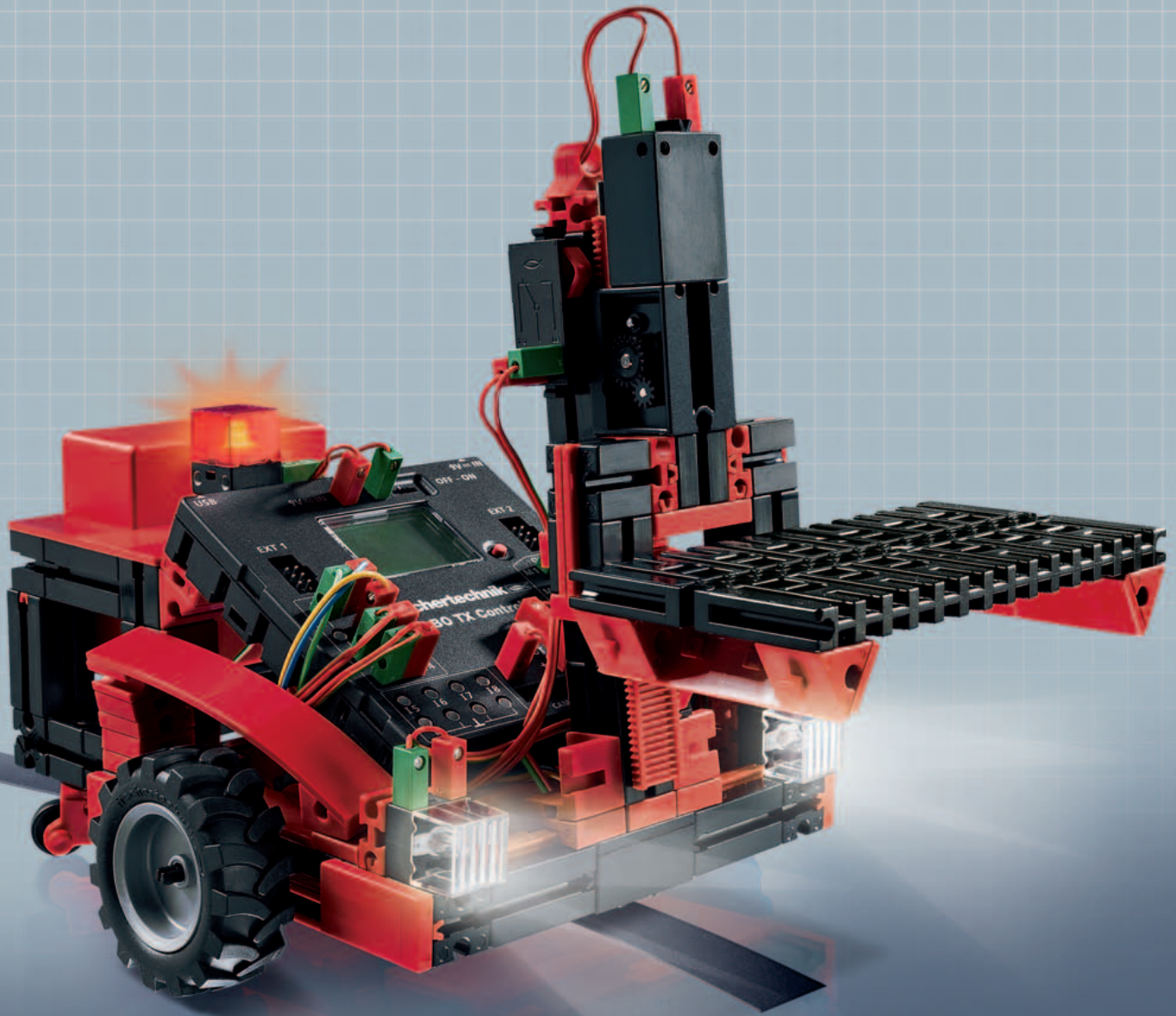


fischertechnik 

COMPUTING

Begleitheft



ROBO TX Training Lab

11 MODELS
MODEL 2

Willkommen beim ROBO TX Training Lab	3
Ein paar allgemeine Infos	3
Elektrizität	3
Über dieses Begleitheft	3
Roboter, der künstliche Mensch?	4
Computing, (fast) alles automatisch	4
Bauteilerklärungen	4
Encoder-Motoren	5
XS-Motor	5
Glühlampe	5
Linsenlampe	5
Fototransistor	5
Spursensor	6
Taster	6
Wärmesensor (NTC)	6
Ein paar Tipps	7
Erste Schritte	7
Starter-Modelle	8
Der Händetrockner	8
Die Ampel	9
Der Aufzug	10
Die Spülmaschine	10
Temperaturregelung	11
Roboter – die nächste Herausforderung	12
Basismodell	12
Der Spurensucher	13
Der Rasenmäher	14
Der Fußball-Roboter	15
Der Messroboter	16
Der Gabelstapler	17
Wenn's nicht gleich funktioniert ...	19
Und wie geht's jetzt weiter?	20

Willkommen beim ROBO TX Training Lab

Hallo!

Wir freuen uns, dass Du Dich für den Baukasten „ROBO TX Training Lab“ von fischertechnik entschieden hast. Und wir versprechen Dir, dass Dein Interesse belohnt werden wird. Denn mit diesem Baukasten kannst Du eine Menge interessanter Experimente durchführen und spannende Aufgaben lösen.

Beim Durchlesen dieses Bildschirm-Heftes und Ausprobieren der Experimente und Aufgaben wirst Du Schritt für Schritt lernen, wie man mit dem ROBO TX Controller von fischertechnik einfache und auch komplizierte Maschinen und Roboter steuern und programmieren kann.

Wie das beim Lernen nun mal so ist, kann man nicht sofort mit den schwierigsten Sachen anfangen, auch wenn die natürlich oft ein bisschen interessanter sind als die etwas einfacheren. Darum haben wir die Experimente und Aufgaben in diesem Heft so aufgebaut, dass Du mit jeder neuen Aufgabe etwas anderes dazu lernst, das Du dann wieder bei der nächsten Aufgabe einsetzen kannst.

Also keine Angst, wir fangen klein an und arbeiten uns dann gemeinsam zu den großen Robotern vor.

Jetzt wünschen wir Dir viel Spaß und Erfolg beim Experimentieren mit dem ROBO TX Training Lab.

Dein Team von

fischertechnik



Ein paar allgemeine Infos

Bevor wir mit dem Baukasten so richtig loslegen können, musst Du noch ein paar Dinge wissen. Die Bauteile, mit denen wir arbeiten werden, sind zwar sehr robust, aber wenn man sie nicht korrekt behandelt, können sie unter Umständen beschädigt werden.

Elektrizität

Wie Du sicherlich weißt, funktionieren sehr viele Bauteile des ROBO TX Training Lab mit elektrischem Strom. Und bei Sachen, die mit Strom zu tun haben, muss man besonders darauf achten, keine Fehler zu machen. Schau Dir darum die Bauanleitung immer genau an, wenn es um das Verkabeln von elektrischen Bauteilen geht.

Auf keinen Fall darfst Du den Plus- und den Minus-Pol einfach so miteinander verbinden, also kurzschließen. Dadurch kann der ROBO TX Controller, oder auch der Akku, beschädigt werden.

Elektrizität und Elektronik ist ein genau so interessantes Thema wie Robotik (also das, worum es in diesem Baukasten geht), und es gibt von fischertechnik einen Baukasten, der sich speziell mit diesen Themen beschäftigt. Wenn Du Dich also dafür interessierst, wirst Du mit dem „PROFI E-Tech“-Baukasten genau so viel Spaß haben wie mit dem ROBO TX Training Lab.



Über dieses Begleitheft

Dieses PDF-Begleitheft hat ein paar Funktionen, die es in einem gedruckten Heft nicht gibt und die Du vielleicht schon aus dem Internet kennst.

[Verknüpfungen innerhalb des Hefts](#)

Wenn irgendwo im Text etwas erwähnt wird, was an einer anderen Stelle in diesem Heft genauer erklärt ist (zum Beispiel die Bauteile), dann ist dieser Text dunkelblau geschrieben und unterstrichen. Du kannst den Text anklicken und damit automatisch auf die Seite blättern, auf der die Erklärung steht. Man nennt sowas einen „Querverweis“.

[Hintergrund-Infos](#)

Zum Teil gibt es in diesem Heft Begriffe oder Fremdwörter, für die man eine Erklärung brauchen kann. Diese Begriffe sind grün geschrieben und unterstrichen. Wenn Du den Text mit dem Mauszeiger berührst, erscheint ein Feld mit einer Erklärung.

Verknüpfung außerhalb dieses Heftes

Für ein paar Verknüpfungen brauchst Du eine Internetverbindung (zum Beispiel zur Internetseite von fischertechnik), bzw. ein installiertes ROBO Pro (für das Aufrufen der ROBO Pro Onlinehilfe). Diese Verknüpfungen sind hellblau geschrieben und unterstrichen.

