

Smart Robots Pro

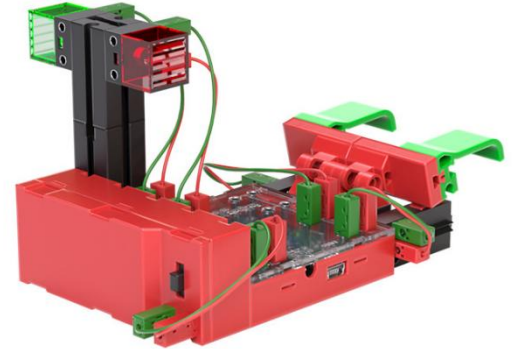
Getting Started

Für deine ersten Schritte in der Welt von Smart Robots Pro empfehlen wir dir, das Einstiegsvideo zu schauen. Bei diesem lernst du unter anderem, wie du dich mit dem Controller verbindest, wie man einen Schnittstellentest macht und die Controllerkonfiguration durchführt.

Danach kannst du schon beginnen zu programmieren.

Es lohnt sich, die Modelle in der hier aufgezeigten Reihenfolge zu bauen und zu programmieren. Nach und nach lernst du neue Programmierbausteine kennen, die du in späteren Aufgaben wieder benötigst.

Viel Spaß!



Karussell

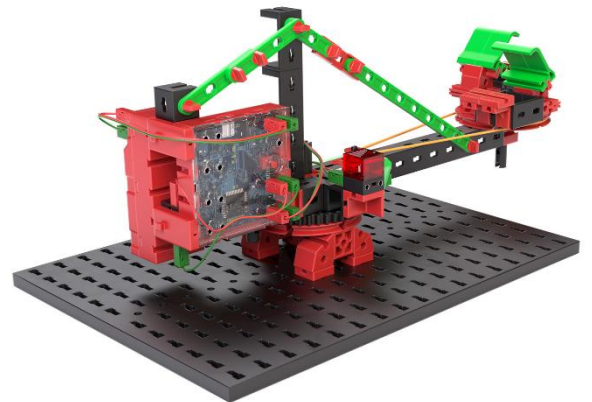
Svenja war letzte Woche auf einem Volksfest und konnte viele verschiedene Kirmesmaschinen testen. Am besten gefiel ihr das Karussell. Sie konnte gar nicht genug vom Fahren bekommen. Deshalb entscheidet Svenja, dass sie sich einfach ihr eigenes Karussell baut. Kannst du ihr beim Zusammenbau und der Programmierung helfen?

- Baue das Modell anhand der Bauanleitung auf.
- Schließe die Kabel entsprechend dem Schaltplan an.
- Starte die Software ROBO Pro Coding.
- Verbinde den BT-Smart Controller über die Bluetooth-Schnittstelle mit dem Computer oder einem anderen mobilen Endgerät.

Aufgabe 1:

Wenn der Controller verbunden ist, kannst du über den Schnittstellentest bereits die erste Testfahrt durchführen.

Schiebe dazu einfach den Regler des Motor 1 nach rechts und beobachte, wie sich das Karussell dreht. Du kannst das Karussell auch in die andere Richtung fahren lassen, indem du die Drehrichtung auf links änderst.

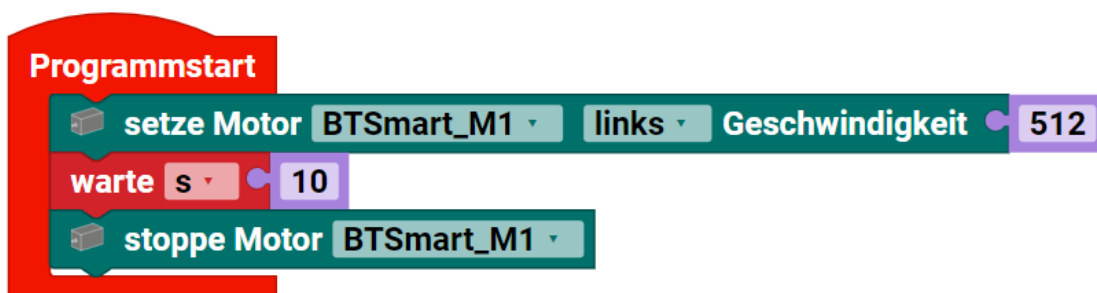


Tipp: Falls du dir nicht sicher bist, wo du den Schnittstellentest findest und wie er funktioniert, schaue dir am besten noch einmal das Einstiegsvideo an.

Das Karussell hat Lichter eingebaut, diese kannst du im Schnittstellentest über den Regler des Motor 2 ein- und ausschalten. Was passiert, wenn du die „Richtung“ des zweiten Ausgangs (M2) änderst?

Aufgabe 2:

Svenja hat im Internet bereits eine Vorlage eines Karussell-Programms gefunden. Sie ist sich aber nicht ganz sicher, was das Programm bewirkt. Kannst du erkennen, was der abgebildete Ablauf bei deinem Karussell bewirkt?



Ziehe die entsprechenden Blöcke in das Programmierfeld, sodass dieses Programm entsteht. Teste es anschließend.

Dazu muss in Robo Pro Coding die Controllerkonfiguration durchgeführt werden. Solltest du nicht wissen, wo du diese findest, schaue am besten noch einmal das Einstiegsvideo.

Lösung: Der Motor wird gestartet, läuft für 10 Sekunden und wird dann gestoppt.

Aufgabe 3:

Svenja hat Angst, dass sich die Passagiere durch das schnelle Anfahren den Kopf anstoßen könnten. Deshalb hat sie eine Idee: Das Karussell soll die ersten 3 Sekunden nur mit halber Geschwindigkeit (256) fahren und danach auf die maximale Geschwindigkeit (512) beschleunigen. Kannst du ihr helfen, so ein Programm zu schreiben?

Aufgabe 4:

Um besser erkennen zu können, ob sich das Karussell in der langsamen oder schnellen Stufe befindet, soll die grüne LED bei der langsamen Geschwindigkeit-

und die rote LED bei der schnellen Geschwindigkeit aufleuchten. Kannst du diese Funktion in deinem bisherigen Programm ergänzen?

