komplizierter, da man hier vier unterschiedliche Genauigkeitsstufen einstellen kann (Oversampling setting, „OSS“; siehe Tab. 2).

Mode	Parameter oversampling_setting	Internal number of samples	RMS noise typ. [hPa]
ultra low power	0	1	0.06
standard	1	2	0.05
high resolution	2	4	0.04
ultra high resolution	3	8	0.03

Tab 2: Oversampling setting (OSS)

Der „Pseudo-Code“ für das Auslesen des unkompensierten Drucks UP im Datenblatt lautet:

```
write 0x34 + (oss << 6) into reg 0xF4
; wait

read reg 0xF6 (MSB), 0xF7 (LSB),
0xF8 (XLSB)

UP = (MSB << 16 + LSB << 8 + XLSB) >>
(8-oss) ; XLSB ist optional für OSS=3
```

Die Unteradresse für den Startbefehl ist hier abhängig vom Oversampling-Modus. So heißt $0x34 + (oss << 6)$ für OSS = 2 „übersetzt“:

$$0x34 + 2 \cdot 2^6 = 52 + 128 = 180 = 0xB4$$

Zur Umrechnung von Hexadezimal-Werten in Dezimalzahlen und umgekehrt gibt es eine feine Website von A. Brünner [9].

Measurement	Control register value (register address 0xF4)	Max. conversion time [ms]
Temperature	0x2E	4.5
Pressure (osrs = 0)	0x34	4.5
Pressure (osrs = 1)	0x74	7.5
Pressure (osrs = 2)	0xB4	13.5
Pressure (osrs = 3)	0xF4	25.5

Tab 3: Datenwerte für die Auslesebefehle sowie die zugehörigen Wartezeiten

Auch die Wartezeit nach dem Startbefehl muss entsprechend OSS eingestellt werden (Tab. 3). Daraus resultiert dann beispielsweise für OSS = 2 (Abb. 8):

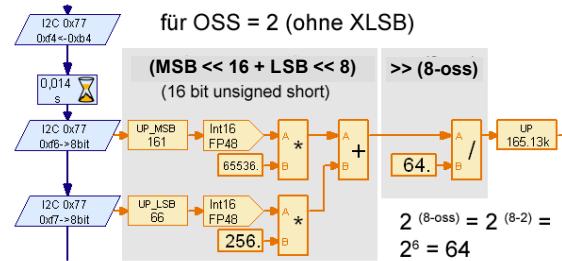


Abb. 8: Auslesen von UP (hohe Auflösung)

Für die höchste Auflösung bei OSS = 3 siehe Abb. 9.

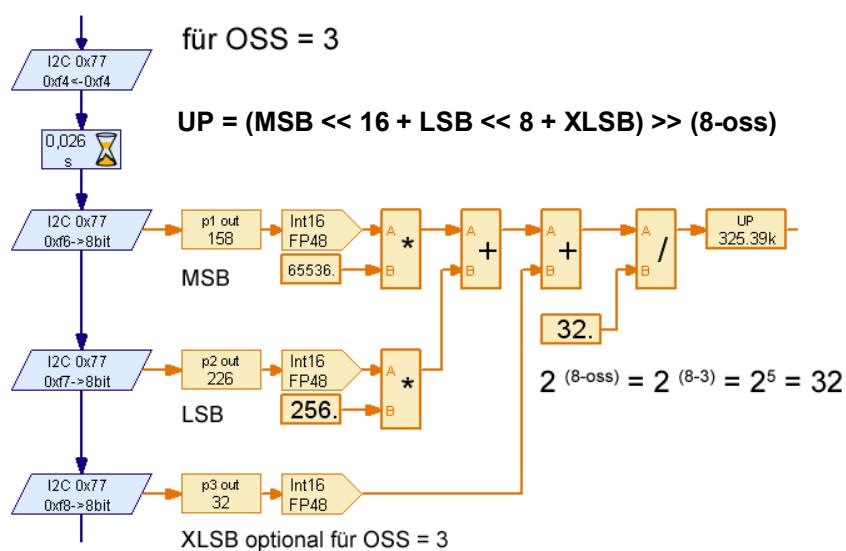


Abb. 9: Auslesen von UP (ultrahohe Auflösung)

Auswertung der Messwerte

Die beiden ausgelesenen Werte für Temperatur UT und Druck UP müssen nun noch in die physikalischen Werte umgerechnet werden (Abb. 10). Dieser Programmteil ist vor allem Fleißarbeit, kann aber dank eines Rechenbeispiels im Datenblatt sehr gut nachvollzogen werden.

```

calculate true temperature
X1 = (UT - AC6) * AC5 / 215
X2 = MC * 211 / (X1 + MD)
B5 = X1 + X2
T = (B5 + 8) / 24

calculate true pressure
B6 = B5 - 4000
X1 = (B2 * (B6 * B6 / 212)) / 211
X2 = AC2 * B6 / 211
X3 = X1 + X2
B3 = ((AC1*4+X3)<< oss + 2) / 4
X1 = AC3 * B6 / 213
X2 = (B1 * (B6 * B6 / 212)) / 216
X3 = ((X1 + X2) + 2) / 22
B4 = AC4 * (unsigned long)(X3 + 32768) / 215
B7 = ((unsigned long)UP - B3) * (50000 >> oss)
if (B7 < 0x80000000) { p = (B7 * 2) / B4 }
else { p = (B7 / B4) * 2 }
X1 = (p / 28) * (p / 28)
X1 = (X1 * 3038) / 216
X2 = (-7357 * p) / 216
p = p + (X1 + X2 + 3791) / 24

```

Abb. 10: Berechnung von T und P

Da viele Multiplikationen und Divisionen durchführt werden, kommt es leider an einigen Stellen zu Rundungsfehlern. Diese werden so gut wie möglich ausgeglichen, indem die Nachkommastellen bei einigen Zwischenergebnissen (Werte zwischen -32.768 und +32.767) eliminiert werden.



Ohne diese Korrekturen erhält man einen um ca. 0,1 hPa höheren Luftdruckwert.

Die gemessene Temperatur lässt sich nun leicht mit Thermometern überprüfen. Sie sollte laut Hersteller +/- 1 °C genau sein.

Der gemessene absolute Luftdruck soll eine Genauigkeit von +/- 0,5 hPa aufweisen, wobei ein Rauschen des Wertes zwischen 0,06 und 0,03 hPa im Datenblatt angegeben wird (abhängig von der Oversampling-Einstellung). Mit meinen beiden Sensoren schaffe ich es bisher nicht, das Rauschen unter +/- 0,1 hPa zu bekommen – vielleicht wird es da noch Nachbesserungen geben.