Bilder

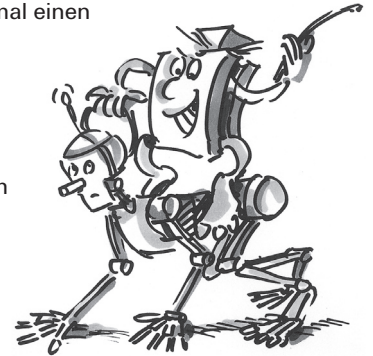
Ein Bild sagt mehr als 1000 Worte. Diesen Satz hast Du bestimmt schon einmal gehört. Und weil da sehr viel wahres dran ist, kannst Du durch Berühren der braun geschriebenen und unterstrichenen Wörter ein Bild einblenden, auf dem Du erkennen kannst was im Text gemeint ist.

Roboter, der künstliche Mensch?

An was denkst Du zuerst, wenn Du das Wort „Roboter“ hörst? Hast Du schon einmal einen Roboter gesehen? Im Kino oder Fernsehen? Oder vielleicht sogar schon in echt?

Es gibt unzählige verschiedene Arten von Robotern. Manche sehen ein bisschen aus wie ein Mensch, andere bestehen nur aus einem oder mehreren Armen. Was genau macht also einen Roboter zu einem Roboter?

Im Lexikon steht: „Roboter sind stationäre oder mobile Maschinen, die nach einem bestimmten Programm festgelegte Aufgaben erfüllen.“



Computing, (fast) alles automatisch

Roboter sind also Maschinen, die von einem Programm gesteuert werden. Und diese Steuerung von Maschinen (oder in unserem Fall Modellen) nennen wir „Computing“.

Mit dem „ROBO TX Training Lab“ kannst Du wunderbar in dieses Thema einsteigen. Denn der Baukasten enthält alles, was Du zum Bauen und Steuern von vielen verschiedenen Maschinen brauchst.

Die Programme zum Steuern der Modelle kannst Du am PC mit Hilfe der Software ROBO Pro erstellen und anschließend per USB- oder Bluetooth-Verbindung auf den ROBO TX Controller übertragen. Der Controller „kontrolliert“ und steuert dann das Modell entsprechend der Programmierung, die Du erstellt hast.

Bauteilerklärungen

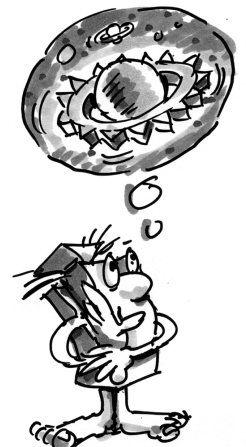
Das ist alles im Baukasten mit drin

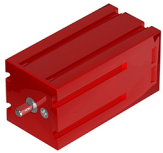
Zunächst einmal findest Du zahlreiche fischertechnik-Bausteine, außerdem Motoren, Lampen und Sensoren, sowie eine farbige Bauanleitung zum Bau der verschiedenen Modelle.

Wenn Du die Bausteine alle ausgepackt hast, musst Du einige Komponenten zuerst montieren, bevor Du loslegen kannst (z. B. Kabel und Stecker). Welche das genau sind, ist in der Bauanleitung unter „Montagetipps“ beschrieben. Erledige das am Besten gleich als Erstes.

Aktoren

Als Aktoren werden alle Bauteile bezeichnet, die eine Aktion ausführen können. Das bedeutet, wenn man sie an einen elektrischen Strom anschließt, werden sie in irgend einer Form „aktiv“. Meistens kann man das direkt sehen. Ein Motor dreht sich, eine Lampe leuchtet, usw.





Encoder-Motoren

Als Antrieb für unsere Roboter verwenden wir die zwei im Baukasten enthaltenen Encoder-Motoren. Auf den ersten Blick sind das normale Elektromotoren, die für eine Spannung von 9 Volt und eine Stromaufnahme von maximal 0,5 Ampere ausgelegt sind.

Die Encoder-Motoren können aber noch mehr: Zusätzlich zum Anschluss für die Stromversorgung des Motors haben sie noch eine Buchse für ein 3-poliges Anschlusskabel, über das man mit Hilfe des so genannten Encoders die Drehbewegung des Motors auswerten kann.

Der Encoder funktioniert so ähnlich wie der Tacho an einem Fahrrad. Ein Magnet (beim Fahrrad sitzt er meistens auf einer der Speichen) kommt bei jeder Umdrehung an einem Sensor (beim Fahrrad meistens an der Gabel befestigt) vorbei, wodurch der Sensor einen Impuls erzeugt. Diese Impulse kann man zählen, und zum Beispiel beim Tacho mit dem Umfang des Reifens multiplizieren. So erhält man die gefahrene Strecke.

Die Encoder an den fischertechnik Encoder-Motoren erzeugen pro Umdrehung der Motorwelle 3 Impulse. Und weil die Encoder-Motoren zusätzlich noch ein Getriebe mit einem [Übersetzungsverhältnis von 25:1](#) (sprich „25 zu 1“) haben, entspricht eine Umdrehung der Welle, die aus dem Getriebe kommt, 75 Impulsen des Encoders.

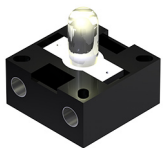


XS-Motor

Der XS-Motor ist ein Elektromotor, der genau so lang und so hoch ist wie ein fischertechnik-Baustein. Außerdem ist er sehr leicht. Dadurch kannst Du ihn an Stellen einbauen, an denen für die großen Motoren kein Platz ist.

Die beiden Getriebe, die ebenfalls im Baukasten dabei sind, passen genau an den XS-Motor.

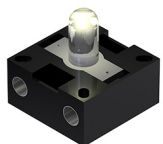
Der XS-Motor ist für eine Versorgungsspannung von 9 Volt und einen Stromverbrauch von maximal 0,3 Ampere ausgelegt.



Glühlampe

Zwei Glühlampen sind im Baukasten enthalten. Sie können sehr vielfältig verwendet werden, zum Beispiel als Signallichter bei einer Ampel, oder auch als Blinklicht an einem Roboter.

Die Glühlampen sind für eine Spannung von 9 Volt ausgelegt und verbrauchen ca. 0,1 Ampere Strom.



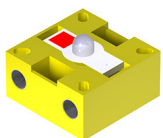
Linsenlampe

In diese Lampe ist eine Linse eingearbeitet, die das Licht bündelt. Sie sieht der Kugellampe sehr ähnlich. Du musst aufpassen, dass Du sie nicht verwechselst. Zur besseren Unterscheidung ist der Stecksockel dieser Lampe grau, während die Kugellampe einen weißen Sockel besitzt. Die Linsenlampe benötigst Du zum Bauen einer [Lichtschranke](#).

Die Linsenlampe ist wie die Glühlampen für eine Spannung von 9 Volt ausgelegt und verbraucht ca. 0,15 Ampere Strom.

Sensoren

Sensoren sind gewissermaßen die Gegenstücke zu den Aktoren. Denn sie führen keine Aktionen aus, sondern reagieren auf bestimmte Situationen und Ereignisse. Ein Taster reagiert zum Beispiel auf einen „Knopfdruck“, indem er einen elektrischen Strom durchlässt oder unterbricht. Ein Wärmesensor reagiert auf die Temperatur in seiner Umgebung.



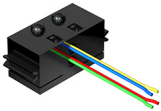
Fototransistor

Man bezeichnet den Fototransistor auch als „Helligkeitssensor“. Das ist ein „Fühler“, der auf Helligkeit reagiert.

Er bildet bei einer [Lichtschranke](#) das Gegenstück zur Linsenlampe. Bei großer Helligkeit, also wenn der Transistor von der Linsenlampe angestrahlt wird, leitet er Strom. Wird der Lichtstrahl unterbrochen, leitet der Transistor keinen Strom.

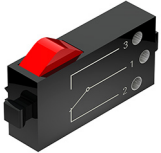
Achtung:

Beim Anschluss des Fototransistors an die Stromversorgung musst Du auf die richtige Polung achten. Der Plus-Pol muss an der roten Markierung am Fototransistor angeschlossen werden.



Spursensor

Der **Infrarot**-Spursensor ist ein digitaler Sensor zur Erkennung einer schwarzen Spur auf weißem Untergrund mit einem Abstand von 5 – 30 mm. Er besteht aus zwei Sende- und Empfängerelementen. Als Anschluss benötigst Du zwei digitale Eingänge und die 9 Volt Spannungsversorgung (Plus- und Minus-Pol) am [ROBO TX Controller](#).



Taster

Der Taster wird auch Berührungssensor genannt. Beim Betätigen des roten Knopfes wird mechanisch ein Schalter umgelegt, es fließt Strom zwischen den Kontakten 1 (mittlerer Kontakt) und 3. Gleichzeitig wird der Kontakt zwischen den Anschlüssen 1 und 2 unterbrochen. So kannst du den Taster auf zwei verschiedene Arten verwenden:

Als „Schließer“:

Kontakte 1 und 3 werden angeschlossen.

Taster gedrückt: Es fließt Strom.