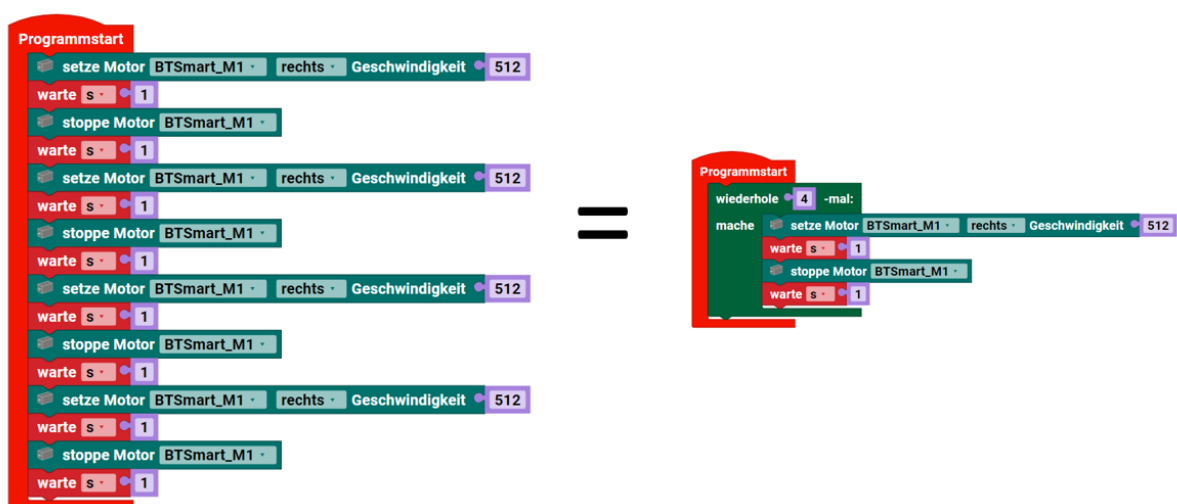
Tipp: LEDs funktionieren nur in eine „Richtung“. Das heißt die LED leuchtet, wenn der Strom von + zu – fließt. Sie leuchtet aber nicht, wenn der Strom von – zu + fließt. Diese Eigenschaft können wir uns zu Nutze machen, sodass wir mit einem Ausgang zwei Lampen ansteuern können. Dazu ändern wir in der Controllerkonfiguration die LED zu einem Motor, denn bei diesem kann man die Drehrichtung, also auch die Richtung, in die der Strom fließt, ändern. Ist der Motoreingang auf linksdrehend eingestellt, leuchtet nun die grüne LED und wenn er auf rechtsdrehend eingestellt ist, leuchtet die rote LED. Probiere diese Funktion erst im Schnittstellentest aus. Danach kannst du sie in dein Programm übernehmen.

Aufgabe 5 (fortgeschritten):

Bevor sich das Karussell in Bewegung versetzt, soll die rote LED 3x blinken. Nach der schnellen Phase wird die Geschwindigkeit wieder für drei Sekunden halbiert. Danach ändert sich die Drehrichtung des Karussells, ebenfalls mit einer 3-sekündigen langsamen- und 7-sekündigen schnellen Phase. Die Fahrt endet mit einer erneuten 3-sekündigen langsamen Phase.

Bei der langsamen Geschwindigkeit leuchtet die rote und bei der schnellen Geschwindigkeit die grüne LED.

Tipp: Beim Programmieren werden oft aneinanderhängende Befehle mehrmals nacheinander benötigt. Um nicht endlos viele Blöcke programmieren zu müssen, kann man auch sogenannte Schleifen verwenden. Hier siehst du zwei Programme, die das gleiche bewirken, nämlich, dass der Ausgang M1 4x nacheinander an- und ausgeschaltet wird (oder blinkt).



Beispiellösungen zum Modell findest du in Robo Pro Coding.

Useless Machine

Niks kleiner Bruder Ole schnappt sich manchmal Niks Süßigkeiten. Deshalb beschließt er, sich eine kleine Box zu bauen, die sich nur ganz kurz öffnet. Da Ole noch nicht so schnell ist, kann sich Nik die Süßigkeiten aus der Box nehmen, Ole aber nicht. Außerdem macht es einfach Spaß, die Box zu bedienen und zu beobachten. Hilfst du Nik, eine solche Box zu bauen und zu programmieren?



- Baue das Modell anhand der Bauanleitung auf.
- Schließe die Kabel entsprechend dem Schaltplan an.
- Starte die Software ROBO Pro Coding.
- Verbinde den BT-Smart Controller über die Bluetooth- oder USB-Schnittstelle mit dem Computer oder einem anderen mobilen Endgerät.
- Prüfe mithilfe des Schnittstellentests, ob alle elektronischen Bauteile richtig angeschlossen sind.
- Führe anschließend die Controllerkonfiguration durch.

Aufgabe 1:

Mit dem Schnittstellentest kannst du alle Motoren auf die Ausgangsposition verfahren. Überprüfe dabei, ob die Taster geöffnet oder geschlossen sind. Versuche, über den Schnittstellentest die Schnauze der Box zu öffnen, den Betätigungshebel aus- und einfahren zu lassen und anschließend die Schnauze wieder zu schließen.

Tipp: Es kann sein, dass der Betätigungshebel den Schalter mit der Kraft des Motors nicht sofort ausschaltet. Betätige dann den Schalter mit dem schwarzen Drehgelenk einfach 10x von Hand ganz nach vorne und hinten und überprüfe, ob deine Batterie noch genügend Spannung liefert. Vermeide, dass die Motoren über längeren Zeitraum auf Block fahren, also dass sie versuchen sich zu drehen, das aber nicht können, weil sie anschlagen.

Aufgabe 2:

Erstelle nun ein Programm für die Useless Box:

Wenn der Taster I4 betätigt ist, soll sich die Schnauze der Box öffnen und nach 0,6 Sekunden anhalten. Danach fährt der Betätigungshebel aus und öffnet den Taster I4 wieder. Anschließend fährt der Betätigungshebel wieder zur Ausgangsposition zurück, sodass der Taster I2 geschlossen ist. Dann schließt die Schnauze wieder. Für dieses Programm kannst du den **warte bis** Block nutzen. Dieser Stoppt ein Programm, bis die angehängte Bedingung erfüllt ist und macht dann erst weiter.

Beispiellösungen zum Modell findest du in Robo Pro Coding.

Tanzroboter

Clara liebt den Karneval und Umzüge jeglicher Art. Allerdings werden ihre Hände von dem vielen Winken irgendwann immer so schwer. Damit sie trotzdem die viele schönen Wägen beim Stadtumzug grüßen kann, baut sie einfach einen Roboter, der für und mit ihr winkt.

- Baue das Modell anhand der Bauanleitung auf.
- Schließe die Kabel entsprechend dem Schaltplan an.
- Starte die Software ROBO Pro Coding.
- Verbinde den BT-Smart Controller über die Bluetooth- oder USB-Schnittstelle mit dem Computer oder einem anderen mobilen Endgerät.
- Prüfe mithilfe des Schnittstellentests, ob alle elektronischen Bauteile richtig angeschlossen sind.
- Führe anschließend die Controllerkonfiguration durch.



Aufgabe 1:

Verfahre mit dem Schnittstellentest die Stellung der Arme so, dass der linke Arm senkrecht nach unten zeigt.

Programmiere den Roboter so, dass er dauerhaft winkt, indem er in einer Schleife für 2 Sekunden die Arme mit der Geschwindigkeit 100 bewegt, dann eine Sekunde die Arme stehen lässt, und wieder für zwei Sekunden die Arme in die andere Richtung bewegt. Nach den zwei Sekunden bleiben die Arme ebenfalls wieder eine Sekunde stehen.