Um den gemessenen absoluten Luftdruck zu überprüfen, kann man diesen mit [meteorologischen Daten im Internet](#) vergleichen. Weiß man, auf welcher Höhe man sich gegenüber dem Meeresspiegel befindet, kann mittels barometrischer Höhenformel [3, 10, 11] der üblicherweise auf Wetterkarten ausgewiesene Luftdruck über NN berechnet werden. Dabei gilt vereinfacht, dass ein Höhenunterschied von 10 m einem Druckunterschied von 1,2 hPa entspricht. Umgekehrt kann man mittels Messung des Luftdruckes die Änderung der Höhe berechnen: eine Druckänderung um 1 hPa entspricht einem Höhenunterschied von 8,43 m.

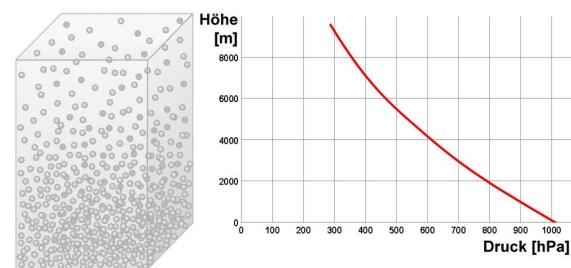


Abb. 10: Luftsäule - die Dichte der Moleküle und somit der Druck nimmt mit der Höhe ab

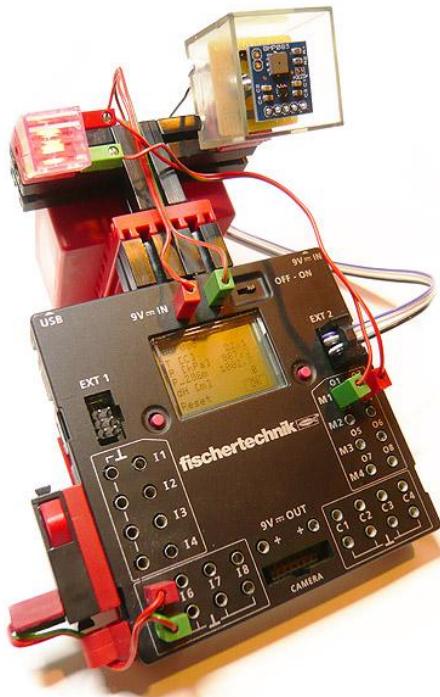
Nun bin ich selbst schon ganz gespannt auf den Einsatz des Sensors in den Bergen – leider dauert es noch bis zum Sommer. Bis dahin suchen wir uns das nächstgelegene Hochhaus mit Aufzug und werden dort erste Messungen vornehmen. Mit Stoppuhrfunktion können wir dann sogar die Geschwindigkeit des Fahrstuhls ermitteln ...

Wer mag, kann vier verschiedene [Robo Pro-Programme](#) downloaden (Achtung: im Dauerbetrieb schaltet der TX gelegentlich das Display ab, läuft aber weiter – fischertechnik arbeitet an dem Problem...):

- BMP085-detailed_v1-2.rpp:
Hier sind die einzelnen Programmsschritte nach dem Rechenweg des Datenblatts kommentiert. Auslesen und Berechnung sind jeweils eigene Unterprogramme (insbesondere OSS=3 wurde berücksichtigt).
- BMP085-simple_v1-2.rpp:
Hier wurden alle unnötigen Variablen entfernt und das Lesen und Schreiben jeweils für Temperatur und Druck in einem Unterprogramm zusammengefasst.
- BMP085-altitude_v1-2.rpp:
Hiermit lässt sich eine Höhendifferenz messen (Genauigkeit +/- 1 m).
- BMP085-simulation_data-sheet.rpp:
Eine Entwicklungsdatei – hier sind die Beispielberechnungswerte aus dem Datenblatt eingesetzt.

Quellen

- [1] Dirk Fox: *I²C mit dem TX – Teil 1: Grundlagen*. [ft:pedia 3/2012](#), S. 32-37.
- [2] Dirk Fox: *I²C mit dem TX – Teil 2: LED-Display*. [ft:pedia 4/2012](#), S. 32-37.
- [3] Bosch Sensortec: *Digital Pressure Sensor BMP085*. Data sheet, v1.3, 22.08.2011.
- [4] Wikipedia: [Silizium-Drucksensor](#).
- [5] Wikipedia: [EEPROM](#).
- [6] Wikipedia: [C++-Programmierung/Einführung in C++/Variablen, Konstanten und ihre Datentypen](#).
- [7] Wikipedia: [Zweierkomplement](#).
- [8] Wikipedia: [Bitweiser Operator](#).
- [9] Mathematik-Seiten von Arndt Brünner: [Umrechnung von Zahlsystemen](#)
- [10] Wikipedia: [Luftdruck](#).
- [11] meteoblue.com: [Luftdruck](#).



Messen

„Einmessen“ eines digitalen Messgeräts

Dirk Fox

Will man Sensoren für die Messung physikalischer Größen verwenden, müssen Spannung, Stromstärke, Widerstandswert oder Zeit ins Verhältnis zu der zu messenden Größe gesetzt werden. Einige Sensoren leisten das in Hardware – so bestimmt z. B. der Ultraschall-Sensor von fischertechnik aus den Signallaufzeiten den Abstand in cm. Die meisten einfachen Sensoren (wie z. B. der ft-Temperatursensor) liefern jedoch nur einen Widerstandswert zwischen 0 und 5 kOhm zurück. Der Beitrag stellt vor, wie man eine geeignete Ausgleichsfunktion ermittelt, die die Messwerte des Sensors in eine physikalische Größe umrechnet. Dabei ist ein wenig angewandte Mathematik unvermeidlich – lasst euch von den Formeln nicht abschrecken, darin kommen fast ausschließlich die Grundrechenarten vor...

Messgeräte

Für die Messung physikalischer Größen wie Entfernung, Gewicht oder Temperatur standen über Jahrhunderte lediglich analoge Messgeräte zur Verfügung. Verwendet wurden dafür Konstruktionen, die sich proportional zur zu messenden Größe verändern, wie beispielsweise eine Waage:

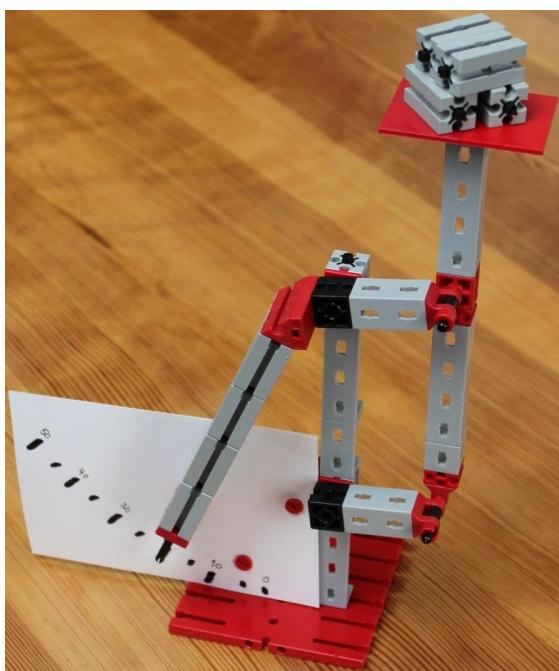


Abb. 1: Briefwaage (frei nach [1], S. 30)

An geeigneter Stelle wurden eine Ablesemarke (meist ein mechanisch bewegter Zeiger) und eine Skale angebracht, deren Einteilung mithilfe von Norm-Gewichten vorgenommen wurde.¹

Analoge Messgeräte (nicht nur aus fischertechnik) haben jedoch meist eine sehr begrenzte Genauigkeit und sind anfällig für mechanische Dejustierungen (*Drift*). Daher müssen sie von Zeit zu Zeit kalibriert werden – dabei wird mit Norm-Größen die Abweichung der Skalenbeschriftung von den tatsächlichen Messgrößen bestimmt.