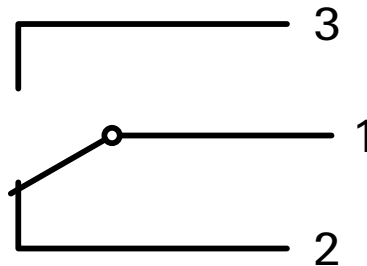
Taster nicht gedrückt: es fließt kein Strom

Als „Öffner“:

Kontakte 1 und 2 werden angeschlossen.

Taster gedrückt: Es fließt kein Strom.

Taster nicht gedrückt: Es fließt Strom.



Wärmesensor (NTC)

Bei diesem Bauteil handelt es sich um einen Wärmesensor, mit dem man Temperaturen messen kann. Bei 20°C beträgt sein **elektrischer Widerstand** 1,5kΩ (kilo-Ohm). NTC bedeutet Negativer Temperatur Coeffizient. Das heißt einfach, dass der Widerstandswert mit steigender Temperatur sinkt.

Die Informationen, die uns die Sensoren liefern (z. B. hell-dunkel, gedrückt - nicht gedrückt, Temperaturwert) kann man, wie wir später noch sehen werden, über den [ROBO TX Controller](#) an den PC weiterleiten und dann mit Hilfe der Software z. B. einen Motor so programmieren, dass ein Ventilator bläst, sobald eine Lichtschranke unterbrochen wird.



Software ROBO Pro 2.x

ROBO Pro ist eine grafische Programmieroberfläche, mit der Du die Programme für den [ROBO TX Controller](#) erstellen kannst.

„Grafische Programmieroberfläche“ bedeutet, dass Du die Programme nicht Zeile für Zeile von Hand „schreiben“ musst, sondern mit Hilfe von grafischen Symbolen einfach bildlich zusammenstellen kannst. Ein Beispiel für so ein Programm findest Du auf dem Bild rechts.

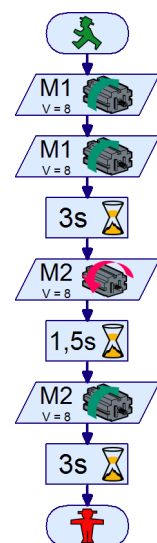
Wie man so ein Programm genau erstellt, ist in der [ROBO Pro Onlinehilfe](#) in den Kapiteln 3 und 4 ausführlich beschrieben.

Die Installation von ROBO Pro und dem Treiber für den ROBO TX Controller ist in der Installationsanleitung beschrieben, die sich im Baukasten befindet.

Die Software befindet sich auf der selben CD wie dieses Begleitheft.

Wenn Du mit der Installation von ROBO Pro fertig bist, kannst Du ROBO Pro gleich starten, denn als nächstes brauchen wir die Onlinehilfe der Software.

Am Besten liest Du Dir jetzt die beiden ersten Kapitel der [ROBO Pro Onlinehilfe](#) durch. Dabei lernst Du die Software gleich ein bisschen kennen, so dass wir anschließend direkt starten können mit dem Experimentieren.





ROBO TX Controller

Der ROBO TX Controller ist das Herzstück dieses Computing-Baukastens. Denn er steuert die [Aktoren](#), und wertet die Informationen der [Sensoren](#) aus.

Für diese Aufgabe verfügt der ROBO TX Controller über zahlreiche Anschlüsse, an die Du die Bauteile anschließen kannst. Welche Bauteile man an welche Anschlüsse anschließen kann, und was die Funktionen der Anschlüsse sind, ist in der Bedienungsanleitung zum ROBO TX Controller beschrieben.

Ein besonderer Leckerbissen ist die integrierte Bluetooth-Schnittstelle. Über sie kannst Du ohne Kabel Deinen PC mit dem ROBO TX Controller verbinden. Oder auch mehrere Controller mit dem PC und untereinander.

Wie der Controller mit den einzelnen Bauteilen umgeht und was diese im Einzelnen machen sollen, legst Du durch das Programm fest, das Du in der [Software ROBO Pro](#) schreibst.



Stromversorgung (nicht enthalten)

Weil ja wie Du weißt viele der Bauteile vom ROBO TX Training Lab mit Strom funktionieren, brauchst Du natürlich auch eine Stromversorgung.

Am besten geeignet ist hierfür das fischertechnik Accu Set. Es ist im ROBO TX Training Lab nicht enthalten.

Ein paar Tipps

Experimentieren macht am meisten Spaß, wenn die Experimente auch funktionieren. Darum solltest Du beim Aufbauen der Modelle ein paar Grundregeln beachten:

- **Sorgfältig arbeiten**

Nimm Dir Zeit und schau genau in die Bauanleitung für das Modell. Wenn man hinterher nach einem Fehler suchen muss, dauert es noch viel länger.

- **Beweglichkeit aller Teile prüfen**

Kontrolliere beim Zusammenbauen immer wieder, ob sich Teile, die sich bewegen sollen, auch leicht bewegen lassen.

- **Interface-Test benutzen**

Bevor Du beginnst, ein Programm für ein Modell zu schreiben, solltest Du alle an den ROBO TX Controller angeschlossenen Teile mit Hilfe des Interface-Test von ROBO Pro testen. Wie das genau funktioniert ist in der [ROBO Pro Onlinehilfe](#) in Kapitel 2.4 erklärt.

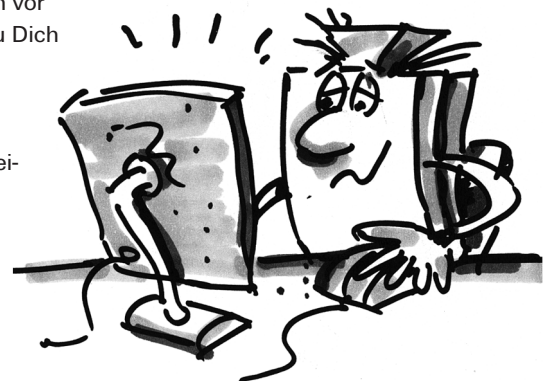


Erste Schritte

So. Nach all den Vorbereitungen und Informationen kannst Du jetzt endlich loslegen.

Weil Du zum Experimentieren außer den fischertechnik-Bauteilen vor allem auch mit der [Software ROBO Pro](#) arbeiten wirst, solltest Du Dich als Erstes damit vertraut machen und lernen, wie man ein Programm erstellt. Und weil das in den Kapiteln 3 und 4 der [ROBO Pro Onlinehilfe](#) wirklich hervorragend erklärt ist, solltest Du am Besten damit weitermachen, diese Kapitel sorgfältig durchzuarbeiten.

Auch hier gilt der Tipp: Nimm Dir Zeit und beschäftige Dich damit, dann wirst Du anschließend mit den Modellen umso mehr Freude haben.

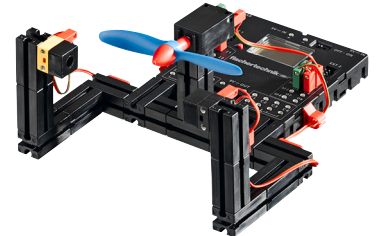


Starter-Modelle

Nachdem Du die Kapitel 3 und 4 der [ROBO Pro Onlinehilfe](#) durchgelesen hast, kannst du jetzt schon einige Modelle des Baukastens programmieren. Deshalb wollen wir auch sofort loslegen. Immer, wenn Du ein Modell fertig gebaut und verkabelt hast, prüfe mit Hilfe des [Interface-Tests](#), ob am [ROBO TX Controller](#) alle Aus- und Eingänge richtig angeschlossen sind und die [Sensoren](#), [Motoren](#) und [Lampen](#) richtig funktionieren.

Der Händetrockner

In Deiner Schule wurden auf der Toilette neben den Waschbecken neue Händetrockner installiert. Diese sind mit einer [Lichtschranke](#) versehen, über die man den Lüfter ein- und ausschalten kann.



- Baue zunächst das Modell wie in der Bauanleitung beschrieben auf.



Aufgabe 1:

- Der Händetrockner soll nun so programmiert werden, dass, sobald die [Lichtschranke](#) unterbrochen wird, der Lüfter ein- und nach 5 Sekunden wieder ausgeschaltet wird.

Programmiertipps:

- Schalte im Programmablauf zuerst die Lampe für die [Lichtschranke](#) am [Ausgang M2](#) ein.
- Danach wartest Du eine Sekunde, damit der [Fototransistor](#) Zeit hat, auf das Licht zu reagieren. Erst dann funktioniert die [Lichtschranke](#) richtig.
- Dann fragst Du den [Fototransistor](#) am [Eingang I1](#) ab. Ist der Wert „1“ ([Lichtschranke](#) nicht unterbrochen), soll der Eingang in einer Schleife dauernd abgefragt werden.
- Sobald der Wert „0“ wird ([Lichtschranke](#) unterbrochen), schaltest Du den Motor M1 ein und nach 5 Sekunden wieder aus.
- Danach soll wieder der [Fototransistor](#) abgefragt werden usw.

Starte Dein Programm mit dem [Start-Button](#) und überprüfe, ob es wie gewünscht funktioniert. Wenn ja, bist Du auf dem besten Weg ein professioneller ROBO Pro Programmierer zu werden.

