

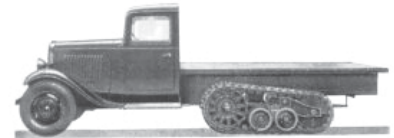
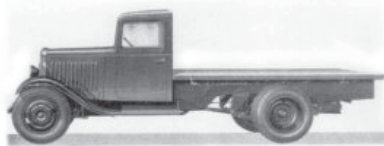
Fahrzeuge mit Raupenantrieb	S.2
Die Lenkung	S.2
Explorer-Modell aus fischertechnik	S.3
Aktoren	S.3
Sensoren	S.4
ROBO TX Controller	S.5
Stromversorgung	S.5
Software ROBO Pro	S.5
Vorüberlegungen	S.6
Grundlagen zum ROBO TX Controller	S.6
Das Basismodell	S.7
Basisprogramm	S.7
Encodermotoren	S.8
Unterprogramme	S.8
Autonome Raupenfahrzeuge	S.9
Spursucher	S.9
Tunnelroboter	S.12
Farberkenner	S.13
Explorer	S.14
RoboCupJunior – Rescue Robot	S.16
Wichtige Tipps	S.16

Inhalt



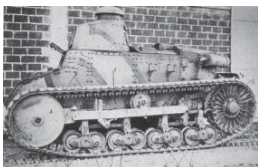
Fahrzeuge mit Raupenantrieb

■ Wozu benötigt man autonome Fahrzeuge mit Raupenantrieb? Die Erfindung des Raupenantriebs für Fahrzeuge wurde notwendig, um auch unwegsames Gelände zu überwinden. Wo ein Reifenantrieb nicht mehr funktionierte, z. B. in der Wüste kam der Raupenantrieb zum Einsatz. So wurden im 1. Weltkrieg die ersten raupenangetriebenen Lastkraftwagen und Panzer gebaut und eingesetzt.



Je nach Gelände konnte man die Fahrzeuge von Reifen- auf Raupenantrieb umrüsten.

Auch in der zivilen Nutzung waren Raupenfahrzeuge im Einsatz. Wie du schon anhand der Bilder erkennen kannst, waren eigentlich immer radangetriebene Fahrzeuge Grundlage für Raupenfahrzeuge.



Aber schon bald stellte man einen Schwachpunkt fest: Die lenkbaren Vorderräder. Deshalb ging man dazu über, den Raupenantrieb auf alle Achsen zu erweitern.

Die Lenkung



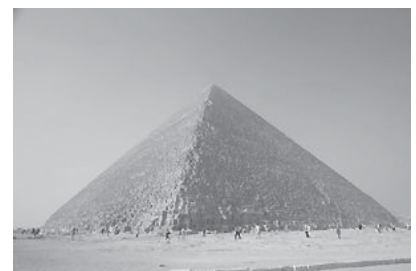
■ Aber wie funktionierte dann die Lenkung? Ganz einfach, durch Verlangsamen oder Beschleunigen einer der beiden Raupen. Wollte man eine Rechtskurve fahren, verlangsamte man durch einen Steuerknüppel (pro Raupe einer) die rechte Raupe. Dadurch drehte sich diese langsamer und somit wurde das Fahrzeug nach rechts bewegt.

Auch heute, natürlich auf dem neuesten Stand der Technik, findest du viele raupenangetriebene Fahrzeuge. Vom Kleinbagger bis zu Riesenkolossen im Übertagebau für Braunkohle.



In der Cheops-Pyramide in Ägypten wollte man mit Hilfe eines Miniroboters weiteren Geheimnissen auf die Spur kommen.

Die Forscher schickten den Roboter von der Größe einer Spielzeuglokomotive durch einen dunklen, schmalen Schacht. Er führt hinauf von einer Kammer im Zentrum der 4500 Jahre alten Pyramide und endet vor einer geheimnisvollen Steintür.



■ Unbekannte Räume erkunden, Abstände messen, Spuren folgen, Fahrrichtungen durch Blinksignale anzeigen, Farben erkennen, Temperaturen messen, berührungslos Hindernissen ausweichen, Tag und Nacht erkennen, Scheinwerfer automatisch ein- und ausschalten, Alarm auslösen, etc. Das alles – und noch viel mehr – ermöglichen die Sensoren des ROBO TX Explorer. Im Einzelnen ist das: der NTC-Widerstand, der Fotowiderstand, der Ultraschall-Abstandssensor, der optische Farbsensor sowie der speziell entwickelte Infrarot-Spursensor. Dank zweier Encodermotoren und des Raupenantriebs kann auch unwegsames Gelände erforscht und befahren werden. Mit dem enthaltenen Modell Rescue Robot ist der Baukasten eine ideale Grundlage zur Teilnahme am RoboCup-Junior.

Bevor du nun loslegst, solltest du dich etwas mit den wichtigsten Bauteilen beschäftigen. Diese werden nachfolgend beschrieben:

Encodermotor

Als Antrieb für unsere Roboter verwenden wir die zwei im Baukasten enthaltenen Encodermotoren. 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 Encodermotoren 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 Encodermotoren erzeugen pro Umdrehung der Motorwelle 3 Impulse. Und weil die Encodermotoren 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.

Die Encodermotoren werden am ROBO TX Controller an den Ausgängen M1 bis M4 angeschlossen. Die Encodersignale werden über die Eingänge C1 bis C4 eingelesen.

Summer

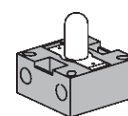
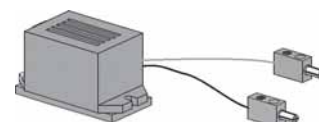
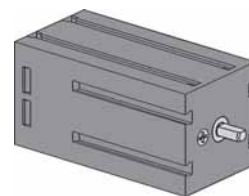
Der Summer dient z. B. dazu, erkannte Hindernisse oder Farben akustisch zu melden. Angeschlossen wird er ebenfalls an den Ausgängen M1 bis M4.

