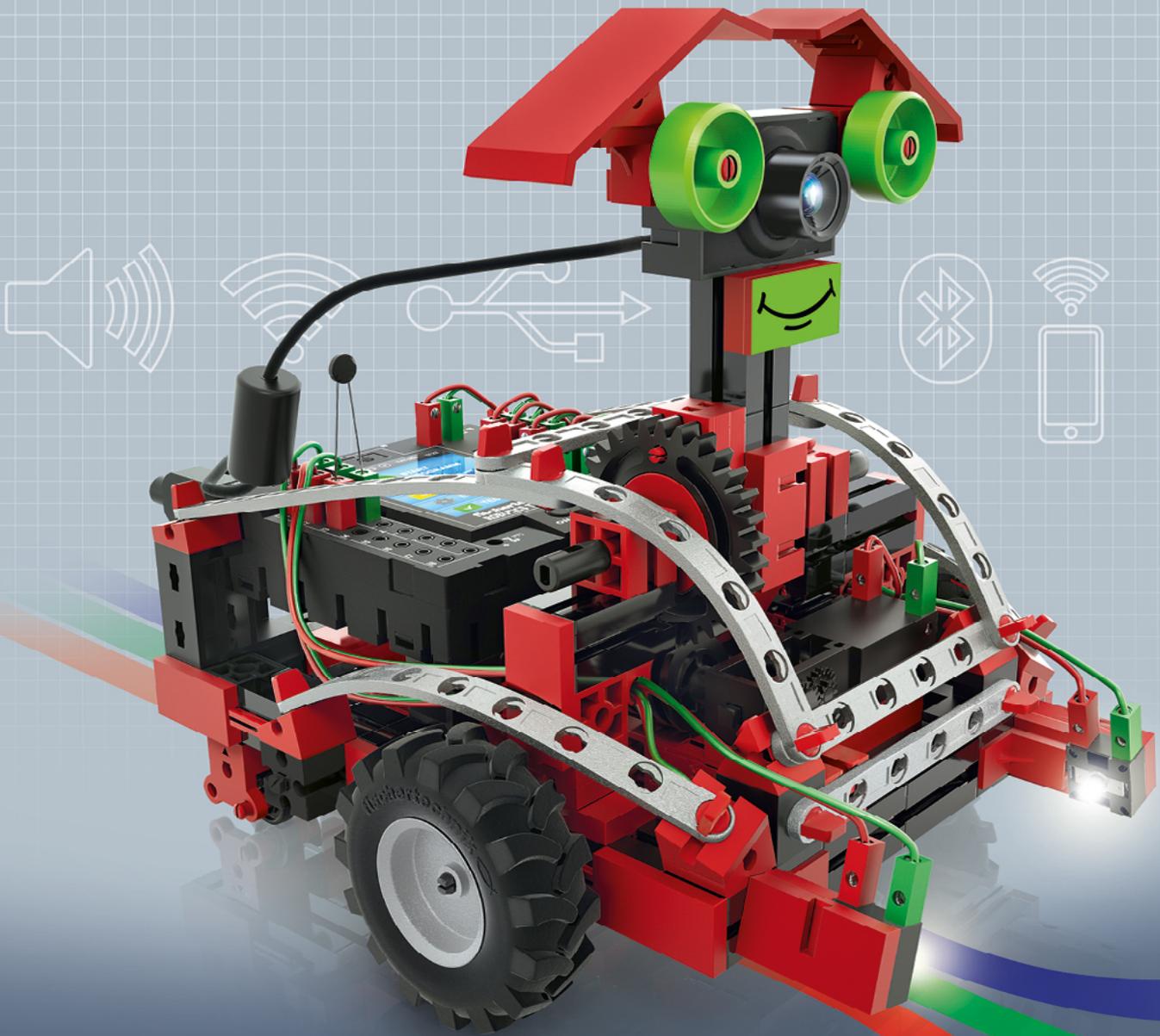




fischertechnik 

ROBOTICS



TXT *Discovery Set*

14 MODELS
MODELS

Willkommen in der fischertechnik ROBOTICS Welt	4
Ein paar allgemeine Infos	5
Elektrizität	5
Über dieses Begleitheft	5
Roboter, der künstliche Mensch?	6
ROBOTICS, (fast) alles automatisch	7
Bauteilerklärungen	8
Encoder-Motoren	8
XS-Motor	9
LEDs	9
Linsenlampe	9
Fototransistor	10
Taster	10
Wärmesensor (NTC)	11
Kameramodul	11
ROBOTICS TXT Controller	12
Ein paar Tipps	13
Erste Schritte	14
Starter-Modelle	15
Fußgängerampel	15
Händetrockner	16
Temperaturregelung	18
Schranke	19
„Kameramann“	20
Schwenkbare Kamera	22

Mobile Roboter – die nächste Herausforderung	23
Fahrroboter	23
Hinderniserkenner	27
Hinderniserkenner mit Kamera	28
Spurensucher	30
Erkundungsroboter	33
Fußball-Roboter mit Bewegungssteuerung	35
Fußballroboter	38
Fehlersuche	43
Interface-Test	43
Kabel und Verkabelung	43
Wackelkontakt	43
Kurzschlüsse	44
Stromversorgung	44
Fehler im Programm	44
Kamerafunktion	44
Die letzten Joker	45
Und wie geht's jetzt weiter?	46

Willkommen in der fischertechnik ROBOTICS Welt

Hallo!

Wir freuen uns, dass du dich für den Baukasten „ROBOTICS TXT Discovery Set“ von fischertechnik entschieden hast. 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 ROBOTICS TXT 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 ROBOTICS TXT Discovery Set.

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 ROBOTICS TXT Discovery Sets 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 ROBOTICS TXT 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 Baukasten „PROFI Electronics“ genau so viel Spaß haben wie mit dem ROBOTICS TXT Discovery Set.



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 so was 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 Hilfe). 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.

- **Das ROBO Pro Symbol**

Das findest du immer in der Nähe von Aufgaben. Das macht Sinn, denn sobald du darauf klickst wird ein passendes Beispielprogramm mit einer möglichen Lösung geöffnet.

Alle Beispielprogramme findest du auch unter **C:\Programme (x86)\ROBOPro\Beispielprogramme\ROBOTICS TXT Discovery Set**.

Klickmich.rpp

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.“



ROBOTICS, (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 „ROBOTICS“.

Mit dem „ROBOTICS TXT Discovery Set“ 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 4.0 (oder höher) erstellen und anschließend per USB-, Bluetooth- oder WLAN Verbindung auf den [ROBOTICS TXT 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

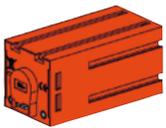
Zunächst einmal findest du zahlreiche fischertechnik-Bausteine, außerdem Motoren, Lampen und Sensoren, sowie eine farbige Bauanleitung zum Bau der verschiedenen Modelle.

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 irgendeiner 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 drei Impulse. Und weil die Encoder-Motoren zusätzlich noch ein Getriebe mit einem Übersetzungsverhältnis von 21:1

(sprich „21 zu 1“) haben, entspricht eine Umdrehung der Welle, die aus dem Getriebe kommt, 63 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.

Das Getriebe, das im Baukasten dabei ist, passt genau an den XS-Motor.

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

LEDs



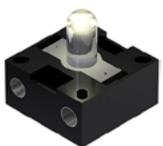
Zwei LEDs sind im Baukasten enthalten. Sie können sehr vielfältig verwendet werden. Zum Beispiel als Signallichter bei einer Ampel, als Blinklicht an einem Roboter oder zur besseren Ausleuchtung des Bildes, das die Kamera liefert, die ebenfalls im Baukasten enthalten ist.

Die LEDs sind für eine Spannung von 9 Volt ausgelegt und verbrauchen ca. 0,01 Ampere Strom.