Funktioniert es noch nicht, versuche herauszufinden woran es liegt:

- Mit dem [Interface-Test](#) kannst Du überprüfen, ob alle Ein- und Ausgänge funktionieren und richtig angeschlossen sind.
- Während das Programm läuft, kannst Du dem Programmablauf anhand der rot markierten Bausteine folgen. Damit kannst Du schnell erkennen, wo sich ein Fehler eingeschlichen hat.
- Zuletzt kannst Du Dein Programm mit dem fertigen Beispielprogramm vergleichen, das Du über das Symbol rechts aufrufen kannst:

Nachdem diese Hürde genommen ist, wollen wir die Aufgabenstellung etwas verändern:



Aufgabe 2:

- Dem Rektor, der stets darauf bedacht ist Energie zu sparen, gefällt es nicht, dass der Händetrockner immer noch eine gewisse Zeit weiterläuft, obwohl die Hände bereits trocken sind. Er fordert Dich auf, das Programm so zu gestalten, dass der Lüfter abschaltet, sobald die Hände zurückgezogen werden. Kein Problem für Dich, oder?

Programmiertipps:

- Wie im ersten Programm fragst Du mit einer Verzweigung den [Fototransistor](#) I1 ab. Ist der Wert „0“, schaltest Du den Motor M1 ein, ist der Wert „1“, schaltest Du den Motor M1 aus usw.
- Auch zu dieser Aufgabe gibt es für den Notfall ein fertiges Programm:

Die Ampel

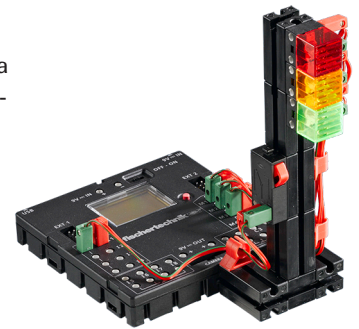
Vor Eurem Haus wurde eine Ampel aufgestellt. Da der Monteur von der Ampelfirma sehr unter Zeitdruck steht, bietest Du ihm an, die Programmierung der Ampelsteuerung für ihn zu übernehmen.

Der Mann erklärt Dir, wie die Steuerung funktionieren soll. Baue aber zuerst das Modell auf.



Aufgabe 1:

- Die Ampel soll zunächst auf Grün stehen. Wird der [Taster I1](#) von einem Fußgänger gedrückt, soll die Ampel 3 Sekunden später auf gelb und nach weiteren 4 Sekunden auf Rot wechseln. Die Rotphase soll 10 Sekunden dauern, die anschließende Rot-Gelb-Phase 3 Sekunden. Dann soll es wieder Grün werden.



Programmiertipps:

- Die verschiedenen Lampen gehören zu folgenden Interfaceausgängen:
 - Rot – M1
 - Gelb – M2
 - Grün – M3
- Schalte die Lampen so hintereinander ein und aus, dass der gewünschte Ablauf zustande kommt.
- Das fertige Programm kannst Du wieder durch Klicken auf das Bild rechts laden:



Aufgabe 2:

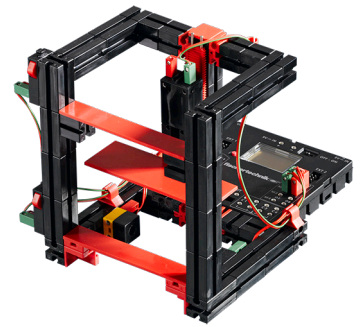
- Am nächsten Tag ruft Dich der Monteur der Ampelfirma an. Er hat vergessen Dir zu sagen, dass sich in dem Schaltkasten auf dem Gehweg ein Schalter I2 befindet, der die Ampel auf gelbes Blinklicht schalten soll, sobald er betätigt wird. Du sicherst dem Monteur zu, diese Funktion noch schnell in Dein Programm zu integrieren.

Programmiertipps:

- Befestige an Deinem Ampelmodell einen zweiten [Taster](#) und schließe ihn am [Eingang I2](#) an.
- Frage mit einer weiteren Verzweigung den [Eingang I2](#) ab. Wird der [Taster I2](#) gedrückt, verzweigt der Ablauf zum Blinklicht. Ansonsten läuft die Ampelsteuerung wie in Aufgabe 1 ab.
- Das Blinklicht erhältst du durch Ein- und Ausschalten der [Lampe M2](#) im Zeitabstand von 0,5 Sekunden. Verwende dazu ein Unterprogramm. Wie man ein Unterprogramm erstellt, kannst Du im Kapitel 4 der [ROBO Pro Onlinehilfe](#) nachlesen.
- Das Beispielprogramm findest Du wie gewohnt durch Klicken auf das Symbol. Versuche aber, bevor Du nachschaust, erst einmal selbst auf die Lösung zu kommen. Viel Erfolg!

Der Aufzug

Dein Nachbar hat in seiner Werkstatt einen Lastenaufzug eingebaut, damit er seine schweren Behälter nicht mehr die Treppe hinauf in den ersten Stock tragen muss. Jetzt benötigt er nur noch eine Steuerung, die Du natürlich gerne für ihn programmierst.



Aufgabe 1:

- Programmiere den Aufzug so, dass er zu Beginn nach unten in seine Ausgangsposition fährt. In dieser Position ist die [Lichtschranke](#) an I3 unterbrochen. Der Aufzug soll, wenn einer der beiden [Taster](#) gedrückt wird (I1 im Erdgeschoss oder I2 im ersten Stock) jeweils in das andere Stockwerk fahren.

Programmiertipps:

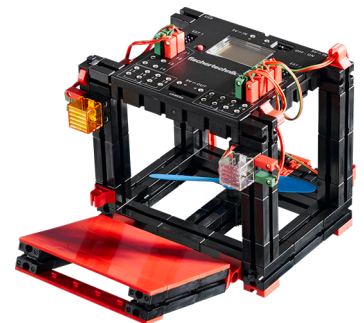
- Wenn die [Lichtschranke](#) unterbrochen ist, steht der Aufzug unten, wenn sie nicht unterbrochen ist, gehst Du davon aus, dass sich der Aufzug im ersten Stock befindet.
- Wie weit er von unten nach oben fährt, steuerst Du über die Zeit, während der der Motor eingeschaltet ist.
- Auch wenn Du das bestimmt auch ohne Hilfe schaffst, für alle Fälle haben wir wieder einen Lösungsvorschlag:

Bevor Du Dich an die nächsten Programmieraufgaben wagst, solltest Du wieder die [ROBO Pro Onlinehilfe](#) öffnen. Arbeite dort das Kapitel 5 sorgfältig durch. Schalte in ROBO Pro auf Level 3 um. Langsam werden die Programmieraufgaben etwas anspruchsvoller. Wir verwenden Analogeingänge, Bedienelemente, Operatoren und Variablen. Aber wenn Du die ROBO Pro Onlinehilfe aufmerksam liest, wird es Dir später leicht fallen damit umzugehen.

Die Spülmaschine

Als Nächstes willst Du Dich an die Programmierung einer Spülmaschine wagen. Die Spülmaschine soll folgende Funktionen haben:

- [Taster](#) zum Ein- und Ausschalten (I1)
- [Taster](#), der erkennt ob Tür geschlossen ist (I2)
- Spülfunktion (Propeller an M1)
- Trockenfunktion (Rote [Lampe](#) an M2)
- Anzeige, dass die Maschine eingeschaltet ist (Orange [Lampe](#) an M3)
- Anzeige, in welchem Betriebszustand sich die Maschine befindet (Transparente [Lampe](#) an M4)
 - Schnell blinken: Maschine spült
 - Langsam blinken: Maschine trocknet
 - Dauerlicht: Maschine ist fertig



Aufgabe 1:

- Erstelle ein Spülprogramm, das erst anfängt, wenn die Tür geschlossen ist und der Startknopf I1 betätigt wird. Als erstes wird gespült, danach getrocknet. Der Betriebszustand soll über die beiden Lampen an M3 und M4 angezeigt werden.

Programmiertipps:

- Hier wieder unser Lösungsvorschlag:



Aufgabe 2:

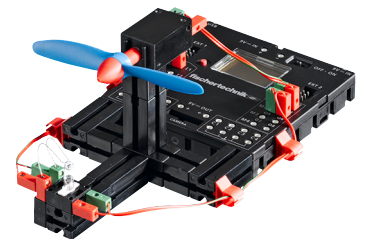
- Wird die Tür geöffnet, soll der Spülvorgang unterbrochen werden. Nach dem Schließen der Türe macht das Programm an der Stelle weiter, an der es unterbrochen wurde.
- Zeige zusätzlich den Betriebszustand der Spülmaschine am Display des [ROBO TX Controllers](#) an.

Programmiertipps:

- Zugegeben, diese Aufgabe hat es schon in sich. Wenn Du sie alleine lösen kannst: Super! Falls Du aber trotz allem Nachdenken nicht weiter kommst, kein Problem. Schau einfach mal in unseren Lösungsvorschlag.

