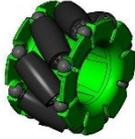


Gabelstapler



Mit diesem Modell erarbeitest du dir weitere Grundlagen der Programmierung. Die meisten der verwendeten fischertechnik-Bauteile kennst du ja schon von den vorherigen Aufgaben.

Im Modell verbaute Aktoren, Sensoren und technisches Zubehör.

Mini Motor	Getriebe	Taster	Omiwheelrad	IR-Spursensor
				

die Erklärung zu den Bauteilen findest du auf der Startseite.

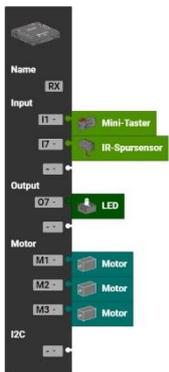
Das Modell „Gabelstapler“ gliedert sich in 2 Programmieraufgaben:

<p>Aufgabe 1</p> <p>Forklift_1.rpp</p>	<p><u>Programmierlevel 3</u></p> <p>Der Gabelstapler wird vor der Palette positioniert. Wird das Programm gestartet, nimmt der Stapler die Palette auf, dreht sich um 180° und setzt die Palette wieder ab.</p>
<p>Aufgabe 2</p> <p>Forklift_2.rpp</p> 	<p><u>Programmierlevel 3</u></p> <p>Palette wird auf „A“ positioniert. Gabelstapler wird auf der Linie in der Mitte, Richtung „A“ positioniert. Er fährt dann der Linie entlang, bis diese endet (Am Ende der Linie korrigiert er kurz) und nimmt dann eine Palette bei „A“ auf. Danach dreht er auf der Linie um, fährt der Linie wieder entlang, bis die Spur bei „B“ endet und setzt dann die Palette bei „B“ ab. Dann ist das Programm zu Ende.</p>

Zur Lösung der ersten Aufgabe kannst du auf mehrere Unterprogramme aus der Programmieraufgabe „Advanced_2_wheeled_robot_2.ft“ zugreifen.

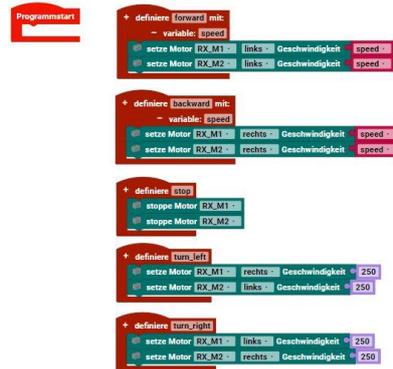
Tipp: Lade einfach das Programm und entferne alle Programmschritte, die nicht benötigt werden. So z.B. alle Befehle im Hauptprogramm, die Funktion „follow_line“ und die Funktion „find_line“.

Du benötigst die Unterprogramme „turn_left“, „turn_right“, „forward“, „backward“ und „stop“. Was du auch nicht neu definieren musst, ist die Variable „speed“.



Ändere die Controllerkonfiguration entsprechend der Bauanleitung ab.

Für beide Aufgaben benötigst du 3 Motoren (zwei für die Räder, einen für die Gabel), eine LED für die Blinkfunktion, einen Mini-Taster als Endtaster für die Gabel und den IR-Spursensor (Aufgabe 2).

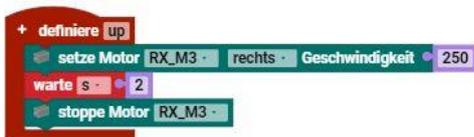


Aufgabe 1

Bevor du das Hauptprogramm definierst, musst du 3 neue Funktionen erzeugen.

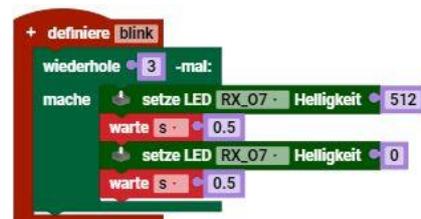


Beginne mit der Funktion „down“. Mit ihr wird die Gabel abgesenkt bis der Endtaster betätigt wird Motor RX_M3“. Da du schon Funktionen erstellt hast, findest du die Lösung auf der rechten Seite.