Achtung:

Beim Anschluss der LED an die Stromversorgung musst du auf die richtige Polung achten. Der Plus-Pol muss bei der roten Markierung an die LED angeschlossen werden.

Linsenlampe



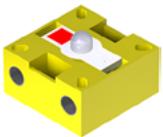
In diese Glühlampe ist eine Linse eingearbeitet, die das Licht bündelt. Sie sieht der LED sehr ähnlich. Du musst aufpassen, dass du sie nicht verwechselst. Bei der Linsenlampe ist die Polung egal, deshalb findest du auf dem Sockel keine Markierung. Du benötigst die Linsenlampe zum Bauen einer [Lichtschranke](#) zusammen mit dem [Fototransistor](#).

Die Linsenlampe ist 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 Linse. Bei großer Helligkeit, also wenn der Transistor von der Linse 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 bei der roten Markierung an den Fototransistor angeschlossen werden.

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.

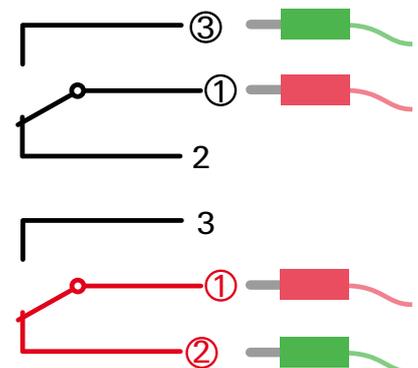
Taster nicht gedrückt: es fließt kein Strom

Als „Öffner“:

Kontakte 1 und 2 werden angeschlossen.

[Taster gedrückt](#): Es fließt kein Strom.

Taster nicht gedrückt: Es fließt Strom.



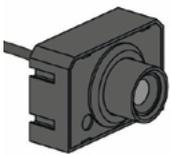
Wärmesensor (NTC)



Bei diesem Bauteil handelt es sich um einen Wärmesensor, mit dem man Temperaturen messen kann. Bei 20 °C beträgt sein **elektrischer Widerstand** 1,5 kΩ (kilo-Ohm). NTC bedeutet Negativer Temperatur Coefficient. 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 [ROBOTICS TXT 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.

Kameramodul



Ein besonders vielseitiger Sensor ist das Kameramodul. Die Bildauflösung beträgt 1 Megapixel (das bedeutet, dass jedes Bild aus einer Million Bildpunkten besteht). Du schließt die Kamera über die große USB Buchse (USB1) an den [ROBOTICS TXT Controller](#) an. Die Bilder, die die Kamera aufnimmt, kannst du an den PC übertragen und am Bildschirm anzeigen. So siehst du immer genau, was dein Roboter gerade sieht. Außerdem kann der ROBOTICS TXT Controller die Bilder verarbeiten und dadurch Bewegungen, Farben oder Spuren erkennen, so dass du damit deine Robotermodelle steuern kannst. Es besteht aber auch die Möglichkeit die Kamera direkt an eine USB-Schnittstelle deines PCs anzuschließen und die Bilder dann in der Software ROBO Pro zu verarbeiten. Auch diese Möglichkeit werden wir bei einigen Modellen verwenden.

Du kannst das Bild der Kamera scharf stellen indem du am Kameraobjektiv drehst.

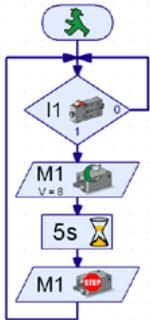
Software ROBO Pro 4.x

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

„Grafische Programmieroberfläche“ bedeutet, dass du die Programme nicht Zeile für Zeile von Hand „schreiben“ musst, sondern mit Hilfe



TXT Discovery Set

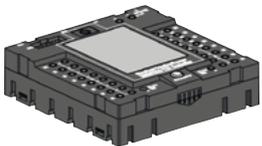


von grafischen Symbolen einfach bildlich zusammenstellen kannst. Ein Beispiel für so ein Programm siehst du links abgebildet.

Wie man so ein Programm genau erstellt, ist in dem Kapitel „Erste Schritte“ ausführlich beschrieben. Auch die [ROBO Pro Hilfe](#) zeigt dir in den Kapiteln 3 und 4 wie das funktioniert.

Die Software wurde bereits mit diesem Begleitmaterial auf deinem PC installiert.

ROBOTICS TXT Controller



Der ROBOTICS TXT Controller ist das Herzstück dieses ROBOTICS-Baukastens. Er steuert die [Aktoren](#), und wertet die Informationen der [Sensoren](#) aus.

Für diese Aufgabe verfügt der ROBOTICS TXT 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 ROBOTICS TXT Controller beschrieben.

Über das farbige Touch Display kannst du den ROBOTICS TXT Controller komfortabel bedienen. An die USB-Host-Buchse (USB-1) kannst du das im Baukasten enthaltene Kameramodul anschließen. Ein besonderer Leckerbissen ist die integrierte Bluetooth- und WLAN Schnittstelle. Über sie kannst du ohne Kabel Deinen PC mit dem ROBOTICS TXT Controller verbinden, oder auch mehrere Controller 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 viele der Bauteile vom ROBOTICS TXT Discovery Set mit Strom funktionieren, brauchst du natürlich auch eine Stromversorgung.

Am besten geeignet ist hierfür das fischertechnik Accu Set. Es ist im Baukasten 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 [ROBOTICS TXT Controller](#) angeschlossenen Teile mit Hilfe des [Interface-Test](#) von ROBO Pro testen. Wie das genau funktioniert ist in der [ROBO Pro Hilfe](#) in Kapitel 2.4 erklärt.

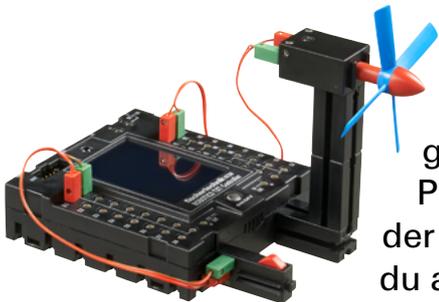
Erste Schritte

So. Nach all den Vorbereitungen und Informationen kannst du jetzt endlich loslegen. In diesem Kapitel wollen wir dir zeigen, wie du



- das erste einfache Modell, nämlich einen Ventilator, baust und an den [ROBOTICS TXT Controller](#) anschließt
- den ROBOTICS TXT Controller mit der Stromversorgung und dem PC verbindest
- die [Software ROBO Pro](#) aufrufst und das Modell testest
- ein ROBO Pro Programm für das Modell lädst und startest
- ein erstes einfaches Programm in ROBO Pro selbst erstellst und startest

[Klicke hier](#), um dir die ersten Schritte leicht verständlich zeigen zu lassen.



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

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 Hilfe](#) durchgelesen hast, kannst du jetzt schon einige Modelle des Baukastens programmieren. Deshalb wollen wir auch sofort loslegen. In ROBO Pro gibt es verschiedene Schwierigkeitsstufen, Level genannt, die du in der Menüleiste einstellen kannst. Wir beginnen mit ganz einfachen Programmen im Level 1. Immer, wenn du ein Modell fertig gebaut und verkabelt hast, prüfe mit Hilfe des [Interface-Tests](#), ob am [ROBOTICS TXT Controller](#) alle Aus- und Eingänge richtig angeschlossen sind und die [Sensoren](#), [Motoren](#) und [Lampen](#) richtig funktionieren.



Fußgängerampel

Vor Eurem Haus wurde eine Fußgängerampel 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: (Level1)

Die Ampel soll zunächst auf Rot stehen. Wird der Taster I1 von einem Fußgänger gedrückt, soll die Ampel 3 Sekunden später auf Grün wechseln. Die Grünphase soll 10 Sekunden dauern, dann soll es wieder Rot werden.