Achte darauf, dass du vor jedem Programmstart die Arme wieder mit dem Schnittstellentest in die Ausgangslage beförderst und dass die Motoren nicht blockieren.

Du kannst die Geschwindigkeit beliebig variieren – passe dann auch die Dauer an.

Aufgabe 2:

Um noch mehr Aufmerksamkeit auf ihre Tanzroboter zu ziehen, überlegt sich Clara den Roboter nach jedem zweiten Winken nicken zu lassen, indem sich die Nockenscheibe nach jedem zweiten Winken für 2 Sekunden mit der Geschwindigkeit 350 linksherum dreht. Kannst du ihr helfen, ein entsprechendes Programm zu schreiben?

Achte darauf, dass du vor jedem Programmstart die Arme wieder mit dem Schnittstellentest in die Ausgangslage beförderst und dass die Motoren nicht blockieren.

Beispiellösungen zum Modell findest du in Robo Pro Coding.

Drivebot

- Baue das Modell anhand der Bauanleitung auf.
- Schließe die Kabel entsprechend dem Schaltplan an.
- Starte die Software ROBO Pro Coding.
- Verbinde den BT-Smart Controller über die Bluetooth -Schnittstelle mit dem Computer oder einem anderen mobilen Endgerät.
- Prüfe mithilfe des Schnittstellentests, ob alle elektronischen Bauteile richtig angeschlossen sind.
- Führe anschließend die Controllerkonfiguration durch.



Aufgabe 1:

Programmiere den Roboter so, dass er nach Programmstart geradeausfährt. Trifft der Roboter auf ein Hindernis, also wird einer der beiden Öffner betätigt, fährt der Roboter für eine Sekunde nach hinten und dreht sich anschließend, indem ein Rad für eine Sekunde vorwärts und das andere rückwärtsfährt. Die Geschwindigkeit der Ränder beim Drehen beträgt für zwei Sekunden lang 200. Danach flitzt der Roboter wieder geradeaus weiter.

Taster können als Öffner oder Schließer verwendet werden. Der Zustand eines Schließers ist im unbestätigten Zustand 0 und im betätigten Zustand 1. Beim Öffner ist es andersherum: Unbetätigt ist der Zustand 1 und betätigt 0.

Tipp: Achte darauf, dass die Gelenke der Feuertaster ganz nach oben geschoben sind, sodass die Taster zuverlässig betätigt werden.

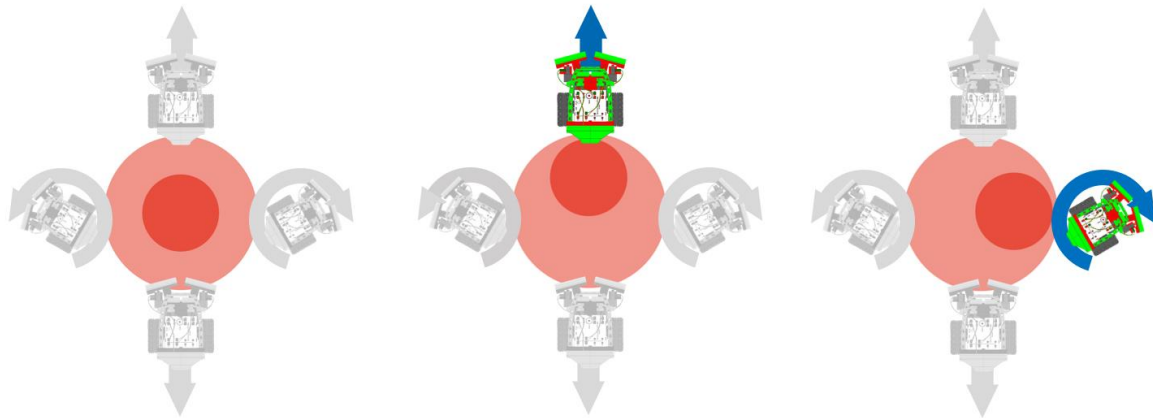
Aufgabe 2:

Ergänze das Programm so, dass der Roboter seinen Pfad nach links korrigiert, wenn er rechts an ein Hindernis stößt und andersherum seinen Pfad nach rechts korrigiert, wenn er links an ein Hindernis stößt.

Beispiellösungen zum Modell findest du in Robo Pro Coding.

Joystick – Fernsteuerung

Nun schreiben wir ein kleines Programm, mit dem wir unsere Fahrroboter ganz einfach fernsteuern können. Das Programm beinhaltet einen digitalen Joystick. Dieser Joystick kann auf deinem Touch-Display oder auf dem Computer hin- und hergezogen werden. Unser Fahrroboter soll dann in die Richtung fahren, in die wir den Joystick ziehen. Zum Beispiel fährt er gerade aus, wenn wir den Joystick nach oben ziehen und er dreht sich, wenn wir ihn auf eine Seite ziehen.



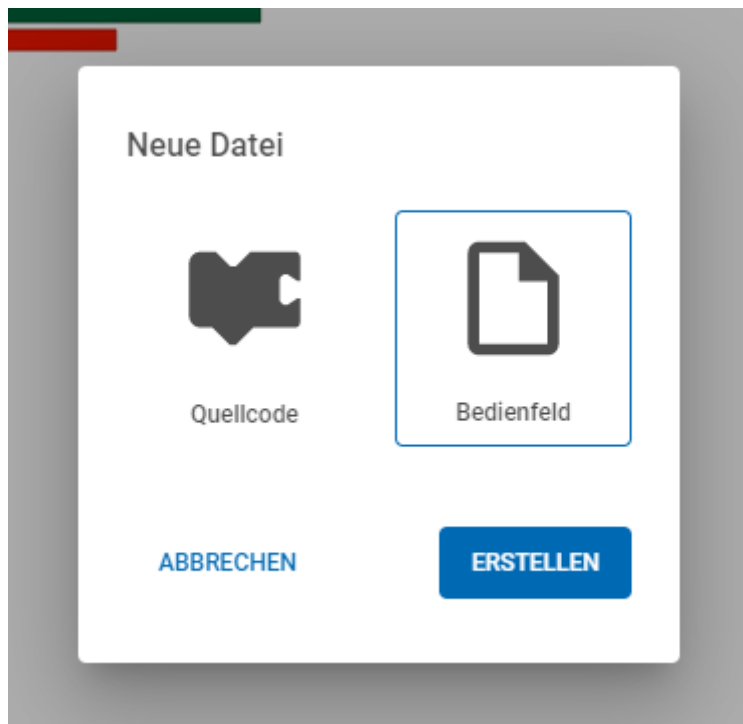
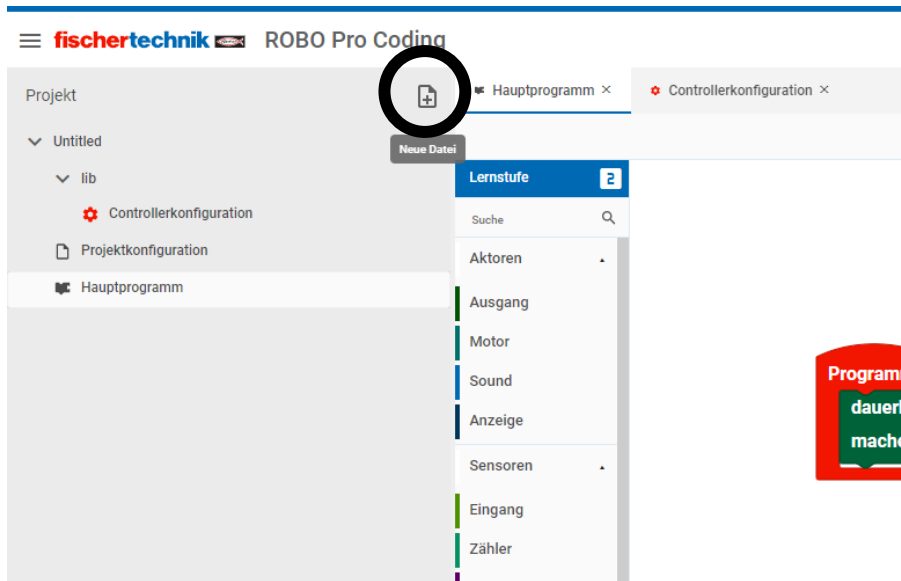
Mit dem programmierten Joystick kannst du den Drivebot oder den Teach-In Roboter fernsteuern.