Kugellampe

Hier handelt es sich um eine Glühlampe für eine Spannung von 9 V. Diese kann als Meldesignal für die Fahrtrichtung oder einfach als Beleuchtung eingesetzt werden. Sie wird an den Ausgängen M1 bis M4 angeschlossen.

Explorer-Modell aus fischertechnik

Aktoren



Sensoren

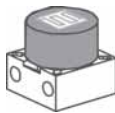
■ Sensoren sind gewissermaßen die Gegenstücke zu den Aktoren. Denn sie führen keine Aktionen aus, sondern reagieren auf bestimmte Situationen und Ereignisse.

Die Sensoren werden an die Universaleingänge I1 bis I8 des ROBO TX Controllers angeschlossen.



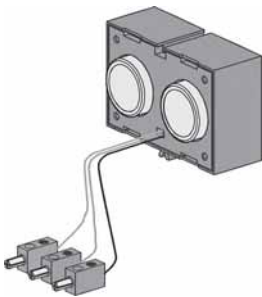
NTC-Widerstand (Negativ Temperatur Coeffizient)

Hier handelt es sich um ein Bauteil, mit dem du verschiedene Temperaturen messen kannst. Man spricht auch von einem Wärmesensor. Bei ca. 20 Grad hat der NTC-Widerstand einen Wert von 1,5 kOhm. Steigt die Temperatur, nimmt der Widerstandswert ab. Diese Information steht in ROBO Pro als Zahlenwert zur Verfügung.



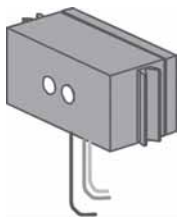
Fotowiderstand

Der LDR 03, ein analoger Helligkeitssensor, reagiert auf Tageslicht und verändert dabei seinen Widerstandswert. Dieser ist ein Indikator für die Lichthelligkeit.



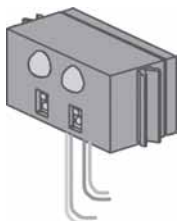
Ultraschall-Abstandssensor

Ein Abstandssensor ist ein technisches Bauteil, das in der Lage ist, den Abstand zwischen sich und einem Objekt zu messen. Abstandssensoren arbeiten mit Licht, Infrarotstrahlung, Radiowellen, oder **Ultraschall** und verwenden unterschiedliche Messmethoden. Schall breitet sich als Welle aus. Ein Echo wird zur Ultraschallquelle reflektiert, welches nun als Signal wieder gefangen und ausgewertet wird. Die Zeitdifferenz zwischen Aussenden und Empfangen des Signals gibt Aufschluss über die Distanz zwischen Hindernis und Sensor. Die Reichweite des Sensors beträgt bis zu 4 m. Der ausgegebene Zahlenwert entspricht dem Abstand in Zentimeter.



Optischer Farbsensor

Farbsensoren werden meist in der Automatisierungstechnik eingesetzt. Dabei soll z. B. die Farbe oder ein Farbaufdruck kontrolliert werden, um sicher zu gehen, dass die richtigen Bauteile eingebaut werden. Der fischertechnik Farbsensor sendet rotes Licht aus, das von verschiedenen Farbflächen unterschiedlich stark reflektiert wird. Die Menge des reflektierten Lichts wird über den Fototransistor gemessen und als Spannungswert zwischen 0 V und 10 V ausgegeben. Der Messwert ist abhängig von der Umgebungshelligkeit sowie vom Abstand des Sensors zur Farbfläche. Diesen Wert kannst du einlesen und als Zahlenwert von 0 - 10 000 in deinem Programm verarbeiten.



Spursensor

Der IR-Spursensor ist ein digitaler Infrarotsensor zur Erkennung einer schwarzen Spur auf weißem Untergrund im Abstand von 5 - 30 mm. Er besteht aus zwei Sende- und zwei Empfängerelementen. Als Anschluss benötigst du zwei Universaleingänge und den 9 V-Spannungsausgang.

■ Das wichtigste Bauteil, um ein Raupenfahrzeug aufzubauen, ist der ROBO TX Controller, welcher fest in die verschiedenen Modelle eingebaut wird. An ihm schließt du je nach Bedarf deine Sensoren und Aktoren an. Die Grundverdrahtung entnimmst du der beliegenden Bauanleitung.

ROBO TX Controller



■ Bei den ROBO TX Explorer-Modellen handelt es sich um autonome Fahrzeuge, die sich frei im Raum bewegen. Als Stromversorgung verwendest du deshalb das fischertechnik Accu Set.

Stromversorgung

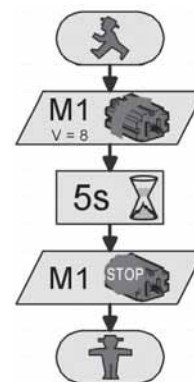


■ ROBO Pro ist eine einfache grafische Programmieroberfläche mit der du deine Programme schreiben kannst. Der Vorteil ist, dass du keine Programmiersprache lernen musst. Eigentlich kannst du sofort loslegen.