Programmiertipps:

Die verschiedenen LEDs gehören zu folgenden Ausgängen des TXT Controllers

- Rot – M1
- Grün – M2

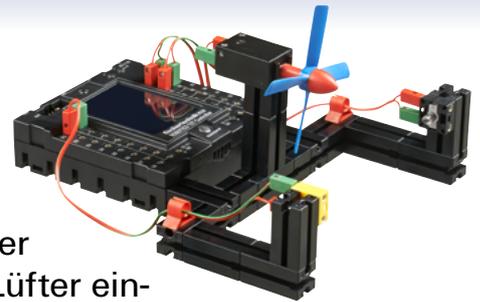
Schalte die Lampen so hintereinander ein und aus, dass der gewünschte Ablauf zustande kommt.

Das fertige Programm kannst du wieder durch Klicken auf das Bild rechts laden:

Fussgaengerampel.rpp



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: (Level 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:

Haendetrockner_1.rpp

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

Aufgabe 2: (Level 1)

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:

Haendetrockner_2.rpp

Temperaturregelung

Bei Euch zu Hause wurde eine neue Klimaanlage installiert. Natürlich hast du den Installateur sofort gefragt, wie die Temperaturregelung funktioniert. Er hat dir erklärt, dass ein Temperaturfühler ständig die aktuelle Temperatur misst. Sobald ein oberer Grenzwert überschritten wird, schaltet die Kühlung ein. Wird 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: (Level 1)

Die Heizung wird simuliert durch die Linse Lampe M2. Als „Kühlaggregat“ dient das Gebläse am Ausgang M1. Zur Temperaturmessung verwenden wir den NTC-Widerstand am Eingang I8.

Programmiere 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.



Programmiertipps:

Beachte: Der Widerstandswert des NTC-Widerstands sinkt mit steigender Temperatur. Der obere Temperaturgrenzwert ist also der kleinste Wert von I8. Bei diesem Grenzwert soll das Gebläse einschalten. Der untere Temperaturgrenzwert ist der größte Wert von I8. Bei diesem Grenzwert soll die Heizung einschalten.

Welchen Wert I8 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.

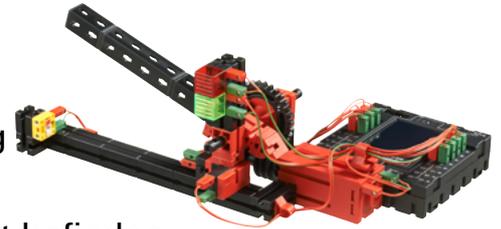


Durch einen Klick auf das Symbol rechts kannst du das fertige Programm öffnen.

Temperaturregelung.rpp

Schranke

Oft findet man eine solche Schranke an der Ein- oder Ausfahrt eines Parkplatzes. Baue zunächst das Modell wie in der Bauanleitung beschrieben auf.



Unsere Schranke könnte sich an der Ausfahrt befinden.

Aufgabe: (Level 2)

Nähert sich ein Fahrzeug der Schranke wird dadurch die Lichtschranke unterbrochen. Dann soll die Schranke sich öffnen, eine gewisse Zeit offen bleiben (z. B. 5 Sekunden) und sich dann wieder schließen, aber nur wenn die Lichtschranke nicht unterbrochen ist, denn sonst beschädigt die Schranke womöglich das Auto, das sich noch genau unter der Schranke befindet. Um dem Autofahrer zu zeigen, ob der Weg durch die Schranke frei ist, soll die Ampel grün leuchten, wenn die Schranke offen ist und Rot, wenn sie sich schließt.



Programmiertipps:

Verwende für die Vorgänge „Öffnen“ und „Schließen“ der Schranke jeweils ein Unterprogramm (siehe auch [ROBO Pro Hilfe](#) Kapitel 4).

In ROBO Pro stellst du dazu Level 2 ein.

Durch einen Klick auf des Symbol rechts kannst du das fertige Programm öffnen.

Schranke.rpp



„Kameramann“

Jetzt wollen wir uns zum ersten Mal in diesem Begleitheft mit dem spannenden Thema Kamera und Bildverarbeitung beschäftigen. Dazu baust du das Modell „Kameramann“ wie in der Bauanleitung beschrieben auf und verbindest die Kamera über einen USB Anschluss direkt mit dem PC. Die Kamera wird erkannt und der zugehörige Kameratreiber automatisch installiert.

Starte jetzt die Software ROBO Pro.



Aufgabe 1:

Stelle den Kameramann vor deinem PC auf und lasse dir in ROBO Pro anzeigen was die Kamera „sieht“.



Programmiertipps:

Dazu brauchst du nicht einmal ein ROBO Pro Programm erstellen. Öffne in ROBO Pro einfach ein neues Programm, gehe auf den Reiter „Kamera“, stelle dort den Kameraanschluss auf „PC“ und klicke auf „Kamera einschalten“.

Durch Drehen am Objektiv der Kamera kannst du das Bild scharf stellen.

Aufgabe 2: (Level 3)

Der Kameramann soll nun auf Bewegungen im Bild reagieren. Sobald jemand durchs Bild läuft soll am TXT Controller eine rote Lampe am Ausgang M1 dreimal blinken.



Programmiertipps:

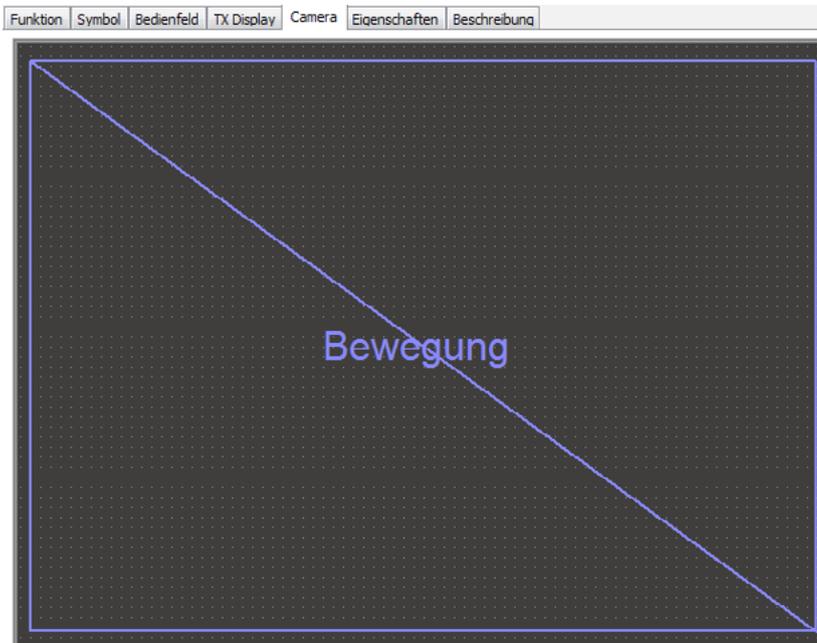
Schließe den TXT Controller über eine zweite USB-Schnittstelle an deinen PC an.

Schließe an den TXT Controller eine LED mit roter Abdeckung an den Ausgang M1 an.

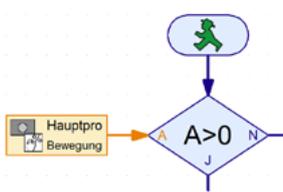
Stelle in ROBO Pro Level 3 ein. Welche Funktionen dort zusätzlich verfügbar sind, kannst du in der [ROBO Pro Hilfe](#) im Kapitel 5 nachlesen.



Verwende in ROBO Pro das Kamera-Sensorfeld „Bewegung“ und zeichne es im Kamerafeld als Rechteck in der gewünschten Größe.



Im Dialogfeld des Bewegungserkenners, das du mit einem rechten Mausklick auf das eingefügte Sensorfeld öffnen kannst, lässt sich einstellen, wie stark die Bewegung sein muss, damit der Sensor anspricht und wie groß die Bewegung im Verhältnis zum gesamten Sensorfeld sein muss, damit er reagiert. Am Besten versuchst du es mit den voreingestellten Werten und probierst später aus, was die Einstellungen bewirken.