Wenn du willst, kannst du auch zunächst testen, was wir programmieren werden. Öffne dazu einfach das Beispielprogramm „BTSmart_RC_Joystick_5“ und teste das Programm.

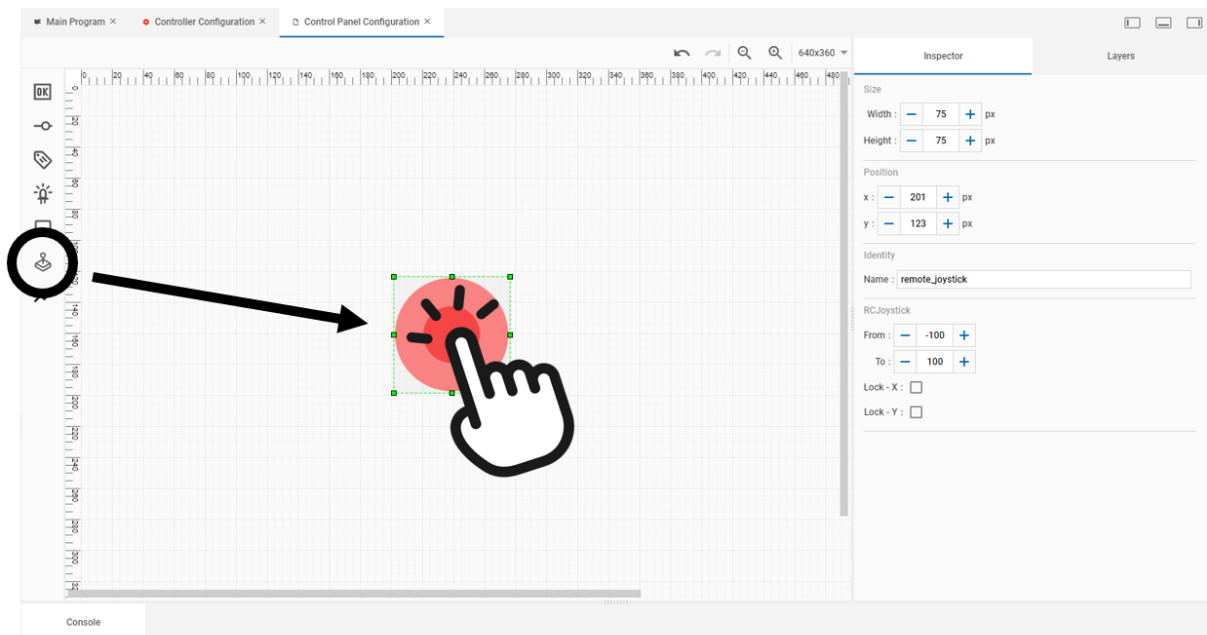
Im folgenden Absatz lernst du einige neue Funktionen von Robo Pro Coding kennen, die wir auch für weitere Roboter benötigen.

- Baue ein beliebiges Fahrroboter-Modell auf
- Schließe die Kabel entsprechend dem Schaltplan an.
- Starte die Software ROBO Pro Coding.
- Verbinde den BT-Smart Controller über die Bluetooth -Schnittstelle mit dem Computer oder einem anderen mobilen Endgerät.
- Prüfe mithilfe des Schnittstellentests, ob alle elektronischen Bauteile richtig angeschlossen sind.
- Führe anschließend die Controllerkonfiguration durch.

Bevor wir anfangen das Programm zu schreiben, benötigen wir noch eine zusätzliche Schaltfläche für den Joystick. Klicke dazu auf **neue Datei** und anschließend auf **Bedienfeld**. Durch das Drücken auf **Erstellen**, wird dieses hinzugefügt. Diese Schritte siehst du in den folgenden Bildern:



Nun erscheint eine neue leere Oberfläche. Diese Oberfläche werden wir auf unserem Computer oder Tablet sehen, wenn wir das Programm starten. Zunächst ziehen wir auf diese Oberfläche den sogenannten **RCJoystick**:



Über die grünen Punkte an den Ecken kannst du den Joystick größer oder kleiner ziehen. Nun haben wir schon einen Joystick erstellt.

Teste ihn aus, indem du deinen Controller verbindest und das Programm in Robo Pro Coding startest.

Du siehst: Mit dem Programmstart öffnet sich das zuvor gezeichnete Fenster und du kannst den inneren Teil des Joysticks hin und her ziehen.

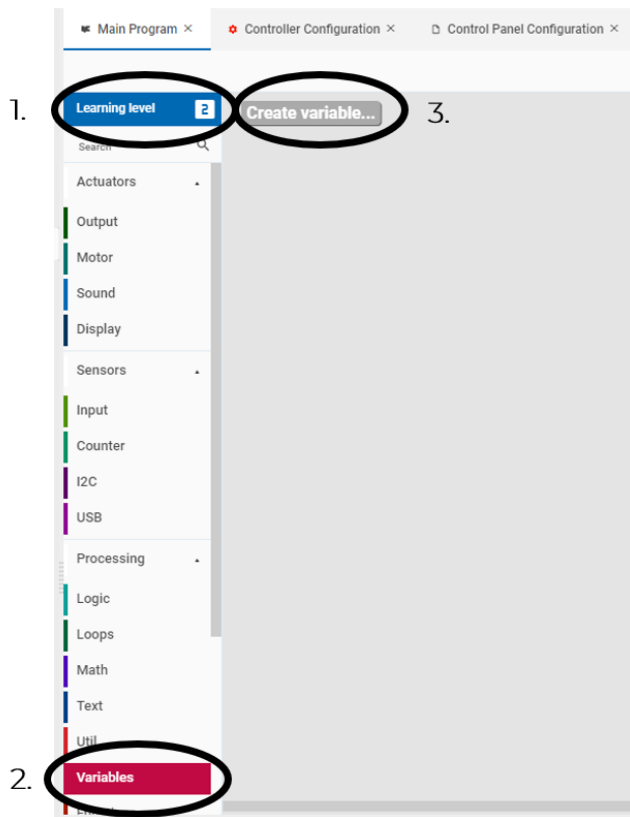


Mithilfe des Joysticks können wir nun ein Programm schreiben, das den Fahrroboter in die entsprechende Joystick-Richtung fahren lässt. Dazu wechseln wir wieder in die Programmierumgebung.