Software ROBO Pro

Für den Baukasten ROBO TX Explorer benötigst du die ROBO Pro Version 2.1.4.2. Falls du eine ältere Version der Software besitzt, kannst du sie kostenlos updaten. Entweder über das Hilfe-Menü in ROBO Pro – Neue Version downloaden oder unter

www.fischertechnik.de/robopro/update.html



Vorüberlegungen

■ Wie bei allen anderen fischertechnik-Robotern wirst du dich beim ROBO TX Explorer schrittweise in die Faszination der Technik und Programmierung einarbeiten. Du beginnst mit einem einfachen Modell und arbeitest dich bis zu immer umfangreicheren Systemen mit faszinierenden Möglichkeiten vor. Im Vordergrund steht bei allen Modellen ein sorgfältiger Aufbau und eine sorgfältige Inbetriebnahme.

Grundlagen zum ROBO TX Controller

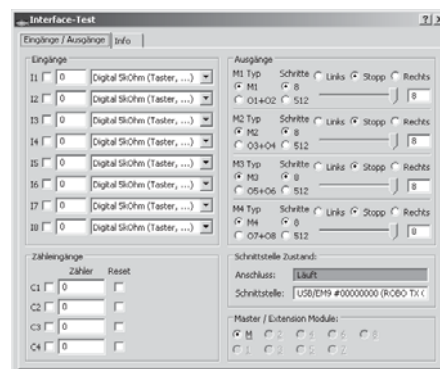
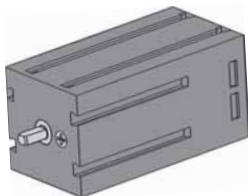
■ Bevor du dich an die einzelnen Modelle heranwagst, solltest du dich mit Hilfe von einigen Experimenten mit dem Controller vertraut machen. Bei Problemen kannst du auch in der ROBO Pro Hilfe nachlesen.

Nachdem du die Software installiert hast, kannst du den Controller über das mitgelieferte Kabel an deinem PC anschließen. Starte nun das Programm ROBO Pro und öffne das Fenster für den Interface-Test mit dem Test-Knopf.



Encodermotor

Verbinde die Anschlüsse des Encodermotors mit dem Anschluss M1. Klicke mit der Maus auf die Auswahl „Links“ oder „Rechts“. Der Motor läuft mit seiner Maximalgeschwindigkeit an. Durch Betätigen des Reglers kannst du die Umdrehungsgeschwindigkeit einstellen. Über „Stopp“ wird der Vorgang beendet.



Fotowiderstand

Schließe den mitgelieferten Fotowiderstand an dem Anschluss I1 an und stelle den Eingang auf „Analog 5 kOhm (NTC, ...)“ ein.

Verändere die Lichtstärke des Fotowiderstands durch langsames Abdecken mit einem schwarzen Papierstreifen. Was passiert? Du wirst sehen, es verändert sich der Zahlenwert des Eingangs.



Der Interface-Test wird im Kapitel 2 der ROBO Pro Hilfe sehr gut erläutert. Ebenfalls findest du Hilfe, falls es Probleme zwischen deinem Rechner und dem Controller und der Software gibt – reinschauen lohnt sich!

■ Jetzt geht es los. Nachdem du mit dem ROBO TX Controller und der Programmierung vertraut bist, kannst du nun die erste Aufgabe lösen. Zuerst wird das Basismodell anhand der Bauanleitung aufgebaut.



Aufgabe 1 - ROBO Pro Level 1:

Dein Raupenfahrzeug soll 6 Sekunden lang geradeaus fahren, anschließend soll es sich 3 Sekunden lang nach rechts drehen und dann stehen bleiben.

Bei deinem ersten Programm wollen wir dich noch etwas unterstützen. Klicke als Erstes auf den Button „Datei-Neu“. Dein Programm beginnt mit einem grünen Ampelmännchen für Programmstart.



Anschließend benötigst du 2 Motorsymbole. Setze das erste Symbol so unter den Programmstart, dass die Verbindung automatisch gezogen wird. Bewege die Maus auf das Motorsymbol und schalte das Eigenschaftsfenster ein (rechte Maustaste). Dort stellst du den Motorausgang „M1“ und bei Aktion die Drehrichtung „Links“ ein. Dann bestätigst du mit OK. Füge in gleicher Weise das zweite Motorsymbol ein und wiederhole den Vorgang für den Motorausgang „M2“.

Als nächstes soll das Programm eine bestimmte Zeit warten. Dazu verwendest du das Wartezeit-Symbol. Dieses platzierst du unter den zweiten Motor und stellst die Zeit auf 6 Sekunden ein.

Anschließend soll sich das Raupenfahrzeug 3 Sekunden drehen. Dazu fügst du nochmals zwei Motorsymbole für M1 und M2 ein. M1 soll sich nach links und M2 nach rechts drehen. Da beide Motoren 3 Sekunden arbeiten sollen, füge anschließend das Wartezeit-Symbol ein und setze den Zeitwert auf 3 Sekunden.

Danach musst du beide Motoren stoppen. Dies geschieht wie vor der Drehung mit dem Einfügen der beiden Motorsymbole und der Parametereinstellung „Stopp“.

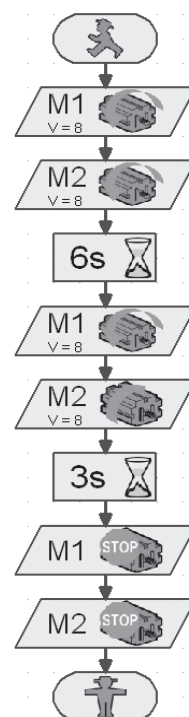
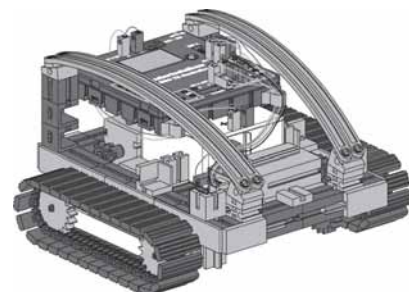
Zum Schluss musst du noch das Symbol für Programmende „Rotes Ampelmännchen“ einfügen. Jetzt ist dein erstes Programm fertig und du kannst es speichern. Teste es dann im Onlinemodus. Klicke dazu auf den Button „Start“.