Erstelle in ROBO Pro einen Ablauf, in dem du abfragst, ob das Sensorfeld eine Bewegung erkannt hat. Zur Abfrage des Sensorfeldes verwendest du das Kamera-Eingangelement aus dem Level 3 von ROBO Pro und fragst mit einer Verzweigung ab, ob eine Bewegung vorhanden ist:

Abfrage des Eingangs „Bewegung C“  (Änderungskontrast). Ist Wert >0? Wenn dies der Fall ist, soll die Lampe blinken.

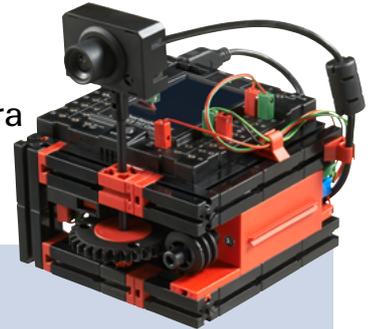
Weitere Hinweise zu den Kamera-Sensorfeldern, siehe [ROBO Pro Hilfe](#) Kapitel 11.

Das fertige Beispielprogramm kannst du hier öffnen.

Kameramann.rpp

Schwenkbare Kamera

Als Nächstes baust du dir eine Überwachungskamera für dein Zimmer. Baue zunächst das Modell wie in der Bauanleitung beschrieben auf.



Aufgabe: (Level 3)

Die Kamera soll sich, angetrieben vom [Encodermotor](#), immer ein kleines Stück drehen, und danach prüfen, ob sich im Raum etwas bewegt. Wenn dies der Fall ist, soll sie über den Lautsprecher des TXT Controllers einen Alarm ausgeben, wenn sich nichts bewegt, soll sie sich ein Stückchen weiter drehen und dann wieder nachsehen.

Ist das Ende des Schwenkbereichs erreicht, soll sie sich wieder in kleinen Schritten zu ihrem Ausgangspunkt zurück bewegen.



Programmiertipps:

Die Kamera soll zunächst zu Ihrer Ausgangsposition fahren (bis Taster I1 gedrückt).

Dann soll sie immer eine bestimmte Anzahl Impulse in die andere Richtung fahren. Probiere mit Hilfe des [Interface-Tests](#) einfach aus wie viele Impulse pro Drehbewegung sinnvoll sind.

Frage mit dem Kamera-Sensorelement „Bewegung“ ab, ob sich im Bild etwas verändert. Wenn ja, gib über den Lautsprecher des TXT Controllers einen Alarmton aus, siehe ([ROBO Pro Hilfe](#)).

Wiederhole diesen Vorgang so oft, bis der Schwenkbereich der Kamera zu Ende ist und fahre dann wieder in die andere Richtung, zurück zum Endschalter I1.

Hier findest du unsere Lösung für diese Aufgabe:

Schwenkbare_Kamera.rpp

Dieses Modell kannst du unabhängig vom PC im Download-Modus betreiben.

Die perfekte Überwachungskamera, um unbefugte Eindringlinge aus deinem Zimmer zu vertreiben.

Mobile Roboter – die nächste Herausforderung

Fahrroboter



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.

Lass dir Zeit beim Zusammenbauen. Schau dir die Zeichnungen in der Bauanleitung genau an, auch die Verkabelung. Wenn du die Bauteile anders mit dem [ROBOTICS TXT Controller](#) verbindest als in der Bauanleitung beschrieben, so wird sich der Roboter möglicherweise anders verhalten als du es erwartest.

Prüfe nach dem Zusammenbauen alle an den ROBOTICS TXT Controller angeschlossenen Bauteile über den [Interface-Test](#) der ROBO Pro Software. Wenn du die Motoren nach links drehen lässt, muss der Roboter vorwärts fahren.

Fahrtrichtung	Drehrichtung Motor 1	Drehrichtung Motor 2
Vorwärts	Links	Links
Rückwärts	Rechts	Rechts
Links	Links	Rechts
Rechts	Rechts	Links
Stopp	Stopp	Stopp

Aufgabe 1: Einfach mal gerade aus

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:

Fahrroboter_1.rpp

Die Lenkung

Auch wenn es Spaß macht dem Roboter beim Geradeausfahren zuzuschauen, 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

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 dieselbe Richtung).

Finde dann 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 kannst du wieder durch Klicken auf das Symbol rechts öffnen:

Fahrroboter_2.rpp

Aufgabe 3: Eine Figur abfahren

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:

Fahrroboter_3.rpp

Immer dasselbe, und doch nicht gleich?

Wie du bestimmt gemerkt hast, ist die Wiederholgenauigkeit des Roboters noch verbesserungsfähig. Auch wenn er mehrmals genau dieselbe Aufgabe ausführt, ist das Ergebnis nicht immer das Gleiche. Das hat verschiedene Gründe. Einer davon ist, dass 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 derselben 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 Hilfe](#) in Kapitel 12.6 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:

Fahrroboter_1_sync.rpp

Fahrroboter_2_sync.rpp

Fahrroboter_3_sync.rpp

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.

Hinderniserkenner

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 ermöglichen auf Hindernisse zu reagieren, und zwar mit Hilfe einer Stoßstange.

Baue zunächst das Modell wie in der Bauanleitung beschrieben auf.



Aufgabe:

Der Hinderniserkenner soll geradeaus fahren. Sobald er mit seiner linken Stoßstange auf ein Hindernis trifft, soll er anhalten, ein kleines Stück zurückfahren, sich etwas nach rechts drehen und dann wieder geradeaus weiterfahren. Trifft er mit seiner rechten Stoßstange auf ein Hindernis, soll er entsprechend nach links ausweichen.



Programmiertipps

Stoßstange links: Taster I6; Stoßstange rechts: Taster I5

Verwende für die verschiedenen Aktionen: vorwärts / rückwärts fahren, sowie nach links / nach rechts ausweichen jeweils ein Unterprogramm.

Achte darauf, dass der Roboter nicht die gleiche Anzahl Impulse nach links und nach rechts ausweicht, sonst kann es sein, dass er aus einer Ecke, in die er hineingerät, nicht mehr herauskommt. Ist die Impulszahl dagegen unterschiedlich, arbeitet er sich auf jeden Fall wieder aus einer Ecke heraus.

Das fertige Programm findest du hier:

Hinderniserkenner.rpp



Hinderniserkennung mit Kamera

Dein mobiler Roboter soll jetzt eine Kamera bekommen, damit er sehen kann, wo er hinfährt und du ihn dann auch fernsteuern kannst. Dazu nimmst du einfach den Hinderniserkennung und baust ihm, wie in der Bauanleitung beschrieben, eine Kamera ein. Die Kamera schließt du an die **USB-1** Schnittstelle des TXT Controllers an.



Aufgabe 1:

Zunächst soll sich der Roboter genauso verhalten wie der Hinderniserkennung ohne Kamera. Zusätzlich soll er anhalten und umdrehen, wenn du ihm eine rote Karte vor die Nase hältst.



Verwende für die Farberkennung das Kamera-Sensorfeld „Farbe“ und ziehe es so groß, dass seine Fläche einen großen Teil des Kamerabildschirms einnimmt.

Programmiere den Roboter nun so, dass er geradeaus fährt, wenn er keine rote Tafel sieht und kein Hindernis vorhanden ist. Die Hinderniserkennung kannst du aus dem Programm „Hinderniserkennung“ übernehmen.

Für die Abfrage der Farbe verwendest du am Besten ein eigenes Unterprogramm. Du fragst über den Kameraeingang ab, ob der Rotwert größer als grün und größer als blau ist und ob die Farbe heller als schwarz ist.