Variablen

Variablen sind Speicher-Bausteine. In diese Bausteine können wir Texte oder Zahlen einspeichern und später in unserem Programm weiterverwenden. Wir verwenden nun Variablen für das Einspeichern der Position, an der sich unser Joystick befindet.

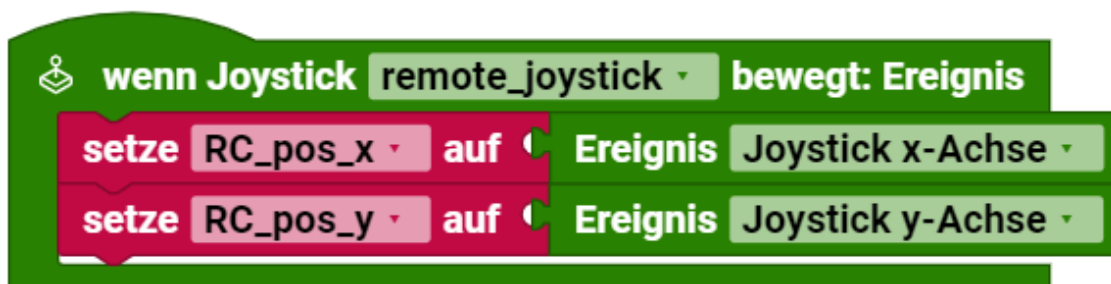
Dazu wechseln wir zunächst in die Lernstufe 2 und erstellen unter Variablen zwei neue Variablen:



Diese Variablen benennen wir „RC_pos_X“ und „RC_pos_Y“. Unter dem Reiter Variablen können wir nun jederzeit auf diese Speicher zugreifen.

Unter dem Feld Fernbedienung gibt es einen Joystick-Event-Block, den wir dazu verwenden können, immer die aktuellen Werte der Joystick-Position in unsere Variablen zu schreiben.

Erstelle mit diesen Blöcken ein Programm, das bei jeder Änderung der Joystick-Position die aktuelle Position in die jeweilige Variable speichert:



Bei dem Joystick ändern sich die Y-Werte, indem er nach oben oder nach unten gezogen wird. Die X-Werte ändern sich, wenn er nach links- oder rechts gezogen wird.

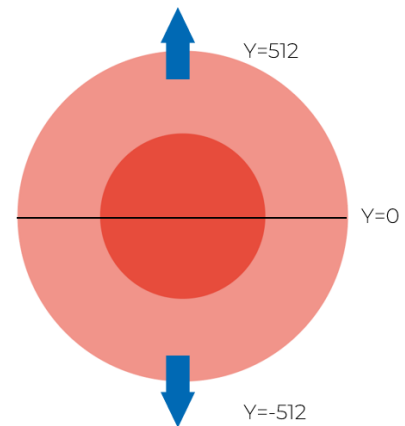
In seiner Ausgangsposition hat er die Werte $X=0$ und $Y=0$.

Je weiter man den Joystick nach oben zieht, desto größer werden die Y-Werte.

Je weiter man den Joystick nach unten zieht, desto kleiner werden die Y-Werte (sie werden zu negativen Werten).

Aufgabe 1:

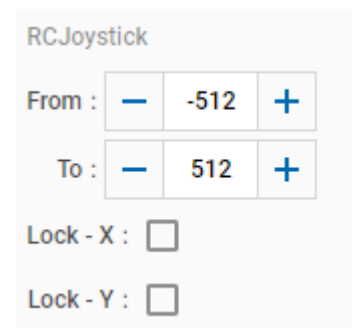
Schreibe ein Programm, das beide Motoren auf volle Geschwindigkeit setzt, wenn die Variable `RC_pos_Y` größer als 0 ist. Ist die Variable kleiner als 0, soll dein Fahrroboter rückwärts fahren, indem beide Motoren in die andere Richtung drehen. Um zu testen, ob die Variable größer oder kleiner als 0 ist, kannst du unter dem Reiter Logik einen Vergleichsblock – und aus dem Reiter Mathe eine Zahl ziehen.



Falls du dir nicht sicher bist, wie das geht, kannst du in den Beispielprogrammen nachschauen.

Aufgabe 2:

Im nächsten Schritt steuern wir auch die Geschwindigkeit des Fahrroboters mithilfe des Joysticks. Dazu wechseln wir noch einmal in die Konfiguration des Bedienfelds und wählen den gezeichneten Joystick aus. In dem rechten Fenster finden wir nun Einstellungsmöglichkeiten für den Joystick. Um die Geschwindigkeit einfacher regeln zu können, erweitern wir den Bereich von -512 bis 512.



Schreibe nun ein Programm, bei dem der Fahrroboter immer so schnell fährt, wie der Wert der Variable `RC_pos_Y` groß ist.

Tipp: Bisher haben wir die Geschwindigkeit der Motor-Blöcke immer mit festen Werten definiert. Da wir aber in unseren Variablen Zahlenwerte gespeichert haben, können wir auch die Variable an den Block anhängen. Ziehe dazu einfach den Variablen-Block über die Zahl. Diese wird automatisch ersetzt.

Tipp: Bei der Verwendung von Variablen kann der Motor auch negative Zahlenwerte annehmen. Das heißt ist ein Wert kleiner als 0, also z.B. -256, fährt er automatisch rückwärts und man muss die Drehrichtung des Motors nicht extra ändern. Du kannst dieses Programm also theoretisch ohne WENN-Block schreiben.

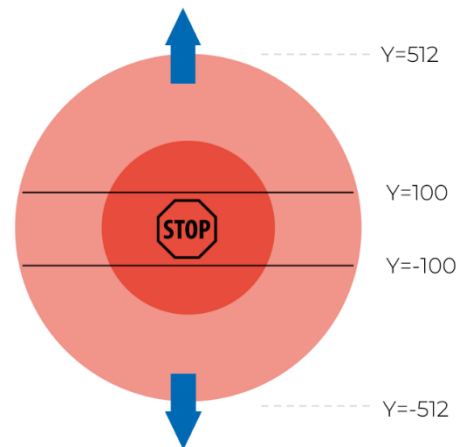
Aufgabe 3:

Da die Spannung bei sehr kleinen Werten nicht ausreicht, um den Fahrroboter zu bewegen, führen wir weitere Zonen ein. Nämlich soll der Fahrroboter erst losfahren, wenn der Wert der Y-Variable größer bzw. kleiner als 100 / -100 ist.

