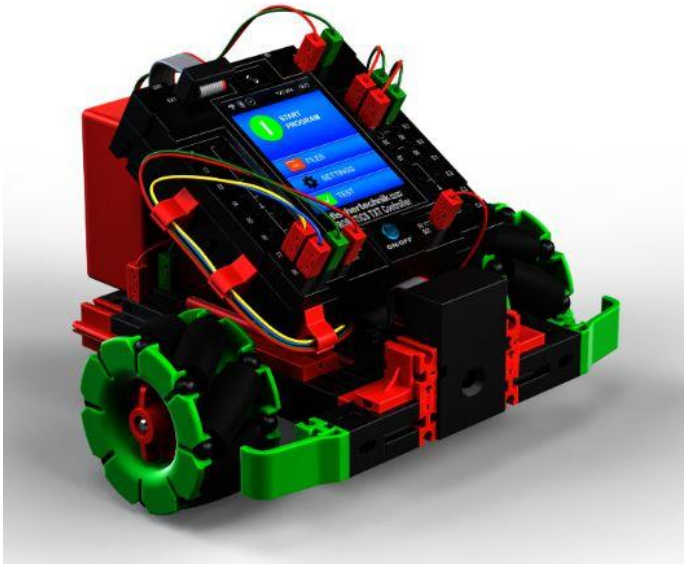
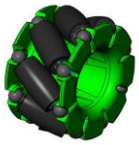
Advanced 2-Radroboter



Mit diesem Modell erarbeitest du dir weitere Grundlagen der Programmierung. Die verbauten fischertechnik Bauteile kennst du ja schon von den vorherigen Aufgaben. Das Modell steuerst du sowohl mit dem IR-Spursensor als auch mit dem Gestensensor.

Im Modell verbaute Aktoren, Sensoren und technisches Zubehör:

Mini Motor	Getriebe	Gestensensor	Omniwheelrad	IR-Spursensor
				

Die Erklärung zu den Bauteilen findest du auf der Einleitungsseite.

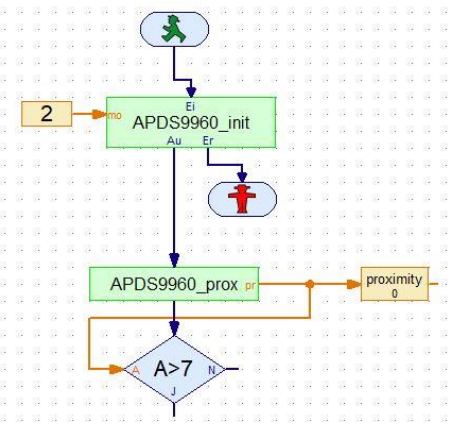
Das Modell „Advanced 2-Radroboter“ gliedert sich in 4 Programmieraufgaben:

Aufgabe 1 Advanced_2-wheeled_robot_1.rpp	<u>Programmierlevel 3</u> Der Fahrroboter fährt geradeaus. Trifft er auf ein Hindernis, soll er diesem ausweichen.
Aufgabe 2 Advanced_2-wheeled_robot_2.rpp 	<u>Programmierlevel 3</u> Der Fahrroboter soll entlang einer schwarzen Linie, des vorgegeben Parcours fahren. Verlässt er die schwarze Linie, so bleibt dieser stehen.

Aufgabe 3 Advanced_2-wheeled_robot_3.rpp	<u>Programmierlevel 3</u> Das Programm soll die Fahrt auf der schwarzen Linie so regeln, dass das Modell die schwarze Linie wieder findet, falls der Roboter die schwarze Linie verlässt.
Aufgabe 4 Advanced_2-wheeled_robot_4.rpp	<u>Programmierlevel 5</u> Der Fahrroboter fährt eine Stecke entlang, erkennt er ein Hindernis reduziert er die Geschwindigkeit. Über einen Zufallsgenerator wird die Drehrichtung festgelegt, mit der dem Hindernis ausgewichen wird.

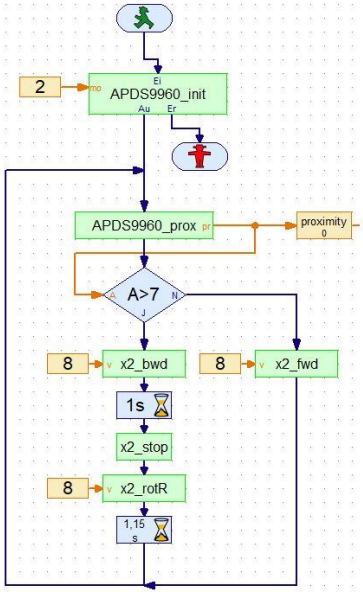
Aufgabe 1

Zur Lösung der ersten Aufgabe kannst du auf mehrere Unterprogramme aus vorherigen Programmieraufgaben zugreifen.