Als Farbtafel verwendest du das rote Kärtchen, das du aus der großen Schnittvorlage ausschneiden kannst, die im Baukasten enthalten ist.

Im Kamerafenster von ROBO Pro kannst du unter „Sensorwerte“ ablesen, welche Farbwerte das rote Kärtchen hat.

Unsere Lösung findest du wieder hier:

Hinweis:

Hinderniserkennung_Kamera_1.rpp

Vermeide direkte Sonneneinstrahlung. Es kann vorkommen, dass der Roboter Sonnenlicht als rote Tafel erkennt. Vor allem helles Sonnenlicht, das einen hohen Rotanteil besitzt.

Aufgabe 2: Fernsteuerung des Roboters

Hinweis: Um diese Aufgabe bearbeiten zu können benötigst du eine WLAN-Verbindung zwischen dem TXT Controller und deinem PC. Wie du diese herstellen kannst ist in der Bedienungsanleitung zum TXT Controller beschrieben.



Tipps:

Nachdem du die WLAN-Verbindung erfolgreich hergestellt hast, öffnest du über das Symbol das passende Programm.

Du wechselst in das Bedienfeld und findest dort die Oberfläche zum Fernsteuern des Modells.

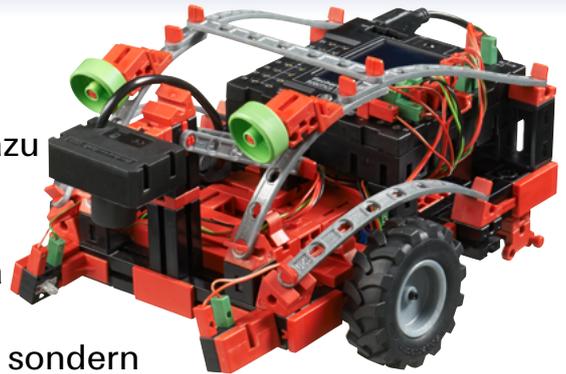
Hinderniserkenner_Kamera_2.rpp

Starte das Programm im Onlinemodus. Nun kannst du den Roboter über die verschiedenen Buttons fernsteuern und siehst außerdem noch wohin er fährt. Übersiehst du einmal ein Hindernis, wird dieses von den Stoßstangen erkannt und das Modell fährt automatisch ein kleines Stück zurück.

Viel Spaß beim Erkunden der Umgebung!

Spurensucher

Bei diesem Modell wird die Kamera dazu verwendet, den Roboter an einer Linie entlang fahren zu lassen. Das Geniale an der Spurverfolgung mit der Kamera ist, dass die Kamera nicht nur erkennt, ob eine Spur vorhanden ist oder nicht, sondern deren genaue Position im Kamerabild erkennen und ausgeben kann. Darauf kann der Roboter reagieren und entweder geradeaus fahren, wenn sich die Spur genau in der Mitte des Bildes befindet oder nach links oder rechts korrigieren, wenn er die Spur außerhalb der Bildmitte „sieht“.



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

Aber eins nach dem anderen. Als Erstes musst du das Modell Spurensucher aufbauen. Wie das geht, steht natürlich in der Bauanleitung.

Nachdem du das Modell fertig gebaut hast, solltest du mit dem [Interface-Test](#) überprüfen, ob alle Bauteile richtig am [ROBOTICS TXT Controller](#) angeschlossen sind und funktionieren.

Aufgabe 1: Eine Spur erkennen

Programmiere den Roboter so, dass er einer schwarzen Spur folgt, auf die er gesetzt wird. Verliert er die Spur oder ist sie zu Ende, soll der Roboter stehen bleiben. Verwende für diese Aufgabe den Parcours aus dem Baukasten.



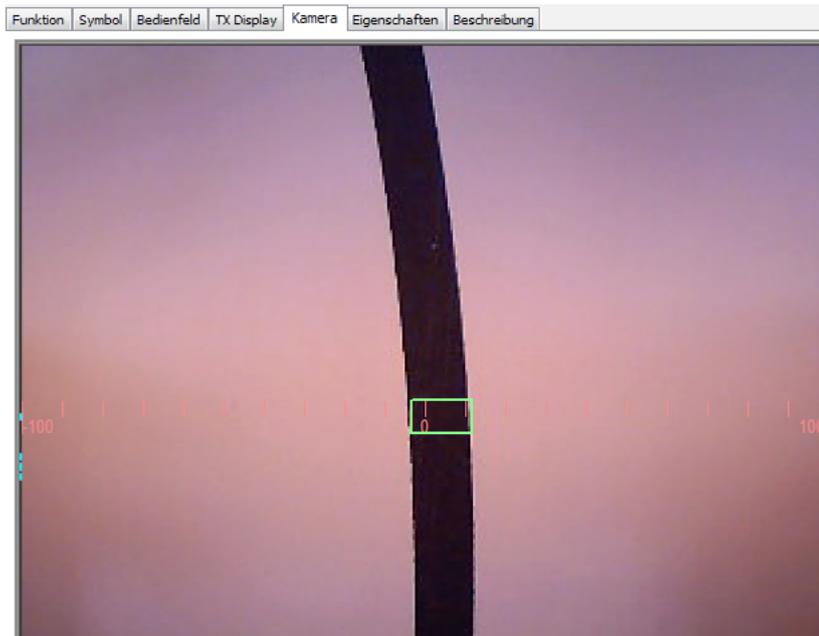
Programmiertipps:

Verwende zum Ausprobieren die schwarze Spur des Parcours aus dem Baukasten.

Im Kamerafenster von ROBO Pro stellst du die Kamera so ein, dass die Spur scharf zu erkennen ist.

Für die Spurverfolgung verwendest du das Kamera-Sensorfeld „Linie“. Es besteht aus einer geraden Linie im Kamerabild, die von links nach rechts durchs Bild gezogen wird und eine Skala von -100 bis +100 enthält. 0 ist genau die Mitte.





Weitere Details zum Sensorfeld „Linie“, siehe [ROBO Pro Hilfe](#) Kapitel 11.

Nun fragst du in deinem Programm ab, an welcher Position sich die Spur befindet.

- Befindet sie sich zwischen -10 und $+10$ soll der Roboter geradeaus fahren (M1 und M2 = links: $v=5$)
- Zwischen -11 und -40 soll er leicht nach links korrigieren (M1links: $v=5$, M2 links: $v=2$)
- Zwischen $+11$ und $+40$ soll er leicht nach rechts korrigieren (M1 links: $v=2$, M2 links: $v=5$)
- Ist der Wert < -40 soll er stark nach links korrigieren (M1links: $v=5$, M2 rechts, $v=3$)
- Ist der Wert $> +40$, soll er stark nach rechts korrigieren (M1rechts: $v=3$, M2 links: $v=5$)
- Verliert er die Spur, soll er stehen bleiben.

Das fertige Programm findest du hier:

Spurensucher_1.rpp

Reagieren kann dein Roboter jetzt. Allerdings wäre es noch besser, wenn der Roboter am Ende der Spur anstatt anzuhalten wieder umdrehen und der Spur erneut folgen würde.

Aufgabe 2: Am Ende der Spur umdrehen und ihr erneut folgen

Erweitere dein Programm um die Funktion, dass der Roboter beim Verlassen der Spur umdreht und ihr wieder folgt.

**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:

Spurensucher_2.rpp

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

Schreibe ein Unterprogramm „suchen“, mit dem der Roboter nach der schwarzen 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:

Spurensucher_3.rpp



Erkundungsroboter

Bei diesem Modell wollen wir nun mehrere Kamerafunktionen kombinieren. Der Roboter soll nach vorne schauen können um seine Umgebung zu erkunden, er soll aber auch in der Lage sein nach unten zu sehen und den verschieden farbigen Spuren auf dem mitgelieferten Parcours zu folgen. Darüber hinaus soll er mit seinem NTC-Widerstand auch noch „Temperaturdaten“ seiner Umgebung sammeln.