Hast du alles richtig gemacht, kannst du das Programm auf den Controller laden. Dazu klickst du auf den Button „Download“. Übernehme die Einstellung des Download-Fensters. Gleich nach dem Download fährt das Modell los. Leider hängt es aber noch am USB-Kabel. Lade das Programm nochmals, aktiviere aber „Programm über Taster am Interface starten“. Ist das Programm übertragen, kannst du das Kabel abziehen. Drücke zum Starten des Programms die linke Auswahl Taste des Controllers.

Das fertige Programm findest du unter:

C:\Programme\ROBOPro\Beispielprogramme\ROBO TX Explorer\Basismodell_1_TX.rpp

Das Basismodell



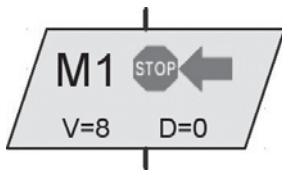
Programm im Online-Modus starten



Programm zum ROBO TX Controller herunterladen

Encodermotoren

■ Wie du bestimmt gemerkt hast, fährt dein Modell nicht exakt geradeaus. Das hat verschiedene Gründe. Einer davon ist, dass sich die beiden Motoren nicht genau gleich schnell drehen. Zum Beispiel kann das Getriebe an 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 Encodermotoren ganz einfach machbar.



Encodermotor-Element

Aufgabe 2 – ROBO Pro Level 1:

Wiederhole die letzte Aufgabe, 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 11.6 beschrieben.

Finde heraus, wie lange Du die Motoren in unterschiedliche Richtungen laufen lassen musst, damit der Roboter sich um 90° dreht. Verändere hierzu die Distanzangabe in dem Encodermotor-Element, mit dem der Roboter gedreht wird.



Das fertige Programm findest du unter:

C:\Programme\ROBOPro\Beispielprogramme\ROBO TX Explorer\Basismodell_2_TX.rpp

Unterprogramme

■ Für die Lösung der nächsten Aufgabe werden Unterprogramme benötigt. Lese dazu in der ROBO Pro Hilfe das Kapitel 4.1 durch. Wichtig ist, dass du in ROBO Pro auf **Level 2** umschaltest.



Neues Unterprogramm erzeugen

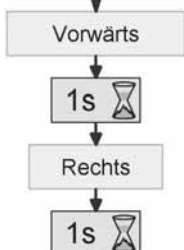


Aktuelles Unterprogramm kopieren



Aktuelles Unterprogramm löschen

Lösung



Aufgabe 3 – ROBO Pro Level 2:

Dein Raupenfahrzeug soll ein Quadrat abfahren. Verwende die gleichen Parameter wie in der 2. Aufgabe. Erstelle für jede Fahrtrichtung ein Unterprogramm.

Erstelle zuerst das Unterprogramm „Vorwärts“ (siehe ROBO Pro Hilfe Kapitel 4). Markiere die Programmteile und kopiere sie in den Zwischenspeicher. Anschließend erstellst du das Unterprogramm „Links“ und „Rechts“. In beide fügst du aus dem Zwischenspeicher die Programmteile für „Vorwärts“ ein und änderst die Parameter entsprechend.

Als kleine Hilfe haben wir dir einen Teilausschnitt aus der Aufgabe aufgezeigt. Die nachfolgende Tabelle soll dir auf einen Blick zeigen, wie du die Motoren für die Fahrtrichtungen programmieren musst.



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

Anhand dieser Tabelle sind alle Motoren in den Beispielprogrammen programmiert.

Fertiges Programm:

C:\Programme\ROBOPro\Beispielprogramme\ROBO TX Explorer\Basismodell_3_TX.rpp

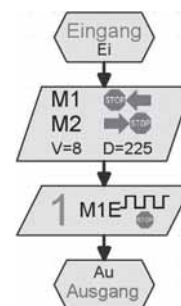
■ Nachdem du mit dem Basismodell genügend experimentiert hast, soll nun dein Roboter auf verschiedene Signale von außen reagieren.

Damit dein Raupenfahrzeug seine Umgebung erkennt und bestimmte Aufgaben erfüllt, musst du es mit Sensoren ausstatten. Die folgenden Modellvorschläge zeigen dir verschiedene Varianten von Raupenfahrzeugen mit unterschiedlichen Sensoren. So sollen verschiedene Wegstrecken, Licht oder Farben aber auch Wärmequellen oder Abstände erkannt werden. Die einzelnen Programme findest du im Verzeichnis: C:\Programme\ROBOPro\Beispielprogramme\ROBO TX Explorer\

Sicher sind dir aus dem Fernsehen Filme von menschenleeren Fabrikhallen, in denen wie von Geisterhand gesteuerte Transportfahrzeuge fahren, bekannt. Zum Teil werden solche Systeme mit im Boden eingelassenen Datenleitungen oder auf dem Boden aufgezeichneten Wegemarkierungen gesteuert.

Grundlage deiner Programmierung soll sein, dass der Roboter an einer schwarzen Linie entlang fährt.

Bevor du mit dem Programmieren beginnst, baue zuerst anhand der Bauanleitung den Spurensucher auf. Einen Experimentier-Parcours mit aufgedruckter schwarzer Linie findest du im Baukasten. Die Linie, an der der Spurensucher entlangfahren soll, sollte zuerst einmal eine Gerade sein.