Temperaturregelung

Bei Euch zu Hause wurde eine neue Klimaanlage installiert. Natürlich hast Du den Installateur sofort gefragt, wie die Temperaturregelung funktioniert. Er hat Dir bereitwillig erklärt, dass ein Temperatursfühler ständig die aktuelle Temperatur misst. Sobald ein oberer Grenzwert überschritten wird, schaltet die Kühlung ein. Wird hingegen ein unterer Grenzwert unterschritten, schaltet die Kühlung aus und die Heizung ein. Nun willst Du anhand des Modells „Temperaturregelung“ versuchen, ebenfalls einen solchen [Regelkreis](#) zu programmieren. Baue zuerst das Modell.



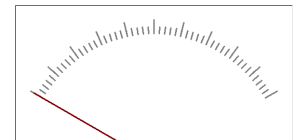
Aufgabe 1:

Die Heizung wird simuliert durch die [Linsenlampe M2](#). Als „Kühlaggregat“ dient das Gebläse am [Ausgang M1](#). Zur Temperaturmessung verwenden wir den [NTC-Widerstand](#) am [Eingang I2](#).

- Programme das Modell so, dass oberhalb eines bestimmten Temperaturwerts die Heizung aus- und das Gebläse eingeschaltet wird. Dieses soll so lange kühlen, bis ein unterer Grenzwert erreicht ist. Dann soll das Gebläse aus- und die Heizung eingeschaltet werden.
- Der aktuelle Wert des Analogeingangs soll an einem Messgerät und einer Textanzeige in ROBO Pro ausgegeben werden, sowie am Display des [ROBO TX Controllers](#).

Programmiertipps:

- **Beachte:** Der Widerstandswert des [NTC-Widerstands](#) sinkt mit steigender Temperatur. Der obere Temperaturgrenzwert ist also der kleinste Wert von I2. Bei diesem Grenzwert soll das Gebläse einschalten. Der untere Temperaturgrenzwert ist der größte Wert von I2. Bei diesem Grenzwert soll die Heizung einschalten.
- Welchen Wert I2 bei Zimmertemperatur besitzt, kannst Du mit dem [Interface-Test](#) herausfinden. Du schaltest die Lampe M2 ein und beobachtest, wie weit der Wert nach unten geht. Dann schaltest Du das Gebläse ein und findest heraus, wie weit der Wert ansteigt. Dementsprechend wählst Du die Grenzwerte für Heizen und Kühlen aus.
- Zeige den Wert des Analogeingangs in Deinem Programm mit einer Textanzeige und oder mit einem Messgerät an (siehe auch [ROBO Pro Onlinehilfe](#) Kapitel 8.1).



Temp. = 0

- Durch einen Klick auf das Symbol rechts kannst Du das fertige Programm öffnen.



Roboter – die nächste Herausforderung

Basismodell

Mit diesem Modell wollen wir gemeinsam herausfinden, wie man einen fahrenden Roboter steuern kann. Wie bringt man ihn zum Fahren, wie funktioniert die Lenkung, und kann man seine Genauigkeit vielleicht noch verbessern? Diese Fragen werden wir mit Hilfe der Aufgaben in diesem Kapitel beantworten.

Aber zuerst musst du natürlich den Roboter zusammenbauen. Die Beschreibung findest Du wie immer in der Bauanleitung.



- Bau den Roboter zusammen, wie es in der Bauanleitung beschrieben ist.
- Lass Dir Zeit beim Zusammenbauen. Schau Dir die Zeichnungen in der Bauanleitung genau an, auch die Verkabelung. Wenn Du die Bauteile anders mit dem [ROBO TX Controller](#) verbindest als in der Bauanleitung beschrieben, wird sich der Roboter möglicherweise anders verhalten als Du es erwartest.
- Prüfe nach dem Zusammenbauen alle an den [ROBO TX Controller](#) angeschlossenen Bauteile über den [Interface-Test](#) der [ROBO Pro Software](#). Wenn Du die Motoren nach links drehen lässt, muss der Roboter vorwärts fahren.



Aufgabe 1: Einfach mal gerade aus (Level 1)

- Lass den Roboter 3 Sekunden geradeaus fahren (**nicht auf dem Tisch, Absturzgefahr!**), und anschließend wieder 3 Sekunden gerade zurück.
 - *Ist der Roboter wirklich wieder auf seinem Ausgangspunkt angekommen?*
- Wiederhole das Programm mehrmals, und beobachte ob sich der Roboter wirklich genau gerade vor und zurück bewegt.

Programmiertipps:

- Auch wenn die Aufgabe sicher kein Problem für Dich ist, hier ist unser Vorschlag:

Die Lenkung

Auch wenn es Spaß macht, dem Roboter beim gerade aus fahren zu zu schauen, ist das doch etwas eintönig. Darum soll er nun lernen, um Kurven zu fahren. Und wie geht das? Ganz einfach:



Aufgabe 2: Auch mal eine Kurve fahren (Level 1)

- Lass den Roboter wieder 3 Sekunden geradeaus fahren (beide Motoren drehen gleich schnell), dann ändere für 1 Sekunde die Drehrichtung des rechten Motors (M1), und lass den Roboter anschließend wieder 3 Sekunden lang geradeaus fahren (also beide Motoren gleich schnell in die selbe Richtung).
- Finde heraus, wie lange Du die Motoren in unterschiedliche Richtungen laufen lassen musst, damit der Roboter sich um 90° dreht.

Programmiertipps:

- Verändere hierzu die Wartezeit nach dem Steuerelement, in dem die Richtung des zweiten Motors geändert wird.
- Das fertige Programm kann Du wieder durch Klicken auf das Symbol rechts öffnen:



Aufgabe 3: Eine Figur abfahren (Level 2)

- Nachdem Du jetzt weißt, wie lange Du die Drehrichtung eines Motors umkehren musst, damit der Roboter nach rechts oder links abbiegt, programmiere den Roboter so, dass er ein Viereck abfährt und anschließend wieder auf seiner Ausgangsposition ankommt.
 - *Prüfe mit einer Markierung, ob der Roboter wirklich wieder genau auf seine Ausgangsposition fährt.*

Programmiertipps:

- Für das „Um-die-Ecke-fahren“ kannst Du ein Unterprogramm erstellen. Dadurch bleibt Dein Hauptprogramm übersichtlicher.
- Sicherlich hast Du die Lösung für die Aufgabe schon im Kopf. Aber für alle Fälle gibt's hier wieder einen Vorschlag von uns:

Immer das Selbe, und doch nicht gleich?

Wie Du bestimmt gemerkt hast, ist die [Wiederholgenauigkeit](#) des Roboters noch verbesserungsfähig. Auch wenn er mehrmals genau die selbe Aufgabe ausführt, ist das Ergebnis nicht immer das gleiche. Das hat verschiedene Gründe. Einer davon ist, das sich die beiden Motoren nicht genau gleich schnell drehen. Zum Beispiel kann das Getriebe am einen Motor schwerer laufen als das am anderen. Und weil beide Motoren mit der selben Spannung (9 Volt) betrieben werden, dreht sich dann eben ein Motor langsamer als der andere. Da wir bisher unseren Roboter über Wartezeiten gesteuert haben, hat sich vielleicht ein Rad während dieser Zeit weiter gedreht als das andere.

Die Lösung wäre also, beide Motoren genau gleich schnell drehen zu lassen. Und genau das ist mit den [Encoder-Motoren](#) ganz einfach machbar.



Aufgabe 4: Encoder-Motoren nutzen

- Wiederhole die letzten drei Aufgaben, und benutze an Stelle der normalen [Motorausgangs-](#) und [Wartezeit-Elemente](#) die [Encodermotor-Elemente](#). Wie man sie benutzt ist in der [ROBO Pro Onlinehilfe](#) in Kapitel 11.6 beschrieben.

Programmiertipp:

- Mit dem [Encodermotor-Element](#) kannst Du mit einem Programmelement beide Motoren gleichzeitig ansteuern. Über das [Distanz-Eingabefeld](#) stellst Du sicher, dass jeder Motor sich auch wirklich so weit dreht, wie er soll.
- Die Lösungsvorschläge kannst Du wieder durch Klicken auf die Symbole rechts öffnen:

Für das Zählen der Impulse an den schnellen [Zähleingängen C1-C4](#) brauchst du kein weiteres Programmelement in [ROBO Pro](#). Dem Motor M1 wird intern automatisch der Zähleingang C1 zugeordnet, M2 gehört zu C2 usw.

Hinweis:

Sollte dein Modell trotz Verwendung der [Encodermotor-Elemente](#) nicht geradeaus fahren, kann die Ursache beim Modell selbst liegen. Wenn z. B. eine Nabenmutter, die die Kraft von der Achse auf die Räder überträgt, nicht fest genug angezogen ist, dreht die Achse durch und das Modell fährt eine Kurve, obwohl sich die Motoren gleich schnell drehen. Ziehe deshalb die Nabenmuttern ganz fest an.

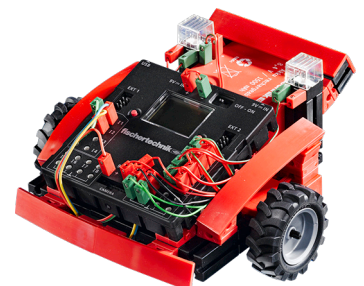
Der Spurensucher



Dein Roboter kann jetzt geradeaus fahren und abbiegen. Und bisher macht er das eben nur genau so, wie Du es ihm mit dem Programm vorgibst.