Seit den ec-Kästen, die Anfang der 70er Jahre des vergangenen Jahrhunderts auf den Markt kamen, bietet fischertechnik Sensoren an, die auf Änderungen physikalischer Zustände wie Helligkeit, Nässe oder Temperatur mit veränderter Leitfähigkeit (Widerstand) reagieren, wie den Fotowiderstand T9011 (31361), den Feuchte-Sensor (36718) oder den Heißleiter 2 kΩ (36386), siehe Abb. 2.

¹ Die grauen B 30 mit Loch wiegen ziemlich genau 5g – perfekt für die Justierung der Briefwaage in Abb. 1.



Abb. 2: Feuchte-Sensor (1974), Fotowiderstand (1969) und Heißleiter 2 kΩ (1971)

Mit diesen Sensoren und den „Silberlingen“ ließen sich einfache Steuerungs- und Regelungsschaltungen realisieren [2]. Messen konnte man damit allerdings nur sehr unzulänglich.

Mittlerweile sind weitere Sensoren wie der Fotowiderstand LDR03 (32698), der NTC-Widerstand² (36437) und ein optischer Farbsensor (128599) verfügbar (Abb. 3).

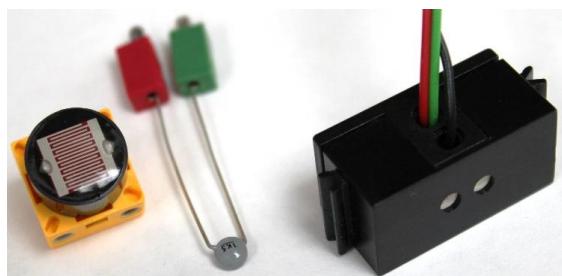


Abb. 3: Fotowiderstand LDR03 (2007), NTC 1,5 kΩ (1987) und Farbsensor (2007)

Diese z. T. über 40 Jahre alten Sensoren lassen sich auch an das Robo Interface und den TX Controller anschließen. Damit lassen sich digitale Messgeräte konstruieren (siehe z. B. [3, 4] und den Beitrag von Georg Stiegler in dieser ft:pedia), die keinem mechanischen Drift unterliegen. Durch die Kombination mehrerer Sensoren kann sogar die Abhängigkeit der Widerstandswerte eines Sensors von Umgebungsbedingungen wie beispielsweise der Temperatur rechnerisch kompensiert werden. Vor allem aber können die Messwerte direkt in die jeweilige physikalische

Größe umgerechnet, grafisch angezeigt und in Tabellen gespeichert werden. Wie aber lässt sich diese Umrechnung bewerkstelligen?

Bei einigen Sensoren kann dem Datenblatt eine Umrechnungsformel entnommen werden. Manchmal findet man dort jedoch nur eine Kennlinie, aber keine Funktionsgleichung für die Umrechnung. Und gelegentlich hat man nicht einmal das, sondern lediglich ein paar Werte aus eigenen (Vergleichs-)Messungen.

Wie aber kann man aus Messwerten eine Umrechnungsformel für den Sensor gewinnen?

Um die Vorgehensweise mit echten Messwerten illustrieren zu können, habe ich mit dem NTC und einem Haushaltsthermometer bei unterschiedlichen Temperaturen ca. 100 Messungen durchgeführt. Damit ich dabei die durch die einfache Messanordnung und das Gehäuse des Thermometers ohnehin begrenzte Genauigkeit der Ergebnisse nicht weiter beeinträchtige, habe ich alle Messungen in geschlossenen Räumen (frei von Luftzug, auf den der NTC sehr empfindlich reagiert) vorgenommen. Bei offensichtlichen „Ausreißern“ habe ich nachgemessen. Tab. 1 zeigt eine Auswahl der Messergebnisse:

R	T	R	T	R	T
1341	25,7	1727	21,8	2478	15,3
1509	24,1	1796	19,3	2685	13,7
1544	24,1	1843	18,2	3177	10,1
1641	21,8	2094	17,6	3285	9,2
1704	21	2249	17	3711	7,8

Tab. 1: Gemessene Temperaturwerte T und zugehörige NTC-Widerstandswerte R

Für eine zuverlässige „Einmessung“ eines Sensors sind das zwar noch viel zu wenige – besser wäre die zehnfache Menge an Messwerten. Aber als „Illustration“ des im Folgenden erläuterten Rechenwegs sollen sie jedoch genügen.

² Negative Temperature Coefficient Thermistor, d. h. der elektrische Widerstand des Heißleiters sinkt mit steigender Temperatur.

Für die Umrechnung dieser Widerstandswerte des NTC in eine Temperatur T liefert fischertechnik das Robo Pro-Unterprogramm „NTC->T“ [3], das aus dem Widerstandswert R des 1,5 k Ω -Heißleiters (0 bis 5 k Ω) die Temperatur °C nach der folgenden Formel bestimmt:

$$T = 1,3932 \cdot (\ln R)^2 - 43,942 \cdot \ln R + 271,87$$

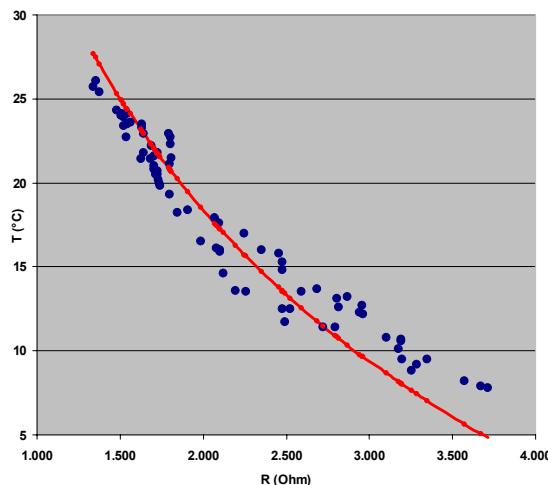


Abb. 4: Widerstandswerte und Temperaturen aus Tab. 1 (blau) und Berechnungen der Robo Pro-Funktion „NTC->T“ (rot)

Aber liefert diese Funktion wirklich eine geeignete Näherung an den tatsächlichen Temperaturverlauf? Einige der berechneten Temperaturwerte weichen, wie Abb. 4 zeigt, deutlich von meinen Messwerten ab. Sofern mir bei meinen Messungen kein systematischer Fehler unterlaufen ist, liefert die Robo Pro-Funktion vor allem bei niedrigen Temperaturwerten keine gute Näherung.