Unterprogramm „Rechts“

Autonome Raupenfahrzeuge

Spursucher





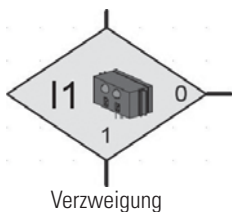
■ **Wie soll nun das Modell funktionieren?**

Der Roboter soll auf einem weißen Untergrund eine schwarze Linie finden und dieser dann folgen. Um dies zu ermöglichen, hast du in deinem Modell den IR-Spursensor eingebaut. Der Baustein sendet ein Licht im Infrarot-Bereich auf den Fahrbahnuntergrund. Dieses wird, je nach Untergrund reflektiert und von Fototransistoren gemessen. Für deine Programmierung bedeutet das: Heller/weißer Untergrund reflektiert das Licht und du erhältst den Wert 1. Bei einem schwarzen Untergrund wird das Licht nicht reflektiert und du bekommst den Wert 0. Haben beide Transistoren den Wert 0 hat dein Roboter den Fahrweg (schwarze Linie) gefunden und muss nun diesem folgen.



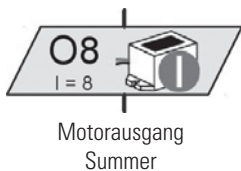
Aufgabe 1 - ROBO Pro Level 2:

Dein Raupenfahrzeug soll auf eine gerade schwarze Spur gestellt werden und dann entlangfahren. Verliert es die Spur oder ist diese zu Ende soll es stehen bleiben und dreimal hupen.



Ein paar kleine Tipps:

Prüfe mit dem Interfacetest die Spurerkennung des Sensors. Denke daran, die Eingänge auf „Digital 10 V (Spursensor)“ einzustellen. Sollte die schwarz-weiß-Erkennung nicht richtig funktionieren, können störende Lichtquellen (z. B. Sonne) die Ursache sein. Gegebenenfalls muss der Sensor etwas näher an die Spur positioniert oder mit einer Bauplatte abgeschirmt werden.

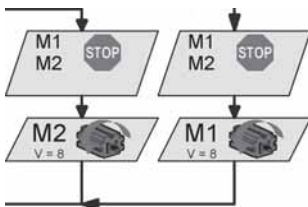


Fertiges Programm: **Spursucher_1_TX.rpp**

■ Sicherlich wirst du mit deiner ersten Lösung nicht zufrieden sein – denn dein Roboter fährt zwar eine kleine Strecke entlang der Linie. Da er aber noch nicht nachjustieren kann, verlässt er die Markierung, bleibt stehen und signalisiert dir dies.

Aufgabe 2 - ROBO Pro Level 2:

Erweitere dein Hauptprogramm in den Abfragezweigen des Spursensors so, dass der Roboter erkennt, wenn er nicht mehr exakt auf der Spur fährt. Dann soll er seine Fahrtrichtung entsprechend korrigieren. Einen Tipp findest du im linken Programm-ausschnitt.

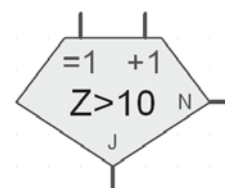


Jetzt sieht es schon besser aus. Dein Roboter bleibt genau auf der vorgegebenen Spur. In einer Industriehalle würden jetzt andere Roboter am Ende der Spur die transportierte Ladung entnehmen oder den Roboter mit einer neuen Ladung versehen. Diese könnte dann zum Ausgangspunkt der Fahrt zurückbefördert werden.

Fertiges Programm: **Spursucher_2_TX.rpp**

Aufgabe 3 - ROBO Pro Level 2:

Dein Roboter ist in den vorherigen Aufgaben an einer bestimmten schwarzen Linie entlanggefahren. In dieser Aufgabe soll er eine Linie suchen. Dazu soll er sich einmal im Kreis drehen. Findet er dabei keine Spur, soll er eine kleine Wegstrecke geradeaus fahren und dann erneut suchen. Hat dein Roboter eine Spur gefunden, soll er der Spur folgen. Endet diese oder verliert er sie, soll er mit dem Suchen erneut beginnen. Hat er sich 10-mal im Kreis gedreht ohne eine Spur zu finden, soll er stehen bleiben und dreimal hupen.



Zählschleife

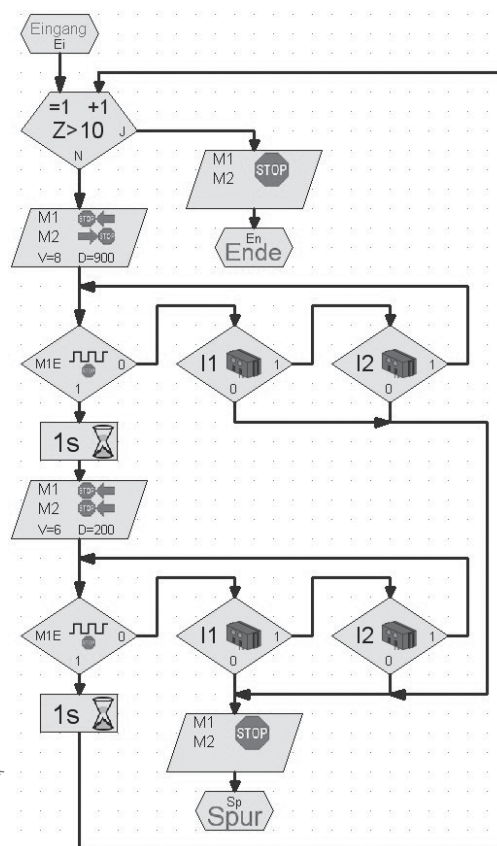
Tipp:

Erinnere dich an die ersten Aufgaben im Basismodell. Hier sollte der Roboter sich um 90 Grad drehen. Dies hast du mit einem Encodermotor-Element bewerkstelligt. Auch hier kann dir diese Technik von Nutzen sein.