Dazu benötigen wir noch eine weitere Abfrage, die wir neu kennen lernen: Die ODER-Verknüpfung:

Bei den Falls Blöcken haben wir bisher nur eine Bedingung verglichen. Mit den ODER-Blöcken können wir auch mehrere Zustände abfragen. Bei dem Oder-Block muss eine von mehreren Abfragen stimmen, damit die Anweisung ausgeführt wird. Bei dem UND-Block müssen alle Abfragen stimmen, damit die Anweisung ausgeführt wird.

Erweitere dein Programm aus Aufgabe 2 so, dass der Fahrroboter erst losfährt, wenn die Variable RC_pos_Y größer als 100 ODER kleiner als -100 ist.



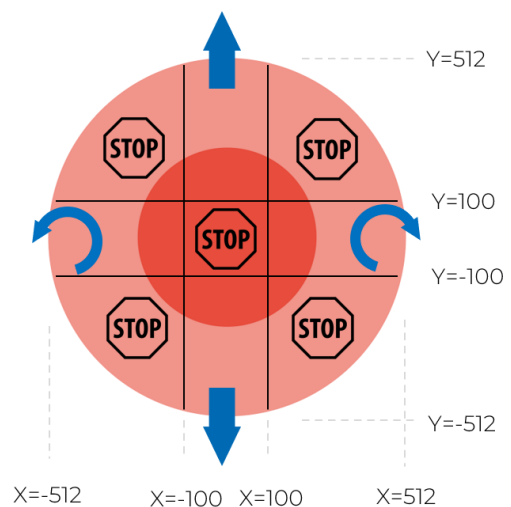
Aufgabe 4 (fortgeschritten):

Nun führen wir weitere Zonen ein, sodass wir den Fahrroboter auch drehen können. Dazu verwenden wir jetzt zusätzlich UND-Verknüpfungen.

Erweitere dein Programm so, dass der Fahrroboter mit den Geschwindigkeiten der Variable RC_pos_Y fährt, wenn folgende Bedingungen erfüllt sind:

RC_pos_X ist kleiner als 100 UND größer als -100.

Gleichzeitig (UND) muss RC_pos_Y größer als 100 ODER kleiner als -100 sein.



➔ Diese Bedingungen sollen die Vor- bzw. Rückwärtsfahrt steuern.

Wenn RC_pos_Y kleiner als 100 UND größer als -100 ist UND gleichzeitig RC_pos_X größer 100 ODER kleiner als -100 ist, sollen sich die Motoren mit der Geschwindigkeit RC_pos_X drehen. Drehe dabei die Laufrichtung eines Motors um.

➔ Diese Bedingungen sollen das Rechts- bzw. Linksdrehen steuern.

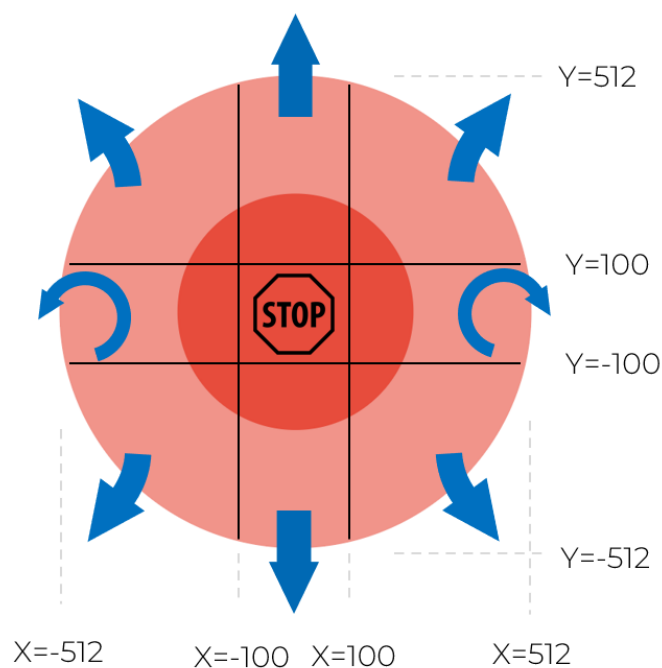
Aufgabe 5: (Expertenaufgabe):

Zieht man den Joystick schräg, hält der Roboter aktuell noch an.

Schreibe nun ein Programm, dass der Roboter eine große Kurve fährt, wenn der Joystick in einem der vier restlichen Felder steht. Ergänze dazu das Programm aus Aufgabe 4. Um das umzusetzen, sollen die Motoren in dieselbe Richtung fahren, allerdings soll dabei ein Motor langsamer, als der andere fahren.

Es kann hilfreich sein, dir ein paar weitere Variablen zu erstellen. Das ist aber nicht zwingen notwendig. Versuche gerne dein individuelles Steuerprogramm zu schreiben.

Folgende Darstellung kann dir dabei helfen die restlichen Zonen zu ermitteln:



Weiter geht's!

Gratuliere! Du bist nun ein erfahrener Programmierer. Mach dir keinen Kopf, wenn du die fortgeschrittenen oder Experten-Aufgaben übersprungen hast. Durch die ersten Aufgaben der Fernsteuerung hast du trotzdem viele neue Programmierblöcke kennengelernt, mit denen wir weitere Programme erstellen können.

Falls du vor der nächsten Programmierung noch eine Pause brauchst, verwende doch den Teach-In-Roboter, für diesen nutzen wir einfach ein vorgefertigtes Programm:

Teach-In Roboter

Teach-In Roboter kann man mithilfe von Eingaben direkt am Roboter beibringen, was sie tun sollen. Sie sind kompliziert zu programmieren, deshalb kannst du direkt das Beispielprogramm in Robo Pro Coding öffnen.



- Baue das Modell anhand der Bauanleitung auf.
- Schließe die Kabel entsprechend dem Schaltplan an.
- Starte die Software ROBO Pro Coding.
- Verbinde den BT-Smart Controller über die Bluetooth -Schnittstelle mit dem Computer oder einem anderen mobilen Endgerät.
- Prüfe mithilfe des Schnittstellentests, ob alle elektronischen Bauteile richtig angeschlossen sind.
- Führe anschließend die Controllerkonfiguration durch.

Aufgabe 1:

Baue dir einen Hindernisparcours. Versuche anschließend diesen mit den Teach-In-Befehlen zu meistern. Die Eingabe des Tasters funktioniert zuverlässiger, wenn du sie mehrere Sekunden pro Befehl betätigst.

Taster links: Linksdrehung (90°)

Taster rechts: Rechtsdrehung (90°)

Beide Taster gleichzeitig: Geradeausfahrt

Vorprogrammiert fährt der Roboter nach 5 Befehlen los. Du kannst die Anzahl aber auch ändern, indem du den Wert der Variable `list_move < 5` änderst. Außerdem kannst du schauen was passiert, wenn du die Geschwindigkeiten änderst.