Ausgleichsfunktion

Wir suchen also eine stetige Funktion, die den gemessenen Sensorwerten jeweils eindeutig eine physikalische Größe zuordnet. Diese Zuordnung hätten wir gerne als Funktionsgleichung f ausgedrückt, um zu jedem gemessenen Sensorwert x_i die entsprechende Größe direkt in der passenden Maßeinheit ausrechnen zu können.

Im besten Fall stimmen alle Funktionswerte $f(x_i)$ mit unseren Messwerten y_i überein (*Interpolation*). In der Praxis ist das aber nicht das Ziel, denn die „Punktwolke“ der Messwerte ist das Ergebnis unvermeidlicher Messungenauigkeiten (z. B. wegen Sensortoleranzen), die durch eine geeignete *Ausgleichsfunktion* „geglättet“ werden sollen. Ziel ist daher, eine möglichst gute Näherung an den Verlauf der „Punktwolke“ zu finden.

Für die Suche nach einer Ausgleichsfunktion benötigen wir zweierlei: Eine Methode, um optimierte Kandidaten für eine solche Funktion zu finden, und ein Maß, um die Güte dieser Funktion zu vergleichen.

Methode

Grob lässt sich die Vorgehensweise in den folgenden Schritten beschreiben:

- Mit Hilfe eines möglichst fehlerfreien Messverfahrens und eines hinreichend genauen Messgeräts für unsere gesuchte physikalische Größe bestimmen wir n ($n > 2$) unterschiedliche Messwerte y_i in einem für die spätere Nutzung interessanten Messintervall sowie die zugehörigen Sensorwerte x_i (siehe Tab. 1).
- Anschließend wählen wir eine Funktionsklasse $f(x)$, deren Kurve dem Verlauf der Messkennlinie ähnelt.³
- Dann bestimmen wir die Parameter der Funktion so, dass die damit berechneten Werte $f(x_i)$ die Messwerte y_i möglichst gut annähern bzw. prognostizieren.
- Schließlich bestimmen wir ein „Näherungsmaß“, das die Güte unserer Ausgleichsfunktion angibt.

Dabei kommt der Güte unserer Messungen (Abdeckung des Messbereichs, keine

³ Für manche Sensoren – auch für den NTC, siehe weiter unten – ist eine Funktionsklasse bekannt, die eine gute Ausgleichsfunktion liefert.

systematischen Messfehler, Gleichverteilung der Messungenauigkeiten) eine große Bedeutung zu, da die Parameter der Ausgleichsfunktion auf die Messwerte hin optimiert werden. Die Ergebnisse der Ausgleichsrechnung sollten mit wiederholten Messwerten reproduzierbar sein.

„Naives“ Beispiel

Stünden Sensorwerte und zugehörige physikalische Größen in einem linearen Verhältnis, dann wäre die Bestimmung einer Ausgleichsfunktion einfach. Die beste Annäherung an die Messkennlinie unseres Sensors wäre eine Gerade

$$f(x) = a \cdot x + b$$

Für die Bestimmung der Parameter a (Steigung) und b (Konstante) genügen zwei Messwertpaare (x_1, y_1) und (x_2, y_2) :

$$y_1 - y_2 = f(x_1) - f(x_2) = a(x_1 - x_2)$$

$$\text{also: } a = \frac{(y_1 - y_2)}{(x_1 - x_2)} \quad (1)$$

$$\text{und: } b = y_1 - a \cdot x_1 \quad (2)$$

Je weiter dabei die Werte x_1 und x_2 voneinander liegen, desto genauer trifft die Funktion f den tatsächlichen Kennlinienverlauf. Bei einem Heißleiter erhält man einen Messwert sogar „frei Haus“: Der *Nennwert* des NTC gibt den Widerstandswert an, den er bei 25°C einnimmt [5]. Beim NTC 1,5 kΩ (Abb. 3) kann daher $(x_1, y_1) = (1.500, 25)$ gewählt werden.⁴

Leider ist die Messkennlinie eines Sensors meist nicht linear. Auch für unseren NTC-Widerstand stellt eine lineare Ausgleichsfunktion keine allzu gute Näherung der Messkennlinie dar, wie die Grafik in Abb. 6 und das Gütemaß des folgenden Abschnitts zeigen.

⁴ In den Begleitheften der Computing-Kästen ist irrtümlich von 20°C die Rede (siehe z. B. Robo TX Training Lab Begleitheft, 11/2009, S. 6).

Kleinste Quadrate

Die Grundidee zur Bestimmung einer Ausgleichsfunktion geht auf den Mathematiker [Carl Friedrich Gauß](#) (1777-1855) zurück, der damit im Jahr 1801 erstmals aus Planetenbeobachtungen deren nicht kreisförmige Umlaufbahnen berechnete [6]. Entwickelt hatte er seine „Methode der kleinsten Quadrate“ [7] bereits sechs Jahre zuvor – im zarten Alter von 18 Jahren.



Abb. 5: Carl Friedrich Gauß (1777-1855) auf einem alten Zehn-Mark-Schein

Die Idee selbst ist bestechend einfach: Man wähle zunächst eine Funktionsklasse für die Ausgleichsfunktion $f(x)$, z. B. eine

- Linearfunktion: $f(x) = a \cdot x + b$
- Polynomfunktion: $f(x) = \sum_{k=0}^r a_k \cdot x^k$
- Exponentialfunktion: $f(x) = b \cdot e^{ax}$
- Potenzfunktion: $f(x) = b \cdot x^a$

Dann gilt: Die Parameter a , b (bzw. a_k) von $f(x)$ sind *optimal*, wenn die Summe q der Quadrate der Abweichungen der Funktionswerte $f(x_i)$ von den Messwerten y_i (auch *Residuen* genannt) *minimal* ist:

$$q = \sum_{i=1}^n (f(x_i) - y_i)^2 \quad (3)$$

Minimal ist die Summe (3) genau dann, wenn die *partiellen Ableitungen* der Fehlerquadratsumme nach den Parametern a , b (bzw. a_k) gleich Null sind. Die mit den so gewonnenen optimalen Parametern a und b gebildete Ausgleichsfunktion wird *Regressionsfunktion* genannt.