Im 2. Teil des Programms benötigst du die folgenden Befehle: **„x2_bwd“, „x2_stop“, „x2_rotR“, „x2_rotL“ und „x2_fwd“.**

Genau wie bei dem Modell „Dämmerungsschalter“, muss der Gestensensor zuerst initialisiert werden. Dazu fügst du den Block „APDS9960_init“ nach dem Programmstart ein. Der Wert des Eingangs „mo“ (mode) wird auf 2 festgelegt, denn du möchtest mit der Funktion „APDS9960_prox“ (Abstandmessung - proximity) arbeiten. Am Ausgang „Er“ wird der Befehl „Stopp“ angefügt.



Anschließend baust du den Block „APDS9960_prox“ zur Messung und Auswertung der Entfernung ein. Im folgenden Verzweigungsbefehl wird der ermittelte Wert ausgewertet. Ist er größer 7 - „J“ geht es im zweiten Teil des Programms weiter. Ist der Wert kleiner 7 „N“ fährt das Modell an.

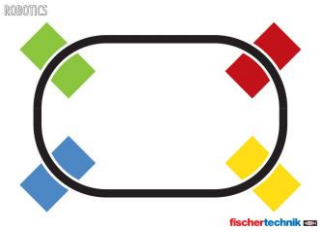
Anschließend folgt der Programmteil, in dem das Modell 1s rückwärtsfährt, dann stoppt und sich anschließend 1,15s dreht (Unterprogramm „x2_rotR“) bevor es wieder vor den Block „APDS9960_prox“ springt. Am „N-Zweig“ baust du das Unterprogramm zur Vorwärtsfahrt ein. Den Ausgang verbindest du als Schleife ebenfalls mit dem Eingang des Unterprogramms „APDS9960_prox“.

Somit ist das Programm fertig und du kannst es unter dem Namen

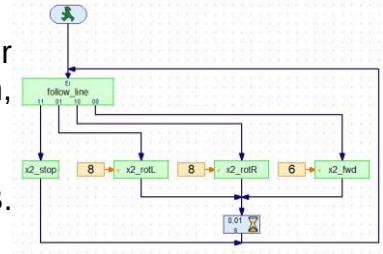
„Advanced_2-wheeled_robot_1.rpp“

auf deinem Rechner abspeichern. Teste anschließen das Programm zusammen mit dem Modell.

Aufgabe 2

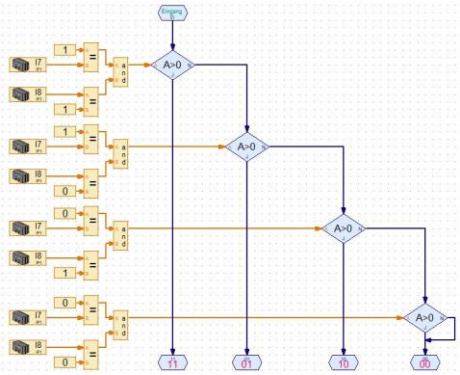


Für die zweite Aufgabe benötigst du, wie du aus der Aufgabenstellung entnehmen kannst, als Fahrstecke die schwarze Linie auf dem im Baukasten beigefügten Parcours.



Im rechten Bild siehst du schon einmal, wie das Hauptprogramm später aussehen soll. Sicher wirst du schon ein paar Unterprogramme erkennen, die bereits in vorherigen Aufgabenstellungen verwendet wurden.

Aber auch hier musst du ein paar neue Unterprogramme erstellen wie z.B. „follow_line“.



Dies ist ein Unterprogramm, welches du einmal programmierst und dann in allen Programmen verwenden kannst, bei denen eine Linienverfolgung benötigt wird.

Der IR-Sensor liefert vier verschiedene Eingangswerte. Je nach Wert z.B. IR7 und IR8 = 1 dann wird der Ausgang „11“ angefahren.

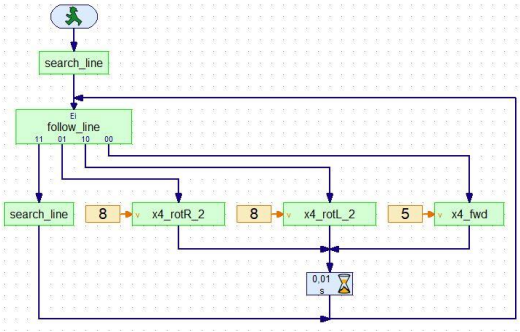
Im Hauptprogramm folgt das Unterprogramm „x2_stop“ - die Motoren stoppen. Über eine Endlosschleife wird somit die Fahrt entlang der Linie kontinuierlich überwacht und gegebenenfalls korrigiert.