Baue zunächst das Modell wie in der Bauanleitung beschrieben auf.

Aufgabe 1: Der Roboter soll wie folgt funktionieren:

Zunächst wird er auf die schwarze Spur gesetzt. An dieser fährt er entlang, bis sie zu Ende ist. Dann dreht er um und fährt in die andere Richtung. Ist er an dem Ende der schwarzen Spur angelangt an der die Spur verbreitert ist, soll er die Kamera heben und nach vorne schauen. Hältst du ihm dann ein rotes oder grünes Kärtchen „vor die Nase“ soll er anschließend zunächst der schwarzen Spur folgen und dann auf die rote oder grüne Spur abbiegen. Bevor er wieder losfährt misst er auch noch die Umgebungstemperatur und gibt sie am Display des TXT Controllers aus.



Die farbigen Kärtchen kannst du aus der großen Schnittvorlage im Baukasten ausschneiden

Programmiertipps:

Mit diesem Programm wollen wir dir zeigen, was mit ROBO Pro alles möglich ist. Da uns bewusst ist, dass das zum Programmieren nicht ganz einfach ist, lade einfach das vorbereitete Programm und probiere es aus:

Erkundungsroboter.rpp

Aufgabe 2:

Das Unterprogramm „Spur“ ist für die Steuerung der Encodermotoren während der Spurverfolgung zuständig. Vergleiche es mit dem Unterprogramm aus dem Modell Spurensucher. Welche Unterschiede fallen dir auf?

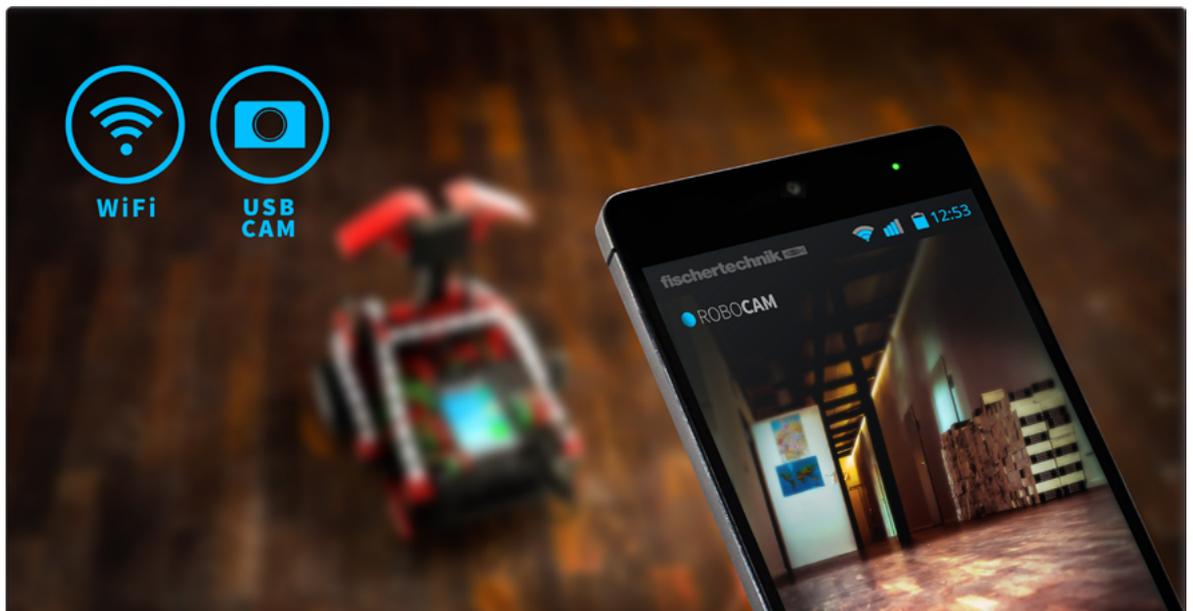


Lösung:

Das Unterprogramm beim Erkundungsroboter ist wesentlich komplizierter als beim Spurensucher. Es enthält eine Regelung, die dafür sorgt, dass das Modell umso stärker die Richtung korrigiert, je weiter die Spur von der Mitte entfernt ist. Beim Spurensucher hingegen wurde nur unterschieden zwischen geradeaus, leicht korrigieren und stark korrigieren. Um so eine Regelung erstellen zu können, muss man beim Programmieren schon sehr viel Erfahrung haben (so wie unser Programmierer). Dafür ist das Ergebnis der Regelung aber auch wesentlich besser als beim Modell Spurensucher. Für eigene Programme kannst du zukünftig einfach dieses Unterprogramm verwenden und dich darüber freuen, dass es so gut funktioniert.

Aufgabe 3:

Falls du ein Smartphone (zunächst mit Android Betriebssystem) besitzt, kannst du damit den Erkundungsroboter fernsteuern. Eine App dafür kannst du dir aus dem Google Playstore herunterladen. Sie heißt „TXTCamdroid“. Verbinde den TXT Controller über WLAN mit deinem Smartphone und los geht es. Am Display des Smartphones kannst du sehen was die Kamera sieht. Über verschiedene Tasten kannst du das Modell steuern und es auf Erkundungsfahrt schicken. Viel Spaß!



Fußballroboter mit Bewegungssteuerung



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 zwei Vorschläge zum Bau von Fußball-Robotern. Sie sind genau so beweglich wie unsere anderen Roboter, haben aber zusätzlich noch eine Lichtschranke um einen Ball zu erkennen, und damit einen „Abschussmechanismus“ auszulösen. Mit der Kamera können wir die Roboter nun auf unterschiedliche Weise ausstatten und sie steuern.

Baue zunächst das Modell „Fußball-Roboter mit Bewegungssteuerung“ wie in der Bauanleitung beschrieben auf.

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 verwendest du den orangen Tischtennisball, der im Baukasten enthalten ist .

Hinweis: Den Taster, der als Endschalter für den Hebel des Abschussmechanismus verwendet wird, musst du eventuell etwas justieren, so dass er zwar gedrückt wird, wenn sich der Hebel vorne befindet, gleichzeitig aber den Hebel nicht blockiert.

Bei diesem Roboter wird die Kamera nicht in das Modell eingebaut, sondern steht auf einem eigenen Ständer neben dem Modell. Sie wird direkt an der USB Schnittstelle des PC eingesteckt.

Aufgabe 1: „Da hat er den Ball erwischt, und er schießt ...“

Im ersten Schritt soll unser elektronischer Ballkünstler lernen, den Ball abzuschießen, 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.

**Programmiertipps:**

Für diese Aufgabe kannst du zunächst den TXT Controller über das USB Kabel mit dem PC verbinden.

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:

Fussballroboter_Bewegungssteuerung_1.rpp

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

Aufgabe 2: Bewegungssteuerung des Fußball-Roboters

Der Fußball-Roboter soll nun so programmiert werden, dass er auf Bewegungen, die du vor der Kamera ausführst, reagiert. Winkst du mit der linken Hand, soll er nach links fahren, winkst du mit rechts, soll er sich nach rechts drehen. Winkst du mit einer deiner Hände in der Mitte des Bildes, soll er gradeausfahren. Erkennt die Lichtschranke den Ball, soll er abgeschossen werden, möglichst natürlich ins Tor.

**Programmiertipps:**

Hinweis: Für diese Aufgabe muss dein TXT Controller über WLAN oder Bluetooth mit deinem PC verbunden werden.

Auch wenn sich diese Aufgabenstellung zunächst etwas schwierig

anhört, Die meisten Funktionen für dieses Programm hast du bei deinen anderen Robotern schon verwendet.