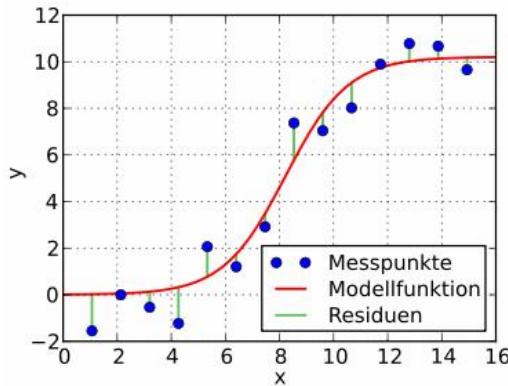


Abb. 6: Abweichungen der Modellfunktion von den Messpunkten (Quelle: Wikipedia [7])

Natürlich ließen sich die Parameter a , b (bzw. a_k) der Ausgleichsfunktion wie in unserem „naiven“ Beispiel auch bestimmen, indem man zwei (bzw. $r+1$) Punkte (x_i, y_i) in die Funktionsgleichung einsetzt und das resultierende Gleichungssystem mit zwei (bzw. $r+1$) Unbekannten löst. Aber das Ergebnis wäre nur hinsichtlich dieser drei Punkte eine optimale Näherung.

Die Güte der Näherung, die wir mit der resultierenden, für die gewählte Funktionsklasse optimalen Ausgleichsfunktion erhalten, können wir über die Fehlerquadratsumme (3) bestimmen: je kleiner q desto besser die Näherung. Falls $q = 0$ liegen alle Messwerte auf dem Funktionsgraphen.

Wenden wir dieses Gütemaß auf unsere Messwerte und die Ergebnisse des Robo Pro-Unterprogramms „NTC->T“ an, so erhalten wir $q = 203,4$. Das entspricht bei 75 Messwerten einer durchschnittlichen Abweichung von $1,65^\circ\text{C}$ – kein besonders überzeugender Wert.

Linearfunktion als Ausgleichsfunktion

Wählen wir zur Veranschaulichung des Verfahrens zunächst als Ausgleichsfunktion eine lineare Funktion. Anders als in unserer Beispielrechnung oben berücksichtigen wir nicht nur zwei, sondern alle n Messwerte unserer Messreihe (Tab. 1).

Dann ist nach (3) der folgende Summenwert zu minimieren:

$$q = \sum_{i=1}^n (a \cdot x_i + b - y_i)^2$$

Die Fehlerquadratsumme q ist minimal, wenn die partiellen Ableitungen nach a bzw. b Null sind, wenn also gilt:

$$\begin{aligned} \frac{\partial q}{\partial a} &= 2 \sum_{i=1}^n x_i (a \cdot x_i + b - y_i) = 0 \\ \Leftrightarrow a \sum_{i=1}^n x_i^2 + b \sum_{i=1}^n x_i &= \sum_{i=1}^n x_i \cdot y_i \end{aligned} \quad (4)$$

$$\begin{aligned} \frac{\partial q}{\partial b} &= 2 \sum_{i=1}^n (a \cdot x_i + b - y_i) = 0 \\ \Leftrightarrow a \sum_{i=1}^n x_i + b \cdot n &= \sum_{i=1}^n y_i \end{aligned} \quad (5)$$

Mit diesen beiden Formeln können wir nun die Parameter a und b der Linearen Ausgleichsfunktion bestimmen [8]:

$$a = \frac{n \sum_{i=1}^n x_i \cdot y_i - \left(\sum_{i=1}^n x_i \right) \left(\sum_{i=1}^n y_i \right)}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2} \quad (6)$$

$$b = \frac{\left(\sum_{i=1}^n y_i \right) \left(\sum_{i=1}^n x_i^2 \right) - \left(\sum_{i=1}^n x_i \right) \left(\sum_{i=1}^n x_i \cdot y_i \right)}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2} \quad (7)$$

Aus den Formeln (6) und (7) erhalten wir für $n = 2$ unsere bekannten Formeln (1) bzw. (2) für die Steigung und Konstante einer Gerade durch zwei Messpunkte in unserer einfachen Beispielrechnung oben.

Für unsere Messwerte aus Tab. 1 erhalten wir die Parameter $a = -0,0078$ und $b = 34,6255$. Die Fehlerquadratsumme liegt bei $q = 165,29$ (Abb. 7). Damit liefert diese Gerade sogar eine bessere Näherung als das „NTC->T“-Unterprogramm in Robo Pro.

$$K_{xy} = \frac{n \sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{\sqrt{\left(n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2 \right) \cdot \left(n \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2 \right)}} \quad (8)$$

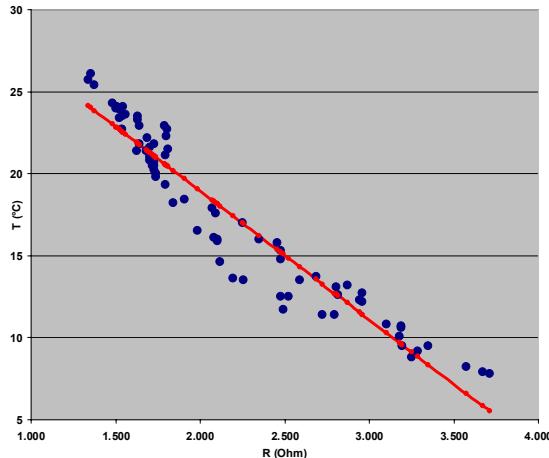


Abb. 7: Lineare Ausgleichsfunktion

Wie gut unsere Ausgleichsfunktion unsere Messpunkte annähert, lässt sich bei Linearen Ausgleichsfunktionen nicht nur über die Fehlerquadratsumme, sondern mit dem Korrelationskoeffizienten K_{xy} bestimmen (8). Er drückt die Güte der Annäherung als einen Wert zwischen 0 und 1 aus – je näher der Wert bei 1 liegt, desto besser ist unsere Näherung [8].

Für unsere Messwerte liegt K_{xy} bei 0,96. Das deutet auf einen fast linearen Kurvenverlauf in unserem Wertebereich hin – auch wenn man an der xy-Grafik in Abb. 4 erkennt, dass mit einer nichtlinearen Funktion möglicherweise eine noch bessere Annäherung an die Messwerte erreicht werden kann.

Polynom als Ausgleichsfunktion

Versuchen wir es also einmal mit einer Polynomfunktion r -ten Grades, eine bessere Ausgleichsfunktion hinzubekommen (9):

$$f(x) = \sum_{k=0}^r a_k \cdot x^k \quad (9)$$

Polynomfunktionen dienen vor allem dazu, eine Messwertkurve zu glätten. Mit einer Polynomfunktion vom Grad $r = n-1$ können wir den Punkteverlauf sogar *interpolieren*, denn es gibt immer eine Polynomfunktion vom Grad $n-1$, bei der alle unsere Messpunkte exakt auf der Kurve liegen – die Methode der kleinsten Quadrate liefert uns dann eine Ausgleichsfunktion ohne Fehler ($q = 0$).