Für die Spurensuche legst du ein eigenes Unterprogramm mit dem Namen „Spursuche“ an. Im Bild siehst du, wie wir es vorschlagen würden.

Fertiges Programm: **Spursucher_3_TX.rpp**

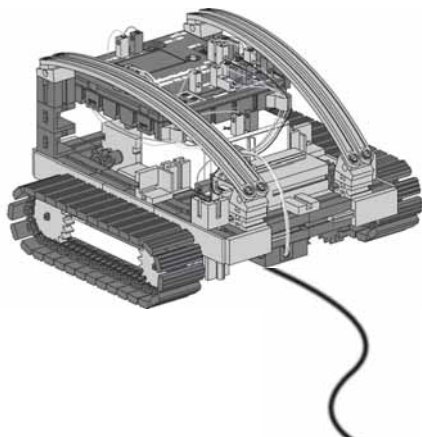
Die bisherigen Fahrstrecken waren immer eine Gerade. Es kommt aber auch vor, dass Fahrstrecken mit Kurven ausgestattet sind. Dies findet man z. B. in Industrieanlagen in denen Material oder Werkstücke von einer Maschine zur nächsten transportiert werden müssen.



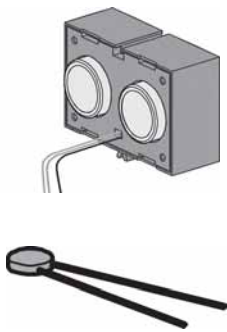
Aufgabe 4 - ROBO Pro Level 2:

Der Experimentier-Parcours enthält verschiedene Kurven mit unterschiedlichen Radien.

Experimentiere bei deinem Rundkurs auch mit unterschiedlichen Geschwindigkeiten von M1 und M2. Mit welchen Einstellungen schafft der Roboter den Parcours am schnellsten? Halte in einer kleinen Tabelle die Ergebnisse fest.



Tunnelroboter, Abstands- und Temperatursensor



- Mit dem Sensor zur Abstandsmessung und dem NTC-Widerstand bieten sich dir weitere Möglichkeiten, dein Modell zu einem Profi-Roboter auszubauen.

Abstände, Temperaturen messen und evtl. Hilfsmaßnahmen einleiten. Was meinst du, wo könnten Roboter mit solchen Fähigkeiten eingesetzt werden? Sicher fallen dir hierzu verschiedene Einsatzgebiete ein. Wir wollen uns mit dem Bereich des Feuerschutzes und der Feuerbekämpfung in Auto- und Eisenbahntunnels befassen.



Die Aufgabe eines solchen Roboters ist es, sich an den Brandherd heranzutasten, Temperaturen im Tunnel zu messen und einer Leitzentrale Daten zu melden. Meist sind die Roboter mit einer mobilen Löscheinrichtung ausgestattet, die je nach Bedingung eingesetzt werden kann.

Baue auch hier gewissenhaft nach der Bauanleitung das Modell „Tunnelroboter“ auf.

Aufgabe 1: ROBO Pro Level 2

Ähnlich dem Spurensucher, der an einer Linie entlang fährt, soll dein Roboter an einer Wand, in einem bestimmten Abstand (ca. 20 cm) eine gewisse Strecke fahren.



Fertiges Programm: **Tunnel_1_TX.rpp**

- Lass uns für die nächste Aufgabe noch einmal auf einen Feuerlöschroboter eingehen. Damit er wie dein Roboter an der Wand entlang fährt, benutzt er Abstandssensoren. Um aber den Brandherd zu erkennen, setzt er Wärmesensoren ein. Dieser Wärmesensor ist für dein Modell der NTC-Widerstand. Die physikalische Eigenschaft dieses Bauteils ist, dass sich bei steigender Temperatur der Widerstandswert verringert. Diese Änderung kannst du wieder mit dem Interface-Test ausprobieren. Schließe den NTC-Widerstand an den Anschluss I6 an. Halte eine Wärmequelle an den NTC und beobachte den Zahlenwert des Eingangs. Denke daran den Eingang auf „Analog 5 kOhm (NTC, ...)“ einzustellen.

Am Analogeingang wird der Widerstandswert des Temperaturfühlers angezeigt, nicht die Temperatur. Um diesen Wert in eine Temperatur umzurechnen, kannst du das Unterprogramm „NTC->T“ verwenden (siehe Tunnel_2_TX.rpp).

Aufgabe 2: ROBO Pro Level 2

Erweitere das Programm, bei welchem der Roboter an der Tunnelwand entlang fährt. Messe zusätzlich die aktuelle Temperatur. Steigt diese auf einen bestimmten Wert, soll dein Roboter anhalten und ein Warnsignal über den Summer ausgeben. Gleichzeitig mit dem Summer soll die rote Warnleuchte ein Blinksignal abgeben.

Nach diesem simulierten Löschvorgang soll dein Roboter wenden und zum Ausgangspunkt zurückkehren.



Fertiges Programm: **Tunnel_2_TX.rpp**

Tipp:

Da dein Roboter nur einen Abstandssensor besitzt, benötigt er zur Rückkehr zum Ausgangspunkt eine zweite Wand an der er entlang fahren kann.

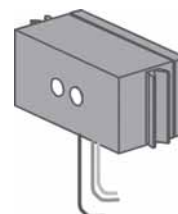
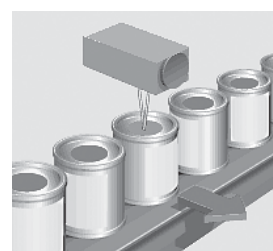
Solltest du in deiner fischertechnik-Sammlung noch einen Motor und eine Luftschraube haben, kannst du auch noch das Löschen des Feuers mit in dein Programm einbeziehen.