Für die Bewegungssteuerung benutzt du drei Kamera-Sensorfelder „Bewegung“*. Du platzierst eines davon in der Mitte des Kamerabildes, eines im linken und eines im rechten Bereich. Platziere die Kamera so vor deinem Bildschirm, dass du mit deinen Armen die drei Bereiche im Kamerafenster gut erreichen kannst, aber die Sensorfelder nicht ungewollt reagieren. Dazu platzierst du die Felder am Besten in der oberen Hälfte der Kamera-Oberfläche.

Dann fragst du einfach in deinem Programm die drei Sensorfelder ab, und je nachdem in welchem Feld eine Bewegung erkannt wird, steuerst du deinen Roboter nach links, rechts oder geradeaus. Den Abschussmechanismus hast du ja schon in Aufgabe 1 programmiert. Ihn kannst du in dein Programm integrieren.

Dann startest du das Programm im Onlinemodus von ROBO Pro. Das Kamerabild kommt direkt über die USB-Schnittstelle in den PC, die Steuerung des Roboters erfolgt ohne Kabel über Bluetooth oder WLAN.

Von dem Fußballstadion, das im Baukasten enthalten ist, benötigst du für dieses Modell nur das Tor. Die „Bande“ kannst du höchstens verwenden, um zu verhindern, dass der Ball irgendwo hinrollt. Zur Orientierung des Roboters wird sie aber erst beim nächsten Modell benötigt. Und nun viel Spaß beim „Winken“ und Tore schießen.

Auch für dieses Programm gibt es wieder einen Lösungsvorschlag:

Fussballroboter_Bewegungssteuerung_2.rpp

* siehe auch [ROBO Pro Hilfe](#), Kapitel 11.

Fußballroboter

Es ist zwar ganz witzig den Fußballroboter durch Winken mit den Armen zu steuern, aber eigentlich sollte ein richtiger Fußball-Roboter ja in der Lage sein den Ball selbstständig ins Tor zu befördern. Das wollen wir ihm nun beibringen und bauen dazu die Kamera direkt in den Roboter ein.



Wie das gemacht wird, zeigt dir wieder die Bauanleitung.

Die Kamera wird an der [USB-1](#) Schnittstelle des TXT Controllers angeschlossen.

Für dieses Modell wird das komplette Fußballstadion benötigt. Es besteht aus der weißen, unbedruckten Rückseite des bereits beim Spurensucher und beim Erkundungsroboter verwendeten Parcours sowie der mit schwarzen Strichen bedruckten Bande und dem Tor.

Den ganz weißen Untergrund braucht der Roboter um darauf mit dem Kamera-Sensorfeld „Ball“ den orangen Ball zu erkennen.

Anhand der schwarzen Striche auf der Bande, die auf jeder Seite anders aussehen, kann der Roboter erkennen, an welcher Stelle des Spielfelds er sich befindet und wo das Tor ist. Die Striche an der Rückwand des Tors zeigen ihm wohin er schießen muss, wenn er den Ball hat.

Aufgabe 1:

Ähnlich wie beim Erkundungsroboter ist auch das Programmieren dieses Roboters nicht ganz einfach. Probiere deshalb zunächst das mitgelieferte Programm aus, indem du es auf den TXT Controller lädst.



Hier unser Lösungsvorschlag:

Fussballroboter.rpp

Tipps:

Stelle die Kamera am Objektiv so ein, dass die Striche auf der Tor-Bande im Stadion scharf zu erkennen sind, wenn der Roboter am anderen Ende des Stadions steht.

Stelle den Roboter in das Stadion und werfe dann den Ball auf das Spielfeld. Der Roboter sucht den Ball, versucht ihn einzufangen und schießt, sobald er sich in Richtung Tor gedreht hat, den Ball in Richtung Tor. Mit etwas Glück gelingt es dem Roboter ein Tor zu schießen.

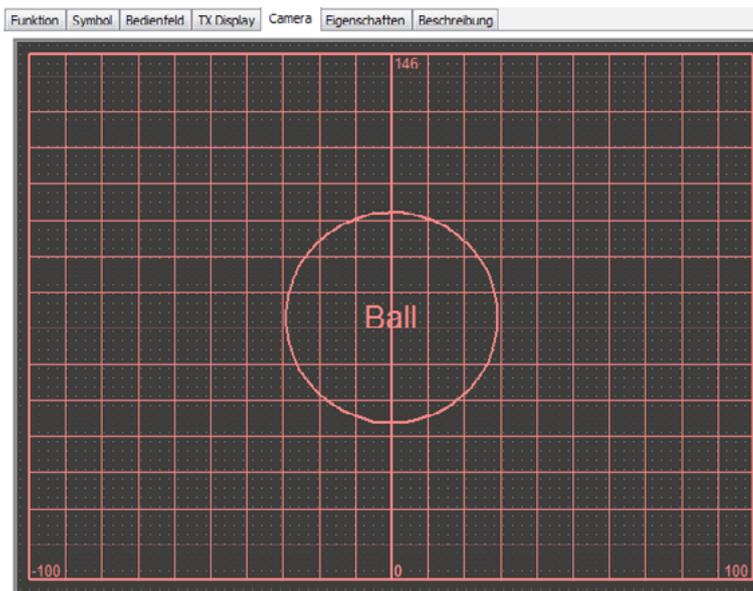
Aufgabe 2:

Versuche anhand der Beschreibung in der ROBO Pro Hilfe zu verstehen, wie der Ballerkenner funktioniert.



Lösung:

Das Sensorfeld „Ball“ erkennt einen farbigen Ball auf einem weißen Untergrund. Es gibt die Position des Ballmittelpunktes in dem Sensorfeld aus. Siehe auch [ROBO Pro Hilfe](#), Kapitel 11.



Von links nach rechts ist, ähnlich wie beim Spursucher, 0 genau die Mitte. In der Höhe geht der Wert von 0 bis zu einem automatisch berechneten Höchstwert. Je nachdem wo sich der Ball befindet, wählt der Roboter seine Fahrtrichtung.

Aufgabe 3:

Welches Kamera-Sensorelement verwendet das Programm um die Striche auf der Bande auszuwerten?



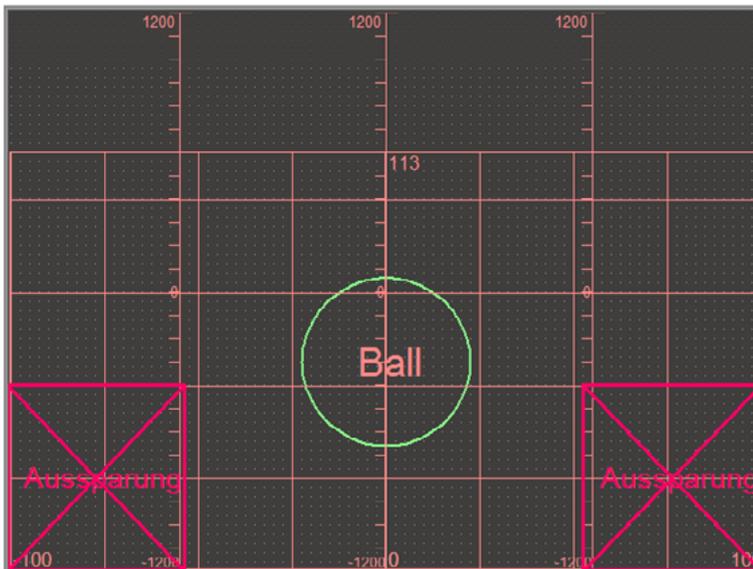
Lösung:

Es wird das Sensorelement „Linie“ verwendet. Drei Linienerkenner werden im Gegensatz zum Spurensucher von oben nach unten im Kamerafenster platziert und erkennen so die drei waagrecht verlaufenden Striche auf der Bande. Anhand ihrer Breite berechnet der Roboter, wo er sich gerade befindet und wo er hin muss.