Aber ein Roboter soll ja eigentlich möglichst selbstständig reagieren können. Darum wollen wir ihm jetzt etwas geben, auf das er reagieren kann: Eine schwarze Linie als Markierung auf dem Boden.

Das Ziel ist, dass der Roboter die schwarze Linie sucht, und an ihr entlangfährt.



Aber eins nach dem anderen. Als Erstes musst Du das Basismodell zum Spurensucher umbauen. Wie das geht, steht natürlich in der Bauanleitung.

Nachdem Du das Modell fertig umgebaut hast, solltest Du mit dem [Interface-Test](#) überprüfen, ob alle Bauteile richtig am [ROBO TX Controller](#) angeschlossen sind und funktionieren. Den [Spursensor](#) kannst Du testen, in dem Du ihn über die schwarze Spur des Parcours hältst und seitlich bewegst. Die Signale an den Eingängen, an denen der [Spursensor](#) angeschlossen ist, sollten sich dadurch verändern.

Denk daran, die Eingänge im [Interface-Test](#) auf „Digital 10V (Spursensor)“ einzustellen.



Aufgabe 1: Eine Spur erkennen (Level 2)

- Programme den Roboter so, dass er einer geraden schwarzen Spur folgt, auf die er gesetzt wird. Verliert er die Spur oder ist sie zu Ende, soll der Roboter stehen bleiben und die beiden Lampen sollen jeweils 3-mal blinken. Verwende für diese Aufgabe den Parcours 1a aus dem Baukasten.

Programmiertipps:

- Zuerst fragst Du die beiden Eingänge des [Spursensors](#) ab. Empfangen beide Eingänge das Signal „0“, steht der Roboter auf der schwarzen Spur. Du kannst ihn losfahren lassen.
- Die Blinkfunktion legst Du in einem Unterprogramm ab.
- Verwende für das Geradeausfahren wieder das [Encodermotor-Element](#), allerdings ohne Distanzeingabe.
- Das fertige Programm findest Du hier:

Reagieren kann Dein Roboter jetzt. Allerdings ist die Funktion doch recht eingeschränkt. Besser wäre es, wenn der Roboter anstatt anzuhalten seine Richtung korrigieren würde, um der Spur zu folgen.



Aufgabe 2: Der Spur folgen (Level 2)

- Erweitere Dein Programm um die Funktion, dass der Roboter beim Verlassen der Spur seine Richtung entsprechend korrigiert und der Spur folgt. Probiere diese Aufgabe zuerst mit Parcours 1a, dann mit Parcours 1b aus.

Programmiertipps:

- Es gibt mehrere Möglichkeiten, die Richtung zu korrigieren. Du kannst einen Motor anhalten und den anderen weiterdrehen lassen, oder einen Motor entgegen der Fahrtrichtung drehen lassen. Probiere aus, welche Methode sich besser eignet.
- Hier unser Lösungsvorschlag:

Na also. Jetzt kann der Roboter auf einer optischen „Schiene“ fahren. Der Nachteil ist nur, dass man ihn erst auf die Schiene setzen muss. Das wollen wir ändern. Der Roboter soll sich seine Spur jetzt selbstständig suchen.



Aufgabe 3: Die Spur finden und ihr anschließend folgen (Level 2)

- Schreibe ein Unterprogramm „suchen“, mit dem der Roboter nach einer Spur sucht, wenn er beim Start des Programms keine findet. Dazu soll sich der Roboter zuerst 1-mal im Kreis drehen. Findet er dabei keine Spur, soll er ein kleines Stück geradeaus fahren. Sobald der Roboter eine Spur erkennt, soll er ihr folgen. Andernfalls soll die Suche wieder von vorne anfangen. Hat er sich 10-mal im Kreis gedreht ohne eine Spur zu finden, soll er stehen bleiben und 3-mal blinken.

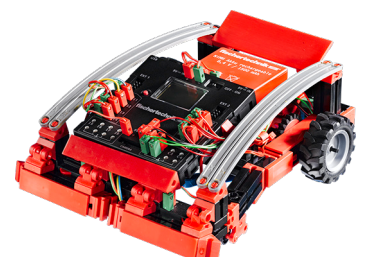
Programmiertipps:

- Falls Du noch nicht ganz auf der richtigen Spur sein solltest, hier unser Lösungsvorschlag:

Der Rasenmäher

Roboter können Rasen mähen? Aber klar. Man muss ihnen nur sagen, wie sie mit Hindernissen umgehen sollen, und wo der Rasen aufhört. Und schon kann man die Arbeit dem Roboter überlassen und den Nachmittag im Freibad verbringen.

Doch bevor es soweit ist, musst Du erst einmal den Rasenmäher-Roboter nach der Bauanleitung zusammenbauen. Anschließend solltest Du mit dem [Interface-Test](#) wieder prüfen, ob alles so funktioniert wie es soll.



Dann kann's an die Programmierung gehen. Der Rasen wird durch die weiße Fläche unseres Fußballstadions Parcours 1b symbolisiert, die von einer schwarzen Kante begrenzt wird. Diese Kante darf der Rasenmäher nicht überfahren (weil er sonst zum Beispiel auf die Tartanbahn gerät). Falls auf dem Rasen ein Hindernis auftaucht, soll der Roboter ausweichen, wenn er dagegen fährt. Außerdem soll das Mähwerk abgeschaltet werden, wenn ein Hindernis erkannt wird.



Aufgabe 1: Grenzen und Hindernisse erkennen und ausweichen (Level 2)

- Programmiere den Rasenmäher so, dass er von seiner Startposition aus (innerhalb der Begrenzung) gerade vorwärts fährt, bis er auf ein Hindernis trifft oder die Rasenbegrenzung (schwarze Linie) erreicht.
- Trifft der Rasenmäher auf ein Hindernis (Stoßstangen vorne), soll er sofort stehen bleiben, das Mähwerk stoppen, ein ganz kleines Stück rückwärts fahren, sich nach links drehen, und wieder vorwärts fahren und das Mähwerk wieder einschalten. Diese Funktion packst Du in das Unterprogramm „ausweichen“.
- Trifft der Rasenmäher auf die Rasenbegrenzung, soll er ebenso sofort stehenbleiben und das Unterprogramm „ausweichen“ starten.

Programmiertipps:

- Unser Lösungsvorschlag:

Abhängig von der Größe Deines „Rasens“ könnte es zu folgendem Problem kommen: Je nach dem, wie man die Dauer der „Wende“ einstellt, fährt der Roboter entweder nur am Rand entlang, oder bleibt evtl. immer im selben Bereich des „Rasens“. Darum soll der Roboter sich beim Ausweichen immer ein bisschen anders verhalten.



Aufgabe 2: Der Zufall (Level 3)

- Ändere das Programm des Rasenmähers so, dass der Rasenmäher jedes Mal beim Ausweichen einen anderen Winkel zum Drehen nimmt. Er soll sich also mal mehr, und mal weniger weit drehen. Außerdem soll er einem Hindernis, das er mit der rechten Stoßstange erkennt, nach links ausweichen und umgekehrt.

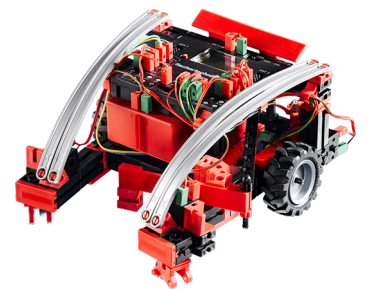
Programmiertipps:

- Um Variablen in ROBO Pro nutzen zu können, musst Du die Software auf „Level 3“ einstellen.
- Hierzu brauchst Du einen [Zufallsgenerator](#). Den kannst Du z. B. erstellen, indem Du den Wert einer Variable von einer [Zählschleife](#) immer wieder von „0“ an auf einen bestimmten Wert hochzählen lässt. Und im Unterprogramm „ausweichen“ setzt Du nach dem Befehl für das Drehen des Rasenmähers ein „Warten auf...“ Element, dass darauf wartet dass die Variable „0“ wird. Da die Variable wahrscheinlich jedes Mal, wenn sie abgefragt wird, einen anderen Wert hat, dauert es immer unterschiedlich lange, bis sie wieder bei „0“ ist. Und dadurch ergeben sich unterschiedliche Zeiten beim Drehen des Roboters.
- Zugegeben, das ist nicht ganz einfach. Falls Du trotz allem Nachdenken nicht auf die Lösung kommst, gibt's natürlich wieder einen Lösungsvorschlag von uns:

Der Fußball-Roboter

Hast Du schon einmal vom Robo-Cup gehört? Das ist die Fußball-Weltmeisterschaft für Roboter. Sie findet jedes Jahr in einem anderen Land statt. Es gibt verschiedene Ligen, in denen verschiedene Roboter-Arten zusammengefasst werden. Nähere Informationen findest Du zum Beispiel auf der Robo-Cup-Homepage <http://www.robocup.org>.