Alarmanlage

In Deutschland wurde 2022 in mehr als 65.000 Wohnungen eingebrochen. Das entspricht 178 Einbrüchen pro Tag. Gut, wer sich zu helfen weiß und sich mit Technik schützen kann. Um kennenzulernen, wie eine Alarmanlage funktioniert, bauen wir selbst eine.



- Baue das Modell anhand der Bauanleitung auf.
- Schließe die Kabel entsprechend dem Schaltplan an.
- Starte die Software ROBO Pro Coding.
- Verbinde den BT-Smart Controller über die Bluetooth- oder USB-Schnittstelle mit dem Computer oder einem anderen mobilen Endgerät.
- Prüfe mithilfe des Schnittstellentests, ob alle elektronischen Bauteile richtig angeschlossen sind.
- Führe anschließend die Controllerkonfiguration durch.

Aufgabe 1:

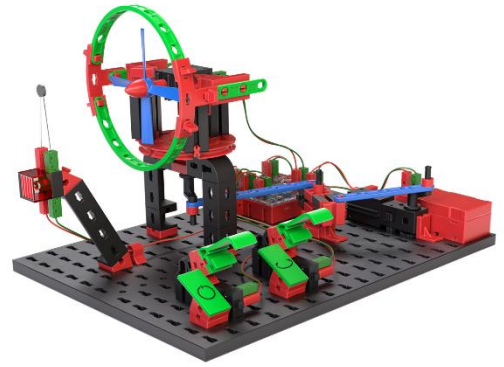
Programmiere die Alarmanlage wie folgt:

- Wenn der Hauptschalter nicht betätigt ist, soll nichts passieren: So kannst du und deine Familie ganz normal die Türe öffnen und schließen.
- Ist der Hauptschalter betätigt und die Türe geschlossen, sodass der Reed-Kontakt den Status 1 hat, leuchtet die grüne LED.
- Ist der Hauptschalter betätigt, aber die Türe offen, blinkt die Rote LED und der Ratterer erzeugt alle zwei Sekunden ein Geräusch für eine Sekunde.
- Wird anschließend die Türe wieder geschlossen, soll der Alarm trotzdem noch weiter ausgelöst sein.
- Erst wenn der Hauptschalter wieder ausgeschaltet ist, stoppt auch der Alarm.

Beispiellösungen zum Modell findest du in Robo Pro Coding.

Ventilator

Ventilatoren sind sehr einfache, aber nützliche Maschinen. Einen einfachen Ventilator kann man ein- und ausschalten, mache auch hin und her bewegen lassen. Wir wollen uns nun einen intelligenten Ventilator bauen, der diese beiden Funktionen automatisch startet.



- Baue das Modell anhand der Bauanleitung auf.
- Schließe die Kabel entsprechend dem Schaltplan an.
- Starte die Software ROBO Pro Coding.
- Verbinde den BT-Smart Controller über die Bluetooth- oder USB-Schnittstelle mit dem Computer oder einem anderen mobilen Endgerät.
- Prüfe mithilfe des Schnittstellentests, ob alle elektronischen Bauteile richtig angeschlossen sind.
- Führe anschließend die Controllerkonfiguration durch.

Aufgabe 1:

Programmiere den Ventilator so, dass man ihn über die beiden Schalter drehen- und hin und herbewegen lassen kann. Der Ventilator soll sich nur hin und herbewegen, wenn er sich auch dreht.

Aufgabe 2:

Mache die Ventilatorsteuerung intelligent, indem der Ventilator automatisch anspringt, sobald der NTC-Widerstand Wert kleiner als 1300 ist. Wird der Wert überschritten, kann der Ventilator trotzdem über die Schalter gesteuert werden.

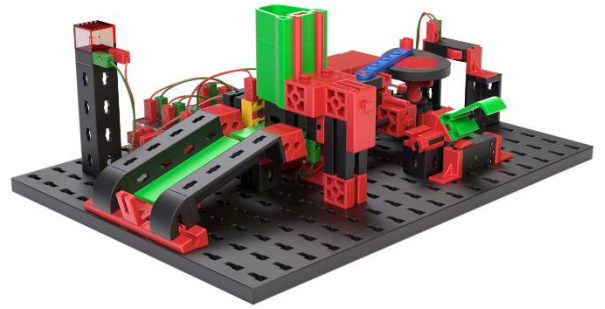
Hierfür lassen wir den Controller rechnen. Um das zu tun, musst du in die Lernstufe 2 wechseln. Hier vergleichen wir in einem Logik-Block ob der Sensorwert kleiner ($<$) als ein angegebener Wert ist.

Tipp: Gegebenenfalls kannst du die Werte anpassen. Schaue im Schnittstellentest, welchen Wert der NTC-Widerstand im „kalten Zustand“ ausgibt. Du kannst die Wärme simulieren, indem du mit den Fingern den NTC-Widerstand anfasst.

Beispiellösungen zum Modell findest du in Robo Pro Coding.

Bauteilspender

- Baue das Modell anhand der Bauanleitung auf.
- Schließe die Kabel entsprechend dem Schaltplan an.
- Starte die Software ROBO Pro Coding.
- Verbinde den BT-Smart Controller über die Bluetooth- oder USB-Schnittstelle mit dem Computer oder einem anderen mobilen Endgerät.
- Prüfe mithilfe des Schnittstellentests, ob alle elektronischen Bauteile richtig angeschlossen sind.
- Führe anschließend die Controllerkonfiguration durch.



Aufgabe 1:

Programmiere den Bauteilspender so, dass er dir mit jedem Tastendruck (I1) ein Bauteil spendet. Wenn kein Bauteil mehr im Magazin erkannt wird, geht die Signal-LED an und der Spender reagiert nicht mehr auf den Tastendruck.

Aufgabe 2:

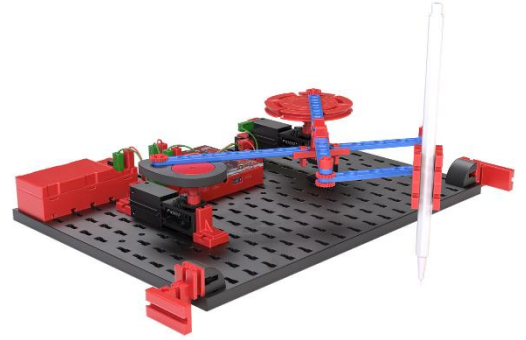
Mache aus dem Bauteilspender einen Bauteilzähler. Nach der Betätigung des Triggers stößt der Zähler Bauteil für Bauteil aus dem Magazin aus, bis dieses leer ist. Die Bauteile sollen parallel gezählt- und das Ergebnis als Text ausgegeben werden.

Hierfür kannst du Variablen verwenden. Du kannst eine Variable mit dem Namen „count“ erstellen und bei Programmstart auf 0 setzen. Mit jedem Baustein, der ausgeworfen wird, kannst du zur Variable +1 rechnen.