Er kann außerdem anhand der Striche erkennen, wenn er zu nahe an die Bande kommt und stoppt rechtzeitig um zu verhindern, dass er an die Bande stößt.

Aufgabe 4:

In dem Kamerafenster des ROBO Pro Programms für diesen Roboter findest du 2 Sensorfelder „Ausparung“. Kannst du dir vorstellen welche Funktion sie haben?



Lösung:

Diese Sensorfelder ergänzen das Sensorfeld Ballerkenner. Die ausgesparten Bereiche werden von der Ballsuche ausgeschlossen.

Das kann hilfreich sein, wenn sich im Bild Gegenstände befinden, z. B. der [Fototransistor](#) des Fußballroboters, die vom Ballerkenner fälschlicherweise als Ball erkannt werden.

Wie orientiert sich der Roboter auf dem Spielfeld?

Vielleicht hast du dich schon gefragt, wie der Roboter es anstellt, sich anhand der Strichcodes auf den Banden auf dem Spielfeld zurechtzufinden. Das ist in der Tat nicht ganz einfach und funktioniert im Prinzip wie folgt:

Tatsächlich ist der Roboter in der Lage anhand der Bande seine Position im Spielfeld auf etwa 2 cm genau und den Winkel auf etwa 5 Grad genau zu bestimmen. Dazu misst der Roboter die Höhe der Bande an verschiedenen Stellen und berechnet aus der Höhe Abstände und daraus schließlich den Winkel und die absolute Position. Dieses Verfahren nennt man Triangulation.

Die Bande hat immer 3 schwarze und dazwischen 2 weiße Striche. Einer der schwarze Striche ist breiter als die anderen Striche. Daran erkennt der Roboter, um welche Bande es sich handelt. Insgesamt ergeben sich so 5 verschiedene Bandenmuster, eines für jede Seite des Spielfelds und eines für das Tor.

Darüber hinaus verwendet der Roboter die Zählimpulse der Encodermotoren um die Position und den Winkel zwischen den Triangulationen zu berechnen.

Dieses Verfahren nennt man Odometrie. Für kurze Zeiten und Distanzen ist die Odometrie genauer. Sie hat aber den Nachteil, dass sich Fehler mit der Zeit aufaddieren. Daher werden die durch Odometrie gewonnenen Daten mit den Daten aus der Triangulation korrigiert.

Im Programm Fußballroboter gibt es für diese Berechnungen die beiden Unterprogramme Triangulation und Odometrie. Wenn du Lust dazu hast, schau sie dir einfach mal genauer an. Aber erschrecke nicht! Sie sehen ziemlich kompliziert aus, und das sind sie auch.

Wenn dir das alles zu theoretisch erscheint ist das auch kein Problem. Nimm einfach das vorhandene Programm, staune darüber wie es funktioniert und schließe mit deinen Freunden eine Wette ab, wie viele Versuche der Roboter benötigt und wie lange es dauert, bis er das Tor trifft.

Hier noch einige Tipps zum Spielfeld:

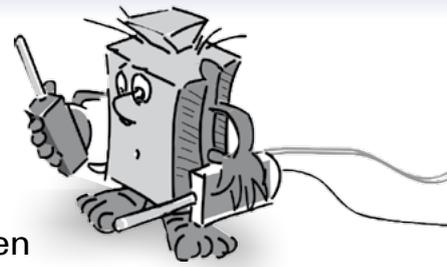
Achte darauf, dass zwischen dem Boden und der Bande kein größerer Spalt besteht. Sonst könnte es sein, dass die Kamera diesen Spalt als Strich erkennt und dann würde die Berechnung der Position nicht stimmen.

Um das zu verhindern ist es wichtig, dass du die Kanten exakt faltest und dass das Spielfeld auf einer ebenen Fläche liegt. Zusätzlich kannst du an den Außenseiten die Banden z. B. mit Büchern oder anderen Gegenständen beschweren, so dass die Bande zusätzlich nach unten gedrückt wird.

Wenn der Ball dauernd in eine Ecke rollt oder öfter direkt an einer Bande liegen bleibt, kannst du das Spielfeld an den Rändern mit flachen Gegenständen, z. B. Zeitschriften, Schulheft, Pappkarton unterlegen, so dass der Ball nicht am Rand liegenbleibt sondern wieder in die Mitte rollt. Dadurch kommt noch mehr Action ins Spiel!

Fehlersuche

Wenn's nicht gleich funktioniert hat das in den meisten Fällen einen einfachen Grund. Nur ist der nicht immer genauso 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 [ROBOTICS TXT Controller](#) angeschlossen hast. Verbinde dazu mit dem Kabel den Akku und die Leuchtlampe. Leuchtet die Lampe 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 [ROBOTICS TXT 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 Ladegerät aufladen. Der Akku ist vollständig geladen, wenn die rote LED am Ladegerät aufhört zu blinken und durchgehend leuchtet.

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 [ROBOTICS TXT 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.

Kamerafunktion

Am besten funktioniert die Kamera bei dem richtigen Licht.

Zu wenig Licht: Im Dunkeln kann sie weder Farben noch Bewegungen erkennen.

Abhilfe: LEDs die das Kamerasichtfeld beleuchten.

Zu viel Licht: Bei grellem Licht – z. B. durch direktes Sonnenlicht – verändert sich der Farbwert und der Kontrast, so dass der Linienerkenner die Linie nicht mehr ausreichend erkennt.

Abhilfe: Direkte Sonneneinstrahlung vermeiden oder verringern, z. B. durch Abschatten mit Vorhängen oder Rollläden.

Zusätzliche Abhilfe: Die Kameraeigenschaften per Software anpassen. Das geschieht in der ROBO Pro Software und dort in den Eigenschaftsfeldern der Kamera-Sensorfelder. Dort kann z. B. die Empfindlichkeit für die Erkennung eines Objekts eingestellt und an die Umgebung angepasst werden. Details dazu kannst du in der [ROBO Pro Hilfe](#) im Kapitel 11 nachlesen.

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?

Noch nicht am Ende 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 „ROBOTICS“.

Fantasie Was wir dir hier gezeigt haben ist nur ein winziger Teil der Möglichkeiten, die du mit dem ROBOTICS TXT 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.

Vorhandenes ändern 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.

Zeichenmaschine Zum Beispiel könnte man am Fahrroboter 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 Fahrroboter ist eine Zeichenmaschine geworden.

Wettrennen mit Freunden Du kannst mit einem dicken Filzstift auch einen eigenen Parcours auf ein Papier zeichnen und den Roboter daran entlang fahren lassen. Und wenn Freunde von dir auch einen ROBOTICS TXT Controller haben, dann wird es noch interessanter. Dann könnt ihr Wettrennen veranstalten, wessen Roboter schneller einen vorgegebenen Parcours meistert. Außerdem kann über die Bluetooth-Schnittstelle nicht nur dein PC mit dem ROBOTICS TXT Controller kommunizieren, sondern auch mehrere Controller untereinander. So könntet Ihr dann zum Beispiel 2 Roboter programmieren, die aufeinander reagieren. Oder miteinander tanzen. In den Kapiteln 4.5 und 7 der [ROBO Pro Hilfe](#) findest du einige interessante Infos zu diesem Thema.

**Kabellos
losgelöst**

Hat dein Rechner eine WLAN-Schnittstelle? Wenn ja, dann kannst du den ROBOTICS TXT Controller anstatt über ein USB-Kabel auch über WLAN mit dem Rechner verbinden. Wenn nicht, kannst du dir auch einen USB-WLAN-Stick kaufen und darüber den ROBOTICS TXT Controller kabellos mit deinem PC verbinden. Wie das geht kannst du in der Bedienungsanleitung des ROBOTICS TXT Controllers und auf <http://www.fischertechnik.de> nachlesen.

**Ausdauer
lohnt sich**

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.