In der Bauanleitung findest Du einen Vorschlag zum Bau eines Fußball-Roboters. Er ist genau so beweglich wie unsere anderen Roboter, hat aber zusätzlich noch eine Lichtschranke um einen Ball zu erkennen, und einen „Abschussmechanismus“. Baue ihn schon mal zusammen, anschließend machen wir uns daran, ihn zu programmieren und mit ein paar Spieltricks zu „trainieren“. Und wie immer solltest Du die Funktionen des Modells zuerst mit dem [Interface-Test](#) prüfen, bevor Du mit dem Programmieren anfängst.



Als Ball kannst Du zum Beispiel einen Tennisball verwenden (nicht im Baukasten enthalten). Je nach dem, was für einen Ball Du verwendest, musst Du vielleicht den Abschussmechanismus etwas anpassen.



Aufgabe 1: „Da hat er den Ball erwischt, und er schießt ...“ (Level 2)

- Im ersten Schritt soll unser elektronischer Ballkünstler lernen, den Ball zu erkennen und darauf zu reagieren. Programme ihn so, dass er den Ball abschießt, sobald dieser von der [Lichtschranke](#) erkannt wird. Experimentiere ein bisschen mit der „Schussgeschwindigkeit“. Eventuell kann auch eine kurze Pause zwischen „erkennen“ und „abschießen“ zu einer Verbesserung führen.
- Dann soll er „Elfmeterschießen“ lernen. Lege einen Ball auf den Elfmeterpunkt des Parcours 1a. Setze den Roboter an den Anfang der schwarzen Linie. Er soll nun an der Linie entlang Anlauf nehmen und sobald er den Ball erkannt hat diesen auf das Tor schießen. Am Ende der Linie soll er anhalten und wenden.

Programmiertipps:

- Wie beim [Händetrockner](#) solltest Du bei der [Lichtschranke](#) nach dem Einschalten der [Linsenlampe](#) eine Sekunde warten, bevor Du den [Fototransistor](#) abfragst.
- Als Trainer hat man’s nicht leicht. Falls Dein Roboter-Spieler nicht auf Dich hört, kannst Du ihn mit unserem Programmvorschlag vielleicht überzeugen:

Da ein richtiger Ballkünstler aber noch mehr können muss, als nur Elfmeterschießen, wollen wir die Fähigkeiten unseres Fußball-Roboters noch ein bisschen erweitern.



Aufgabe 2: Wer suchet, der findet. (Level 2)

- Der Fußball-Roboter soll nun innerhalb des Stadions (Parcours 1b) umherfahren und dabei nicht über die Begrenzungslinie hinausfahren. Findet er den Ball, soll er ihn abschießen, natürlich möglichst ins Tor.

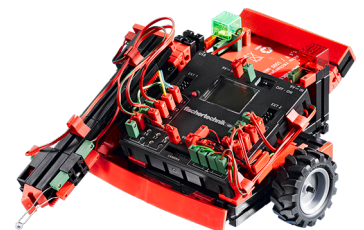
Programmiertipps:

- Die meisten Funktionen für dieses Programm hast Du beim [Rasenmäher](#) schon programmiert. Du kannst also das Rasenmäher-Programm verwenden, unter einem anderen Namen abspeichern und es mit den Funktionen des Fußball-Roboters erweitern.
- Wahrscheinlich wirst Du ihn nicht brauchen, aber trotzdem gibt es auch hier wieder einen Lösungsvorschlag:

Der Messroboter

Ist es unter Deinem Bett wärmer als unter dem Schreibtisch? Und wie heiß ist eine Kerzenflamme? Und kann man mit einem Eiswürfel sein Zimmer kühlen? Zu solchen (und anderen) Fragen kannst Du mit Hilfe des Messroboters Antworten finden. Er ist mit einem [Temperatursensor \(NTC\)](#) ausgestattet und kann an verschiedenen Stellen die Temperatur messen und anzeigen. Zusätzlich hat der Messroboter auch einen [Spursensor](#), so dass Du ihm seine Strecke durch eine schwarze Linie vorgeben kannst.

Baue zunächst den Messroboter auf wie in der Bauanleitung beschrieben und prüfe seine Funktionen mit dem [Interface-Test](#).



Aufgabe 1: Steuerung und Temperaturmessung (Level 3)

- Nimm das Programm des [Spurensuchers](#) und erweitere es um ein Unterprogramm zum Steuern des Messarms. Der Roboter soll auf der Spur des Parcours 1b fahren und in bestimmten Zeitabständen die Temperatur messen.
Die Temperatur soll auf dem Display des [ROBO TX Controllers](#) angezeigt werden.

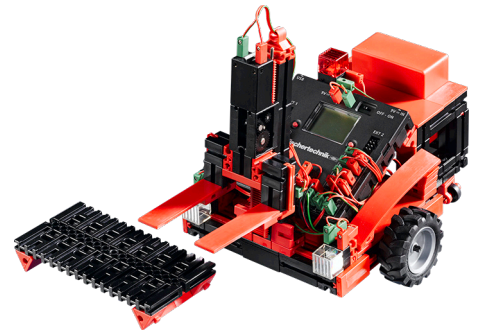
Programmiertipps:

- Um den Widerstandswert des [Temperaturfühlers](#) anzuzeigen benutzt Du einen [Analogeingang](#), einen [Bedienfeldausgang](#) und ein [„Anzeigen“-Bedienelement](#).
- Am Analogeingang wird der Widerstandswert des [Temperaturfühlers](#) angezeigt, nicht die Temperatur. Um diesen Wert in eine Temperaturanzeige umzurechnen, kannst Du ein Unterprogramm verwenden.
- Unseren kompletten Lösungsvorschlag für diese Aufgabe findest Du hier:

Der Gabelstapler

Ein Bereich, in dem Roboter in der Industrie sehr gerne eingesetzt werden, ist die Logistik. Also überall da, wo Dinge bewegt werden müssen, von „A“ nach „B“, wie man so schön sagt.

Solche Transportaufgaben kannst Du mit dem Gabelstapler-Modell wunderbar nachspielen. Baue das Modell auf, so wie es in der Bauanleitung beschrieben ist. Prüfe anschließend mit dem [Interface-Test](#), ob die Bauteile alle korrekt funktionieren.



Aufgabe 1: Rauf und runter (Level 3)

- Schreibe für die Funktionen „anheben“ und „ablassen“ jeweils ein Unterprogramm.
- Weil während der Fahrt die Gabel nicht zu weit oben oder unten sein darf, brauchst Du noch ein Unterprogramm „fahrposition“.

Programmiertipps:

- Beim „anheben“ soll die Gabel so weit nach oben gefahren werden (Motor M3 dreht nach links), bis der obere Endschalter I4 betätigt wird.
- Beim „absenken“ soll die Gabel so weit nach unten gefahren werden (Motor M3 dreht nach rechts), bis das Getriebe am unteren Endschalter I3 vorbei gefahren ist und der Schalter wieder offen ist. Hierzu verwendest Du ein [„Warten-Auf“-Element](#) und stellst es auf „1 -> 0 (fallend)“.
- Für die „fahrposition“ soll die Gabel so weit nach oben fahren, bis sich die Gabel knapp oberhalb des unteren Endschalters befindet.
- Wie immer gibt's von uns einen Vorschlag für die Lösung der Aufgabe:

Nachdem die Mechanik nun ganz gut funktioniert, wollen wir sie auch benutzen. Schließlich soll ein Gabelstapler ja Dinge transportieren, und nicht nur seine Gabel bewegen können.



Aufgabe 2: Von „A“ nach „B“ (Level 3)

- Im Baukasten gibt es einen Parcours 2a für den Gabelstapler. Er soll von seiner Startposition aus losfahren, eine Palette, die auf dem Feld „A“ steht, aufnehmen, auf der Spur zu Feld „B“ transportieren und dort abladen.

Programmiertipps:

- Programmiere den Gabelstapler so, dass er immer wenn die Spur zu Ende ist, die Palette auf- oder ablädt. Verwende für diese beiden Vorgänge jeweils ein Unterprogramm.
- Lasse den Gabelstapler so langsam fahren, dass er auf keinen Fall unbeabsichtigt die Spur verliert. Sonst meint er, er sei am Ende angekommen und müsse die Palette auf- oder abladen.
- Unser Lösungsvorschlag:



Aufgabe 3: Abzweigungen erkennen

Jetzt wird es spannend. Der Gabelstapler soll eine Abzweigung erkennen. Verwende dazu Parcours 2b.

- Der Gabelstapler startet bei seiner Startposition. Er soll zum Feld „A“ fahren, dort die Palette abholen und zum Feld „B“ transportieren. Anschließend soll er zurück zu seiner Startposition fahren. Während er der Spur folgt, sollen seine Scheinwerfer leuchten. Dreht er sich oder fährt er rückwärts, soll die rote Lampe leuchten.