Allerdings ist eine Polynomfunktion ($n-1$)-ten Grades für große n meist als Näherung für eine Messkurve nicht geeignet, da diese üblicherweise (streng) monoton steigt bzw. fällt, der Kurvenverlauf einer Polynomfunktion jedoch um die Messpunkte „schwingt“. Polynomfunktionen ab Grad fünf spielen daher in der Praxis als Ausgleichsfunktion keine Rolle.

Die Bestimmung der optimalen Parameter a_i einer Polynomfunktion ist etwas aufwändiger als bei einer Linearen Funktion, denn es müssen die Nullstellen von n partiellen Ableitungen berechnet werden. Dafür muss ein lineares Gleichungssystem mit $r+1$ Gleichungen und ebenso vielen Unbekannten (den Parametern a_k) gelöst werden.

Für den (noch recht einfachen) Fall $r = 2$ werden die Nullstellen der partiellen Ableitungen wie folgt bestimmt:

$$\begin{aligned} \frac{\partial q}{\partial a_2} &= 2 \sum_{i=1}^n (a_2 x_i^2 + a_1 x_i + a_0 - y_i) \cdot x_i^2 = 0 \\ \Leftrightarrow a_2 \sum_{i=1}^n x_i^4 + a_1 \sum_{i=1}^n x_i^3 + a_0 \sum_{i=1}^n x_i^2 &= \sum_{i=1}^n x_i^2 \cdot y_i \end{aligned}$$

$$\begin{aligned} \frac{\partial q}{\partial a_1} &= 2 \sum_{i=1}^n (a_2 x_i^2 + a_1 x_i + a_0 - y_i) \cdot x_i = 0 \\ \Leftrightarrow a_2 \sum_{i=1}^n x_i^3 + a_1 \sum_{i=1}^n x_i^2 + a_0 \sum_{i=1}^n x_i &= \sum_{i=1}^n x_i \cdot y_i \end{aligned}$$

$$\frac{\partial q}{\partial a_0} = 2 \sum_{i=1}^n (a_2 x_i^2 + a_1 x_i + a_0 - y_i) = 0$$

$$\Leftrightarrow a_2 \sum_{i=1}^n x_i^2 + a_1 \sum_{i=1}^n x_i + a_0 \cdot n = \sum_{i=1}^n y_i$$

Dieses lineare Gleichungssystem mit den drei Unbekannten a_0 , a_1 und a_2 lässt sich wie folgt in Matrixnotation darstellen und nach den vielleicht noch aus der Schule bekannten Methoden ([Gaußsches Eliminationsverfahren](#)) lösen:

$$\begin{bmatrix} \sum_{i=1}^n x_i^4 & \sum_{i=1}^n x_i^3 & \sum_{i=1}^n x_i^2 \\ \sum_{i=1}^n x_i^3 & \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i & n \end{bmatrix} \times \begin{bmatrix} a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n x_i^2 y_i \\ \sum_{i=1}^n x_i y_i \\ \sum_{i=1}^n y_i \end{bmatrix}$$

Wer das manuelle Lösen des Gleichungssystems vermeiden möchte, dem sei der [Online-Rechner von Arndt Brünner](#) wärmstens empfohlen. Auch mit der Funktion [Polyfit](#) in MATLAB oder dem freien [GNU Octave](#) lässt sich die Polynomfunktion automatisch bestimmen. Für unsere Messwerte aus Tab. 1 liefert das Gleichungssystem die folgenden Parameter (Abb. 8):

$$a_0 = 47,27, a_1 = -0,0196 \text{ und } a_2 = 2,4474 \cdot 10^{-6}.$$

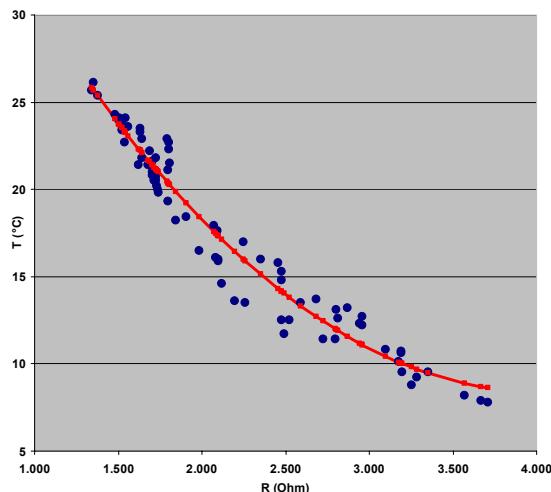


Abb. 8: Polynomfunktion als Ausgleichsfunktion

Die Summe der Fehlerquadrate liegt bei dieser Regressionsfunktion bei $q = 96,32$ – wieder ein etwas besseres Ergebnis.

Exponentialfunktion als Ausgleichsfunktion

Als nicht-lineare Ausgleichsfunktion für eine monotone Messkurve eignen sich auch Exponentialfunktionen:

$$f(x) = b \cdot e^{ax} \quad (10)$$

Bei einer Exponentialfunktion sind nicht nur lediglich zwei Parameter a und b zu optimieren. Mit zwei einfachen Substitutionen lässt sich zudem die Berechnung der partiellen Ableitungen nach a und b vermeiden, indem wir die Funktionsgleichung auf den einfachen Fall einer linearen Ausgleichsfunktion zurückführen:

$$\begin{aligned} \ln y &= \ln(b \cdot e^{ax}) = a \cdot x + \ln b \\ B &= \ln b, Y = \ln y \\ Y &= a \cdot x + B \end{aligned} \quad (11)$$

Aus den Formeln (6) und (7) können wir nun die Bestimmung der Parameter a und B ableiten:

$$a = \frac{n \sum_{i=1}^n x_i \cdot \ln y_i - \left(\sum_{i=1}^n x_i \right) \left(\sum_{i=1}^n \ln y_i \right)}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2} \quad (12)$$

$$B = \frac{\left(\sum_{i=1}^n \ln y_i \right) \left(\sum_{i=1}^n x_i^2 \right) - \left(\sum_{i=1}^n x_i \right) \left(\sum_{i=1}^n x_i \cdot \ln y_i \right)}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2} \quad (13)$$

Anschließend lässt sich aus B der Parameter b mit einer Potenzierung gewinnen: $b = e^B$. Mit den Messwerten aus Tab. 1 erhalten wir die folgenden Parameter unserer Regressionsfunktion (Abb. 9):

$$a = 0,0005 \text{ und } b = 50,1933.$$

Die erneut etwas kleinere Fehlerquadratsumme $q = \mathbf{98,85}$ bestätigt den guten Eindruck des Funktionsverlaufs aus Abb. 9: Die Exponentialfunktion liefert eine ähnlich gute Näherung an unsere Messwerte wie die Polynomfunktion.