Teste das Programm und speichere es unter dem Namen

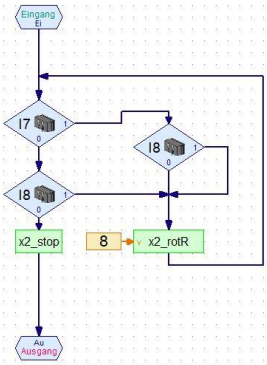
„Advanced_2-wheeled_robot_2.rpp“

auf deinem Rechner ab.

Aufgabe 3



Jetzt geht's an die 3. Aufgabe. Diese ähnelt der vorherigen Aufgabe – nur, dass zuerst eine schwarze Linie gesucht werden soll. Auch hier vorab das Hauptprogramm mit seinen Unterprogrammen. Neu hinzugekommen ist das Unterprogramm „search_line".



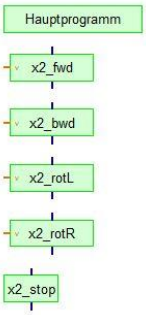
Mit diesem Unterprogramm wird die schwarze Linie gesucht. Haben beide IR-Sensoren den Wert „0", stoppen die beiden Motoren und das Unterprogramm wird verlassen. Ansonsten dreht das Modell über das Unterprogramm „x2_rotR". Um eine Drehbewegung zu erzeugen, laufen die Motoren gegenläufig.

Das Unterprogramm kennst du ja schon, so dass du es nur einfügen musst.

Das Gleiche gilt für die Unterprogramme im Hauptprogramm.

Wichtig: Solltest du nicht mehr wissen, wie man schon vorhandene Unterprogramme einfügt, ganz einfach – lade das Programm, in dem die Unterprogramme verwendet werden.

Wähle in der Elementgruppe „Geladene Programme".



Klicke nun auf den Ordner, in dem die Unterprogramme stehen. Es werden die Programmelemente angezeigt.

Das Unterprogramm, welches du benötigst, ziehst du einfach mit der Maus in den Programmierbildschirm und fügst es an ansprechender Stelle an.

Teste das Programm und speichere es unter dem Namen

„Advanced_2-wheeled-robot_3.rpp"

auf deinem Rechner ab.

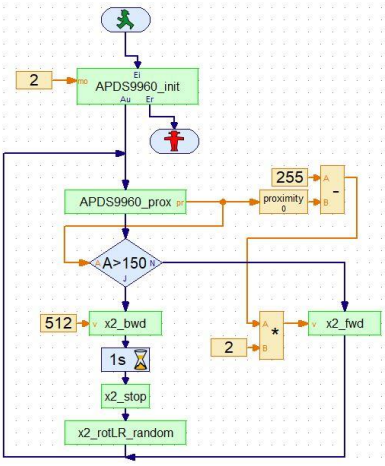
Das komplette Beispielprogramm findest du unter C:\Programm(x86)\ROBOPro\Beispielprogramme\ROBOTICS Smarttech\Advanced_2-wheeled_robot_3.rpp

Aufgabe 4

Jetzt aber noch zur 4. Aufgabe. Hier geht es nochmals um die Möglichkeit, einem Hindernis auszuweichen.

Lade zuerst das Programm „Advanced_2-wheeled-robot_4.rpp aus dem Ordner “C:\Programme(x86)\ROBOPro\Beispielprogamme\ROBOTICS Smarttech\

Auch hier kannst du dir schon einmal das Hauptprogramm anschauen. Im Vergleich zum Programm „Advanced_2-wheeled-robot_1.rpp “ findest du nur das Unterprogramm „x2_rotL_R_random“ sowie ein paar Parameterveränderungen und Erweiterungen. Möchtest du das Programm selbst erstellen, kannst du das rechts dargestellte Programm übernehmen.



Ansonsten empfiehlt es sich die Funktion des Programms anhand der Fahrwege des 2-Radroboters zu erkunden.

Stelle dazu einfach einen Parcours mit mehreren Büchern auf und starte das Programm. Wenn alles funktioniert, weicht der Roboter problemlos allen Büchern aus.

Speichere das Programm unter dem Namen

„Advanced_2-wheeled_robot_4.rpp“

auf deinem Rechner ab.

Demontiere anschließend das Modell und widme dich der nächsten Aufgabe.