Beispiellösungen zum Modell findest du in Robo Pro Coding.

Malroboter

- Baue das Modell anhand der Bauanleitung auf.
- Schließe die Kabel entsprechend dem Schaltplan an.
- Starte die Software ROBO Pro Coding.
- Verbinde den BT-Smart Controller über die Bluetooth- oder USB-Schnittstelle mit dem Computer oder einem anderen mobilen Endgerät.
- Prüfe mithilfe des Schnittstellentests, ob alle elektronischen Bauteile richtig angeschlossen sind.



Aufgabe 1:

Schnapp dir ein Blatt und klebe es mit zwei Klebestreifen auf den Tisch. Nimm einen Stift, der in die Stifthalterung des Malroboters passt und befestige ihn in der Halterung. Richte dann den Malroboter zum Papier aus.

Im Schnittstellentest kannst du nun unzählig viele Muster erstellen, indem du die Motorgeschwindigkeiten und Drehrichtungen änderst. Wenn dir ein Muster besonders gut gefällt, kannst du auch ein Programm mit den gewählten Parametern erstellen, sodass du es immer wieder zeichnen lassen kannst. Lass den Malroboter eine Weile zeichnen, bis du ein Muster erkennen kannst.

Verwende immer ein Papier bei der Benutzung des Malroboters und gehe sicher, dass das Papier keine Farbe durchlässt.

Aufgabe 2:

Ein weiterer Einfluss für die Entstehung der Muster sind die Drehachsen. Versuche was passiert, wenn du Motorpositionen änderst?

Ballspiel

- Baue das Modell anhand der Bauanleitung auf.
- Schließe die Kabel entsprechend dem Schaltplan an.
- Starte die Software ROBO Pro Coding.
- Verbinde den BT-Smart Controller über die Bluetooth- oder USB-Schnittstelle mit dem Computer oder einem anderen mobilen Endgerät.
- Prüfe mithilfe des Schnittstellentests, ob alle elektronischen Bauteile richtig angeschlossen sind.
- Führe anschließend die Controllerkonfiguration durch.

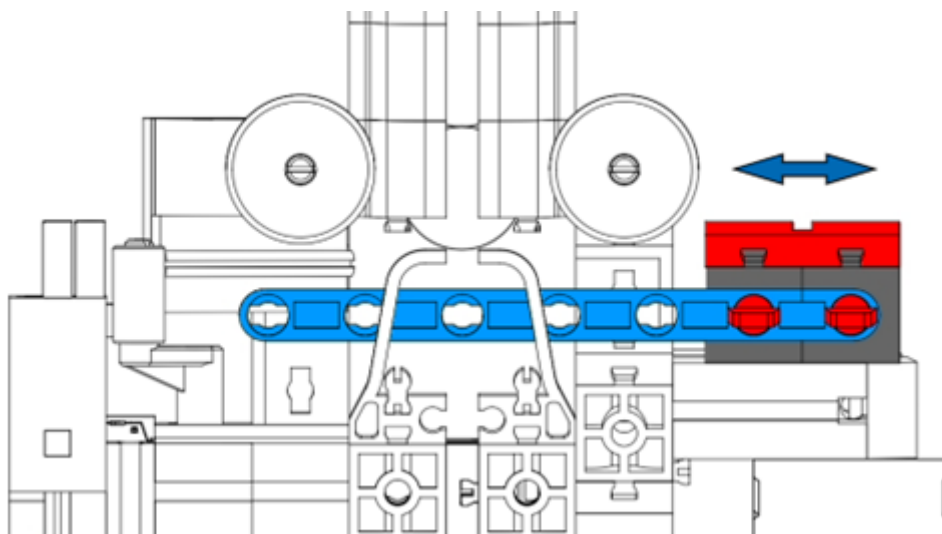


Aufgabe 1:

Erstelle ein Programm, das bei jedem Tastendruck einen Wurf auslöst. Achte auf die korrekte Drehrichtung bei dem Wurf. Breche das Programm ab, wenn die Drehrichtung nicht stimmt und der Motor blockiert.

Im Schnittstellentest kannst du die Neigung des Auswurfs durch Steuerung des zweiten Motors anpassen. Wähle zunächst eine flache Steigung.

Die Stärke des Wurfmechanismus kannst du einstellen. Sollte der Ball nicht weit genug fliegen, kannst du zunächst die Strebe nach links und rechts verschieben. Achte darauf, dass die Strebe vom drehenden Hebel beim Schuss zurückgedrückt wird. Sollte der Hebel nicht stark genug sein, die Strebe nach hinten zu drücken, überprüfe die Spannung der Batterie und ersetze sie gegebenenfalls.



Schiffschaukel

- Baue das Modell anhand der Bauanleitung auf.
- Schließe die Kabel entsprechend dem Schaltplan an.
- Starte die Software ROBO Pro Coding.
- Verbinde den BT-Smart Controller über die Bluetooth- oder USB-Schnittstelle mit dem Computer oder einem anderen mobilen Endgerät.
- Prüfe mithilfe des Schnittstellentests, ob alle elektronischen Bauteile richtig angeschlossen sind.
- Führe anschließend die Controllerkonfiguration durch.



Achte beim Bau des Modells darauf, dass das Schiff gerade so auf dem Reifen aufliegt. Schwenke dann die Schaukel von Hand und stelle mithilfe des Schnittstellentests sicher, dass der Reed-Kontakt auslöst, wenn der Magnet an der Schiffschaukel den Reed-Kontakt passiert.

Aufgabe 1:

Erstelle ein Programm, das durch die Reifendrehung die Schiffschaukel beschleunigt. Sobald der Reed-Kontakt geschlossen wird, soll sich der Reifen für ca. 0,2 Sekunden in die andere Richtung drehen. Anschließend ändert sich erneut die Drehrichtung, bis der Reed-Kontakt wieder geschlossen wird.

Das Programm soll erst dann starten, wenn der Hauptschalter (I1) am Modell geschlossen ist. Sobald der Hauptschalter wieder geöffnet wird, soll die Schiffschaukel wieder ausschlagen, noch 1x den Reed Kontakt betätigen, in die andere Richtung zurückschwingen und dann stoppen, sodass das Schiff senkrecht am Reifen ansteht.

Tip: Damit der Reifen die Schiffschaukel zuverlässig beschleunigt, kannst du ihn mit dem Motor nach links oder rechts schieben. Der Reifen sollte im ausgeschwungenen Zustand immer die Schiffschaukel berühren.

Aufgabe 2:

Erstelle eine Variable mit dem Namen „rounds“ und speichere in ihr den Wert 4. Bevor die Fahrt startet, sollen die LEDs abwechselnd blinken, nämlich so oft, wie der Wert rounds groß ist. Anschließend schwingt die Schiffschaukel ebenfalls so oft, wie der Wert round groß ist.

Schaffst du es der Variable einen zufälligen Wert zwischen 1 und 6 zuzuweisen?

Beispiellösungen zum Modell findest du in Robo Pro Coding.