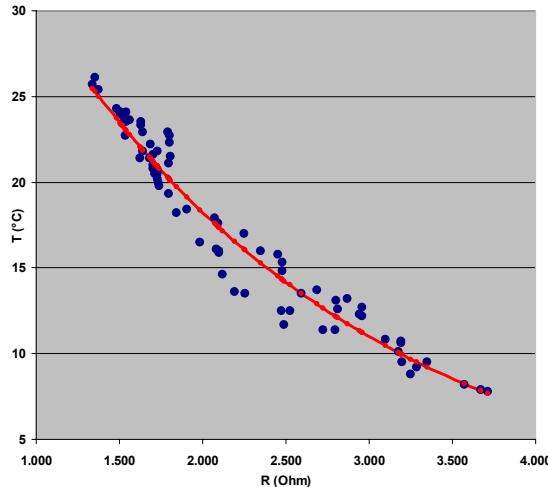


Abb. 9: Näherung durch eine Exponentialfunktion

Potenzfunktion als Ausgleichsfunktion

Das Vorgehen bei einer Potenzfunktion ähnelt dem der Exponentialfunktion: Auch bei ihr sind nur zwei Parameter a und b zu bestimmen:

$$f(x) = b \cdot x^a \quad (14)$$

Wieder können wir durch einfache Substitutionen die Berechnung der partiellen Ableitungen nach a und b auf den einfachen Fall einer linearen Ausgleichsfunktion zurückführen:

$$a = \frac{n \sum_{i=1}^n \ln x_i \cdot \ln y_i - \left(\sum_{i=1}^n \ln x_i \right) \left(\sum_{i=1}^n \ln y_i \right)}{n \sum_{i=1}^n (\ln x_i)^2 - \left(\sum_{i=1}^n \ln x_i \right)^2} \quad (16)$$

$$B = \frac{\left(\sum_{i=1}^n \ln y_i \right) \left(\sum_{i=1}^n (\ln x_i)^2 \right) - \left(\sum_{i=1}^n \ln x_i \right) \left(\sum_{i=1}^n \ln x_i \cdot \ln y_i \right)}{n \sum_{i=1}^n (\ln x_i)^2 - \left(\sum_{i=1}^n \ln x_i \right)^2} \quad (17)$$

$$\begin{aligned} \ln y &= \ln(b \cdot x^a) = a \cdot \ln x + \ln b \\ Y &= \ln y, B = \ln b, X = \ln x \\ Y &= a \cdot X + B \end{aligned} \quad (15)$$

Die Parameter a und B der solcherart „linearisierten“ Ausgleichsfunktion (15) bestimmen wir auch hier mit den angepassten Formeln (16) und (17) für unsere Messwerte aus Tab. 1. Wieder erhalten wir dabei den Parameter b aus B mit einer Potenzierung $b = e^B$:

$$a = -1,1582 \text{ und } b = 116867,91.$$

Die Güte dieser Ausgleichsfunktion liefert uns die Fehlerquadratsumme: $q = \mathbf{101,18}$ – fast so gut wie die bisher besten Näherungen. Abb. 10 veranschaulicht den Kurvenverlauf.

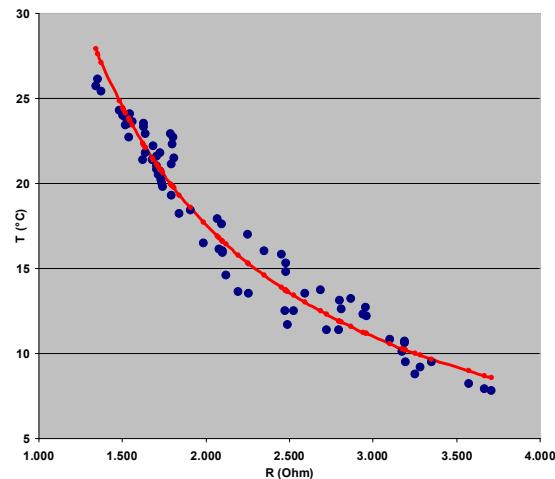


Abb. 10: Näherung durch eine Potenzfunktion

Steinhart-Hart-Gleichung als Ausgleichsfunktion

Für die Berechnung des Temperaturwerts T (in Kelvin) aus dem Widerstandswert R eines NTC wird üblicherweise die folgende *Steinhart-Hart-Gleichung* [5, 9] verwendet:

$$\frac{1}{T} = c(\ln R)^3 + b(\ln R) + a \quad (18)$$

Die Temperatur in °C erhält man aus T durch Subtraktion von 273,15.

Auf diese Funktion lässt sich die Methode der Kleinsten Quadrate nicht anwenden, denn in (18) dürfen wir nicht $y = 1/T$ substituieren – das Gauss-Verfahren ist nicht invariant unter Kehrwertbildung. Daher bestimmt man die Parameter a , b und c der Regressionsfunktion durch die Auswahl dreier Messwerte aus dem betrachteten Intervall und löst das daraus resultierende lineare Gleichungssystem mit drei Unbekannten:

$$\begin{bmatrix} 1 & \ln(x_1) & \ln(x_1)^3 \\ 1 & \ln(x_2) & \ln(x_2)^3 \\ 1 & \ln(x_3) & \ln(x_3)^3 \end{bmatrix} \times \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \frac{1}{y_1 + 273,15} \\ \frac{1}{y_2 + 273,15} \\ \frac{1}{y_3 + 273,15} \end{bmatrix}$$

Besonders einfach gelingt die Berechnung mit dem [Online Thermistor Calculator](#) der Stanford Research Systems Inc.

Für die Punkte (1.353 Ω; 26,1°C), (2.100 Ω; 15,9°C) und (3.177 Ω; 10,1°C) liefert das Verfahren die Lösung

$$\begin{aligned} a_0 &= 1,124018613 \cdot 10^{-3} \\ a_1 &= 3,422680237 \cdot 10^{-4} \\ a_2 &= -6,707557136 \cdot 10^{-7}. \end{aligned}$$

Die Summe der Fehlerquadrate liegt bei dieser Steinhart-Hart-Ausgleichsfunktion bei sehr guten $q = 96,09$ – die bisher beste Näherung (Abb. 11). Aber Vorsicht: Die Güte sagt hier mehr über die geeignete Wahl der drei Kurvenpunkte als über die

Qualität der Näherungsfunktion als solcher.