Programmiertipps:

- Die Unterbrechung der Spur kann der Gabelstapler als Abzweigung erkennen. Erreicht er diese Stelle kannst du entscheiden, wie er weiterfahren soll: Er kann geradeaus fahren bis er die Spur findet, links abbiegen und dabei die Spur suchen oder rechts abbiegen bis er die Spur wieder findet. Erstelle für diese 3 Fälle jeweils ein Unterprogramm.
- Um beim Abbiegen nach links oder rechts die Spur wieder zu finden, muss der Roboter zuerst ein kleines Stück geradeaus fahren. Sonst dreht er sich an der Spur vorbei.
- Falls Du die Spur beim Programmieren verloren haben solltest, schau doch mal in unseren Vorschlag:



Aufgabe 4: Transportieren ohne Ende

- Erweitere das Programm aus Aufgabe 3 so, dass der Gabelstapler nach der Rückkehr zu seiner Startposition eine kurze Pause einlegt und dann die Palette vom Feld „B“ abholt, nach „A“ zurückbringt und dann wieder zum Start fährt. Den gesamten Ablauf soll der Stapler jetzt wie am Fließband ständig wiederholen.

Programmiertipps:

- Erzeuge eine Endlosschleife indem du vom letzten Programmelement eine Linie zum Anfang des Programms zurückführst.
- Auch wenn diese Aufgabe sicherlich kein Problem für Dich darstellt, hier ist unser Lösungsvorschlag:

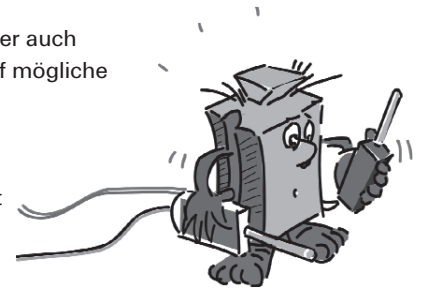


Wenn's nicht gleich funktioniert ...

... hat das in den meisten Fällen einen einfachen Grund. Nur ist der nicht immer auch einfach zu finden. Darum wollen wir Dir an dieser Stelle ein paar Hinweise auf mögliche Fehlerquellen geben.

Interface-Test

Auch hier nochmal der Hinweis: Prüfe die Funktion der einzelnen Bauteile mit Hilfe des [Interface-Tests](#) in [ROBO Pro](#).



Kabel und Verkabelung

Wenn ein elektrisches Bauteil überhaupt nicht funktioniert, prüfe einmal das Kabel, mit dem Du es an den [ROBO TX Controller](#) angeschlossen hast. Verbinde dazu mit dem Kabel den [Akku](#) und eine [Gühlampe](#). Leuchtet die Glühlampe auf, sollte das Kabel in Ordnung sein.

Auch falsch montierte Stecker (z. B. ein grüner Stecker auf einem roten Kabel) können eine Fehlerquelle sein.

Prüfe auch, ob „+“ und „-“ richtig angeschlossen sind. Vergleiche dazu Dein Modell mit den Bildern in der Bauanleitung.

Wackelkontakt

Ein Bauteil, das abwechselnd mal funktioniert und dann wieder nicht mehr funktioniert, hat vermutlich irgendwo in seiner Verkabelung einen Wackelkontakt.

Die häufigsten Ursachen dafür sind:

- **Lose sitzende Stecker**

Wenn die Stecker der Kabel zu lose, also wackelig in den Buchsen sitzen, haben sie keinen ausreichenden Kontakt. In diesem Fall kannst Du mit dem Schraubendreher die Kontaktfedern vorne an den betreffenden Steckern **vorsichtig** auseinanderbiegen. Nur ganz leicht, so dass die Stecker wieder fest in den Buchsen sitzen, wenn Du sie einsteckst.

- **Schlechter Kontakt zwischen Kabel und Stecker**

Überprüfe auch den Kontakt zwischen den [abisolierten](#) Kabel-Enden im Stecker und dem Stecker selbst. Möglicherweise genügt es auch, die Schrauben im Stecker ein bisschen fester anzuziehen.

Kurzschlüsse

Einen Kurzschluss hast Du dann, wenn sich ein Plus- und ein Minusanschluss berühren. Sowohl der [Akku](#), als auch der [ROBO TX Controller](#) haben eine Sicherung eingebaut, so dass sie bei einem Kurzschluss nicht beschädigt werden. Sie schalten die Stromversorgung einfach für eine Weile ab. Dein Modell funktioniert dann jedoch natürlich auch nicht mehr.

Die Ursache für einen Kurzschluss kann entweder ein Fehler in der Verkabelung sein, oder auch zu lose angezogene Schrauben in den Steckern. Die können sich berühren wenn die Stecker entsprechend eingesteckt sind, und so einen Kurzschluss erzeugen. Du solltest die Schrauben darum immer komplett eindrehen und die Stecker so einstecken, dass sich die Schrauben nicht berühren können.

Stromversorgung

Kleine Aussetzer oder langsam laufende Motoren deuten meistens auf einen leeren [Akku](#) hin. In diesem Fall solltest Du den Akku mit dem beiliegenden Ladegerät aufladen. Wenn die rote LED am Ladegerät aufhört zu blinken und durchgehend leuchtet, ist der Akku vollständig geladen.

Fehler im Programm

Auch wenn es niemand gerne zugibt: Jeder macht Fehler. Und vor allem bei komplexeren Programmen schleicht sich so ein Fehler ganz schnell ein.

Wenn Du also am Modell selber alles überprüft, und alle Fehler beseitigt hast, und Dein Modell trotzdem nicht das macht was Du willst, dann solltest Du auch noch Dein Programm überprüfen. Gehe es Stück für Stück durch und kontrolliere, ob Du den Fehler findest.

Im Online-Modus, also wenn der [ROBO TX Controller](#) mit dem PC verbunden ist, kannst Du das Programm auch am Bildschirm verfolgen, während es läuft. Das jeweils aktive Programmelement wird hervorgehoben, so dass Du immer sehen kannst, an welcher Stelle das Programm gerade ist und wo der Fehler auftritt.

Die letzten Joker

Wenn Du trotz allem den Fehler nicht gefunden hast, gibt es noch zwei Möglichkeiten, wie Du Hilfe bekommen kannst:

- **E-Mail-Joker**

Du kannst uns bei fischertechnik eine E-Mail schicken, und uns Dein Problem schildern. Die E-Mail-Adresse ist info@fischertechnik.de.

- **Publikums-Joker**

Du kannst uns auch im Internet auf <http://www.fischertechnik.de> besuchen. Dort gibt es unter anderem ein Forum, wo Dir bestimmt geholfen werden kann. Außerdem kannst Du kostenlos Mitglied im fischertechnik Fanclub werden.

Und wie geht's jetzt weiter?

War's das jetzt schon? Nein, natürlich nicht. Die Experimente und Modelle, die Du in diesem Heft kennen gelernt und ausprobiert hast, sollen nur der Anfang sein. Sozusagen Deine ersten „Gehversuche“ in dem riesigen und spannenden Themenbereich „Computing“.

Was wir Dir hier gezeigt haben ist nur ein winziger Teil der Möglichkeiten, die Du mit dem [ROBO TX Controller](#) und den fischertechnik Bauteilen hast. Und ab jetzt bist Du selber gefragt. Du darfst Deiner Fantasie freien Lauf lassen und einfach mal drauf los bauen.

Wenn Du noch keine Idee für ein komplett eigenes Modell hast, dann schau Dir doch einfach mal die Modelle aus diesem Heft an. Vielleicht fällt Dir ja etwas ein, was Du an einem Modell anders machen würdest. Oder Du veränderst die Funktion eines Modells.

Zum Beispiel könnte man beim [Gabelstapler](#) an Stelle der Gabel auch einen Stift befestigen, der dann hoch und runter gelassen werden kann und auf einem großen Papier malt, während der Roboter darüber fährt. Du kannst dann Figuren nicht nur abfahren, sondern sogar aufzeichnen. Und aus dem Gabelstapler ist eine Zeichenmaschine geworden.

Und wenn Freunde von Dir auch einen [ROBO TX Controller](#) haben, dann wird es noch interessanter. Denn über die Bluetooth-Schnittstelle kann nicht nur Dein PC mit dem [ROBO TX Controller](#) kommunizieren, sondern auch mehrere Controller untereinander. So könntet Ihr dann zum Beispiel 2 Roboter programmieren, die aufeinander reagieren. Oder miteinander tanzen. Im Kapitel 7 der [ROBO Pro Onlinehilfe](#) findest Du einige interessante Infos zu diesem Thema.

Hat Dein Rechner eine Bluetooth-Schnittstelle? Wenn ja, dann kannst Du den [ROBO TX Controller](#) anstatt über ein USB-Kabel auch über Bluetooth mit dem Rechner verbinden. Wenn nicht, kannst Du Dir auch einen USB-Bluetooth-Stick kaufen und darüber den [ROBO TX Controller](#) kabellos mit deinem PC verbinden. Wie das geht kannst du in der Bedienungsanleitung des [ROBO TX Controllers](#) und auf <http://www.fischertechnik.de> nachlesen.

Also, worauf wartest Du? Los geht's. Erfinde und experimentiere. Und lass Dich nicht von kleinen Rückschlägen aufhalten. Zum Experimentieren gehört vor allem auch Geduld und Ausdauer. Die Belohnung ist danach ein funktionierendes Modell.

Beim Ausprobieren Deiner eigenen Ideen wünschen wir Dir viel Spaß.