■ Als weiteren Sensor wirst du den Farbsensor kennen lernen. Das nebenstehende Bild soll dir ein Beispiel aus dem industriellen Einsatz geben. Wie du erkennen kannst, werden Farbdosen, die falsch eingereiht wurden, aussortiert.

Das vom Tastgut reflektierte Licht wird empfangen, digitalisiert und mittels eines Computers und einer Software aufbereitet. Aufgabe des Sensors ist es, die verschiedenen Farben zu erkennen und Messdaten an den ROBO TX Controller zu senden.

In dem Modell Farberkenner ist der Farbsensor eingebaut. Angeschlossen wird er mit dem schwarzen Draht an I4, dem roten Draht an + und dem grünen Draht an \perp . Für die ersten Testprogramme verwendest du die auf dem Parcours aufgedruckten Farbflächen.

Farberkenner



Aufgabe 1 - ROBO Pro Level 2:

Überprüfe zuerst die Werte, die das Interface für die verschiedenen Farben im Interface-Test ausgibt. Verwende neben den 3 vorgegebenen Farben noch Schwarz und Weiß.

Erstelle eine kleine Tabelle und trage die Werte, die du gemessen hast ein. Beobachte auch Veränderungen, wenn sich der Abstand zur Farbfläche oder das Umgebungslicht ändert.



Aufgabe 2 - ROBO Pro Level 2:

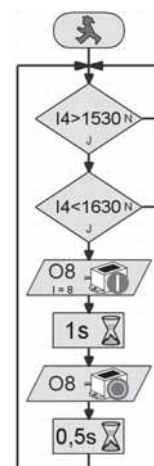
Schreibe ein kleines Programm, mit dem der Sensor die grüne Farbfläche erkennt. Liegt der gemessene Wert im vorgegebenen Wertebereich, wird der Summer eine Sekunde aktiviert. Anschließend springt das Programm zum Start.

Farbe	Wert
Weiß	
Schwarz	
Blau	
Rot	
Grün	

Fertiges Programm: **Farberkenner_2_TX.rpp**

Tipp:

Für die nächste Aufgabe benötigst du die drei Lampen mit verschiedenfarbigen Leuchtkappen, die in dem Modell bereits eingebaut sind.





Aufgabe 3 - ROBO Pro Level 3:

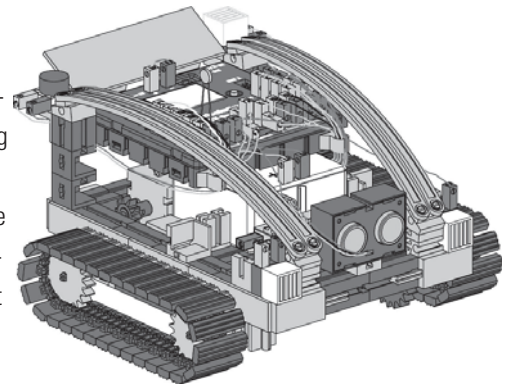
Schreibe ein Programm, welches deinen Roboter eine bestimmte Strecke geradeaus fahren lässt. Auf dieser Strecke befinden sich drei Farbflächen. Erfasst der Sensor eine Farbe, soll der Roboter für 3 Sekunden stehen bleiben. In dieser Zeit schaltet er die Lampen mit der entsprechenden Farbe ein und gibt über den Summer ein Akustiksignal aus. Danach fährt er zur nächsten Fläche und wiederholt seine Arbeit. Anschließend fährt er zur letzten Fläche, meldet das Ergebnis und bleibt dort stehen.

Fertiges Programm: **Farberkenner_3_TX.rpp**

Gesamtmodell Explorer

■ Im Modell „Explorer“ sind alle Aktoren und Sensoren enthalten, die für ein autonomes Roboterfahrzeug nötig sind.

Jetzt sind dir keine Grenzen mehr gesetzt um einfache wie schwierige Aufgaben zu lösen. In den vorangegangenen Baustufen hast du meist nur einen Sensor benutzt um die Einsatzmöglichkeit kennen zu lernen.



Aufgabe 1 - ROBO Pro Level 2:

Programmiere deinen Roboter so, dass er auf seiner Fahrt auf ein vorhandenes Hindernis zufährt. In einem Abstand von ca. 60 cm soll er seine Geschwindigkeit reduzieren. Bei einem Abstand von 40 cm hält er an. Bewegt sich das Hindernis weiter auf deinen Roboter zu, soll dieser ab einem Abstand von 20 cm langsam und ab 10 cm schnell rückwärts fahren.

Fertiges Programm: **Explorer_1_TX.rpp**

Aufgabe 2 - ROBO Pro Level 2:

Jetzt geht dein Roboter auf seine Erkundungsreise. Erstelle ein Programm für den Einsatz von 2 Sensoren-Spursensor und Abstandsensor. Zuerst soll der Roboter auf dem Experimentier-Parcours der schwarzen Linie folgen. Auf der Strecke platzierst du ein Hindernis. Er soll ca. 10 cm vor dem Hindernis anhalten und einen Zentimeter zurückfahren. Danach soll er umdrehen und in der anderen Richtung der Spur folgen.



Fertiges Programm: **Explorer_2_TX.rpp**

Aufgabe 3 - ROBO Pro Level 2:

Das Programm aus Aufgabe 2 soll um drei Sensoren erweitert werden – den Farberkennner, den Temperatursensor und den Fotowiderstand zur Helligkeitsmessung.