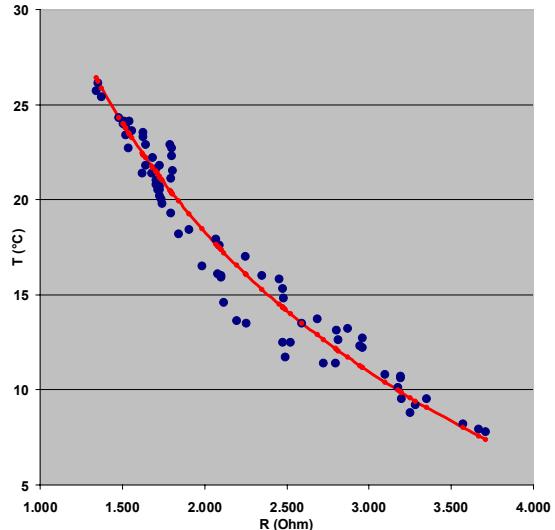


Abb. 11: Näherung mit der Steinhardt-Hart-Gleichung

Schließlich kann man auf die R/T-Kennlinien im [Datenblatt](#) des NTC (Nr. 1013) zurückgreifen und die Steinhardt-Hart-Ausgleichsfunktion über die darin in 5°-Schritten angegebenen Widerstandswerte bestimmen [10]. Das Ergebnis ist fast deckungsgleich mit den Werten des „NTC->T“-Unterprogramms (Abb. 4, 12).

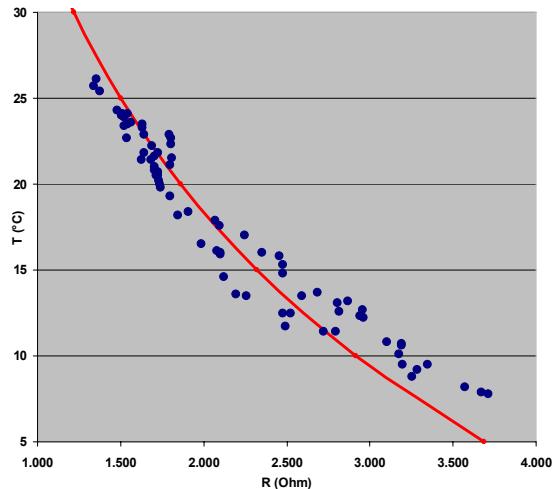


Abb. 12: NTC-Kennlinie nach [Datenblatt](#)

Fazit

Mit den vorgestellten Rechenwegen dürfte nun der Bestimmung einer Regressionsgleichung zu euren eigenen Messwerten beliebiger Sensoren nichts mehr im Wege stehen.

Sollten die aus den in diesem Beitrag vorgestellten Funktionsklassen gewöhnlichen Regressionsfunktionen keine zufrieden stellende Näherung darstellen, könnt ihr es auch mit Linearkombinationen verschiedener Funktionsklassen oder weiteren Substitutionen versuchen – die Vorgehensweise ist dieselbe, der Rechenweg im einen oder anderen Fall allerdings möglicherweise etwas komplizierter.

Bei euren eigenen Messungen solltet ihr berücksichtigen, dass Messergebnisse von zahlreichen Umwelteinflüssen wie Wind, Temperatur oder Luftfeuchtigkeit beeinflusst werden können. Auch in der Messanordnung selbst (korrodierte Kontakte, schlecht sitzende ft-Stecker etc.) können Fehlerquellen stecken. Zudem solltet ihr eine möglichst große Zahl (> 100) an Messungen durchführen, damit die Messungenauigkeiten möglichst gleich verteilt auftreten. Je genauer und zahlreicher eure Messwerte, desto präziser die spätere Ausgabe eures „eingemessenen“ fischertechnik-Messgeräts.⁵

Kennt ihr die Funktionsgleichung, dann ist die Programmierung der Umrechnung bspw. der NTC-Werte in Robo Pro kinderleicht – Abb. 13 zeigt ein Unterprogramm nach der Steinhart-Hart-Gleichung.

Die Parameter des Beispielprogramms könnt ihr gerne an die Ergebnisse aus euren eigenen Messungen anpassen.

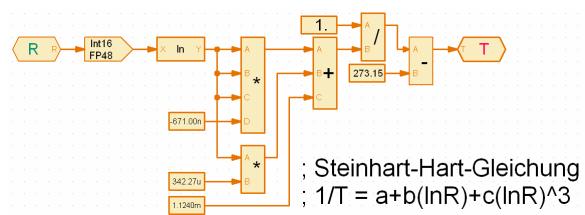


Abb. 13: Robo Pro-Unterprogramm zur Umrechnung des NTC-Widerstands in Temperaturwerte (Steinhart-Hart-Gleichung)

Quellen

- [1] Artur Fischer: *fischertechnik hobby. Experimente + Modelle*. Bd. 1-1. Fischer-Werke, Tumlingen, 1973.
- [2] Artur Fischer: *fischertechnik hobby. Experimente + Modelle*. Bd. 4-1/3. Fischer-Werke, Tumlingen, 1973.
- [3] Dirk Fox: *Der TX Controller als Messgerät*. [ft:pedia 1/2011](#), S. 33-36.
- [4] Dirk Fox: *Radar und Sonar*. [ft:pedia 2/2011](#), S. 4-8.
- [5] Wikipedia: [Heißleiter](#).
- [6] Wikipedia: [Carl Friedrich Gauß](#).
- [7] Wikipedia: [Methode der kleinsten Quadrate](#).
- [8] G. Krucker: [Ausgleichs- und Interpolationsrechnung](#). Kapitel 3, Vorlesungsskript, Institut für Informatik und angewandte Mathematik, Hochschule für Technik und Architektur, Bern, 1996/98.
- [9] John S. Steinhart, Stanley R. Hart: [Calibration curves for thermistors](#). In: Deep Sea Research and Oceanographic Abstracts, August 1968, Vol. 15, Issue 4, S. 497-503.
- [10] Siemens Matsushita: [Datenblatt B57164, K164](#). S. 112-113.

Vielen Dank an Thomas Püttmann für die wertvollen Anmerkungen zu einer Vorversion des Beitrags.

⁵ Wer misst, misst Mist: Messen ohne systematische Fehler ist eine Herausforderung – und zweifellos einen eigenen ft:pedia-Beitrag wert...

In eigener Sache

ft:pedia gedruckt

ft:pedia-Team

Einen oft geäußerten Wunsch vieler Leser der ft:pedia hat Ralf Knobloch wahr werden lassen: Sämtliche Ausgaben der ft:pedia gibt es – neben dem auch zukünftig kostenlosen pdf-Download – im Knobloch-Shop gedruckt und in Farbe.

Seit Ende 2012 bietet die Knobloch GmbH nicht nur die zur Convention 2011 gedruckte ft:pedia 3/2011, sondern auch alle weiteren Ausgaben der ft:pedia als EDV-Farbdruk an (Bestellnummer 80001 ff.). Die Druckqualität ist bestechend; die Ausgaben kommen in Spiralbindung und

mit transparenter Schutzfolie vor der Titel- und nach der Rückseite. Die Kosten pro Ausgabe liegen bei 8,95 € (inkl. MwSt.) – ggf. abzüglich Convention-Aussteller-rabatt. Im [Online-Shop](#) findet ihr die Ausgaben am einfachsten durch Suche nach „pedia“.