Entlang der Spur befinden sich verschiedene Farbflächen. Diese meldet der Roboter durch verschiedene akustisches Zeichen. Wird während der Fahrt die Umgebungstemperatur zu hoch, soll die rote Warnlampe blinken. Sobald es im Raum dunkel wird, schaltet dein Roboter seine 2 Scheinwerfer ein. Wird es wieder hell, schalten die Scheinwerfer wieder aus.

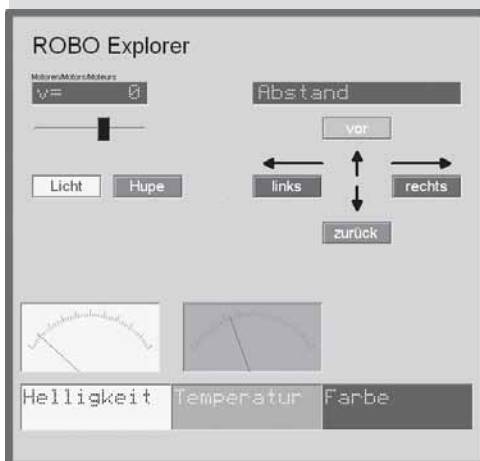


Fertiges Programm: **Explorer_3_TX.rpp**

■ Unser Explorer lässt sich als ferngesteuerter Roboter programmieren um unbekannte Welten zu erforschen. Dazu wird der ROBO TX Controller über die Bluetooth-Funkschnittstelle mit dem Rechner verbunden.

**Aufgabe 4 - ROBO Pro Level 3:**

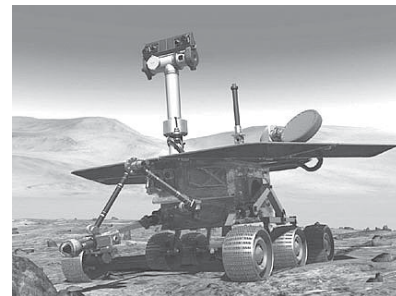
Für diese Aufgabe ist dein Computer die Steuerzentrale für eine Mars-Expedition. Auftrag ist, dass Messwerte von einer Marslandschaft zur Bodenstation übermittelt werden. Die Steuerung für deinen Roboter wird in ROBO Pro im Bedienfeld erzeugt (siehe ROBO Pro Hilfe Kapitel 9).



Dein Roboter soll so programmiert werden, dass er Messwerte von Bodenfarbe, Temperatur, Helligkeit und Hindernissen übermittelt. Gesteuert wird der Roboter von Hand über das Bedienfeld des Hauptprogramms in ROBO-Pro.

Fertiges Programm: **Explorer_4_TX.rpp**

Der Mars – Aufbruch zum vierten Planeten



RoboCup Junior Rescue Robot



Jetzt hast du alle Aufgaben erfüllt, dich in den Roboterbau und die Programmieretechnik eingearbeitet – nun könntest du die Früchte deiner Arbeit ernten und mit dem Modell „Rescue Robot“ in der Rescue-Liga des RoboCupJunior antreten.

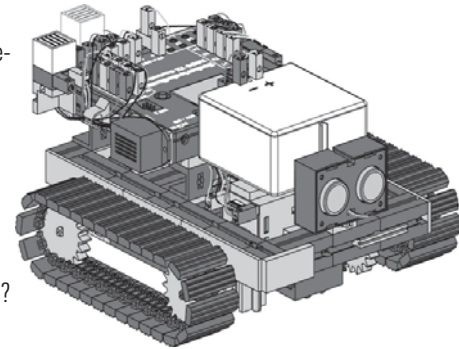
■ RoboCupJunior ist eine weltweite, projektorientierte Bildungsinitiative, die regionale, nationale und internationale Roboter-Veranstaltungen für junge Menschen fördert. Das Ziel besteht darin, Kindern und Jugendlichen Roboter und ihre Anwendung vorzustellen.

Anhand der Aufgabenstellung, die du aus dem Internet unter

<http://rcj.robocup.org>

beziehen kannst, fällt es dir sicher leicht, deinen Rescue Robot für diesen Wettbewerb zu programmieren.

Das linke Bild zeigt dir einen Parcours, den dein Roboter durchfahren muss. Auf dieser Strecke muss er verschiedene Aufgaben erledigen, z. B. einer Linie entlang fahren, verschiedenfarbige Figuren am Boden suchen oder durch eine Tür fahren usw. Wäre das nicht etwas für dich?



Wichtige Tipps

■ Der Spaß am Thema Robotik kann sehr schnell verloren gehen wenn der Roboter nicht so funktioniert wie man es gerne hätte.

Oft lassen sich mit einfachen Mitteln Fehler erkennen und beheben.

Kabel

Hier solltest du genau vorgehen. Zuerst werden die Kabel auf die vorgeschriebene Länge zugeschnitten und dann die Enden abisoliert und mit den Steckern fest verbunden. Überprüfe mit Hilfe eines Leuchtsteins (38216) mit Kugelstecklampe (37869) und dem Akku-Pack die Funktionsfähigkeit.

Stromversorgung

Oft ist ein fast leerer Akku Ursache von Fehlverhalten deines ROBO TX Explorers. Sinkt die Spannung unter 5 V, schaltet der ROBO TX Controller automatisch ab. Ein Fehlverhalten kann auch auftreten, wenn der Akku noch nicht so leer ist. Auch dann muss der Akku geladen werden.

Programmierung

Sind alle mechanischen Probleme gelöst und der Roboter will immer noch nicht so recht, liegt es oft an einer fehlerhaften Programmierung. Hier bietet dir ROBO Pro den Online-Modus, in dem du den Programmfluss auf dem Bildschirm verfolgen kannst. Hier findest du meist die kleinen Programmfehler, die sich eingeschlichen haben.