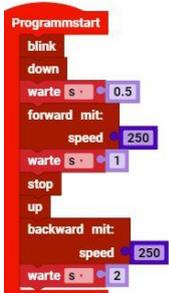
Im nächsten Schritt erzeugst du die Funktion „up“. Dazu duplizierst du einfach die Funktion „down“ und änderst den „warte bis ...“ Block auf „warte s ...“ und die Drehrichtung von „links“ auf „rechts“ ab.

Zum Schluss benötigst du eine Funktion, in der die LED blinken soll. Der Funktion gibst du den Namen „blink“. Die LED soll in einer „wiederhole“-Schleife „3“ mal aufleuchten. Hell- und Dunkelphase beträgt jeweils „0,5“ Sekunden. Erzeuge die nebenstehende Funktion.





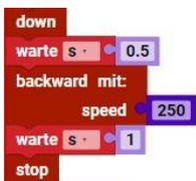
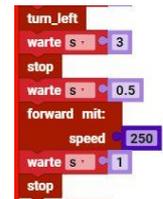
Nachdem alle Funktionen definiert sind, kannst du das Hauptprogramm beginnen. Beginne zuerst mit den Funktionen zur Aufnahme der Palette. Das Modell steht mit der Gabel vor der Palette. Wird das Programm gestartet, blinkt die LED. Anschließend fährt die Gabel in die Endposition und wartet dort „0,5“ Sekunden.



Das Modell fährt 1 Sekunde nach vorne, bleibt stehen und hebt die Palette an. Anschließend fährt das Modell 2 Sekunden rückwärts.

Wichtig: In den weiteren Darstellungen werden nur die neuen Befehle bzw. Blöcke im Dokument dargestellt.

Anschließend dreht das Modell „3“ Sekunden nach links „turn_left“ stoppt dann „0,5“ Sekunden und fährt dann „1“ Sekunde nach vorne um dann zu stoppen.



In den nächsten Schritten soll die Gabel gesenkt werden bis der Endtaster erreicht ist. Nach einer Wartezeit von „0,5“ Sekunden fährt das Modell „1“ Sekunde zurück, um dann zu stoppen.

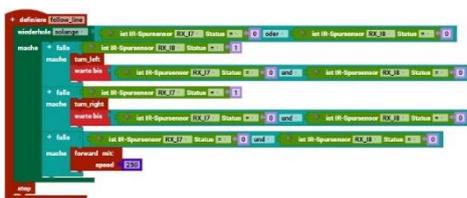
Nach einer weiteren Wartezeit von „0,5“ Sekunden fährt die Gabel nach oben und die Blinkfunktion der LED wird eingeschaltet.



Und schon ist das Programm fertig definiert. Teste das Programm und speichere es unter dem Namen **Forklift_1** auf deinem Rechner ab.

Aufgabe 2

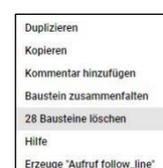
Als Basisprogramm verwendest du die Aufgabe 1. Erinnerst du dich noch an die Aufgabe „Advanced_2_wheeled_robot“. Dort sollte das Modell einer schwarzen Spur folgen. Diese Funktion benötigst du auch in der Aufgabe 2.



ROBOPRO Coding verfügt über eine Funktion, Programmteile zu kopieren, um sie in einem anderen Programm einfügen zu können.

Gehe wie folgt vor: Öffne die Datei „Advanced_2_wheeled_robot_3“. Dort findest du die Funktion „follow_line“. Diese wird in deiner jetzigen Aufgabe benötigt.

Klicke mit der „rechten Maustaste“ auf die Funktion. Es erscheint ein Kontextfenster, indem du „Kopieren“ auswählst. Die Funktion liegt in einem Speicherbereich deines Rechners (Zwischenablage).



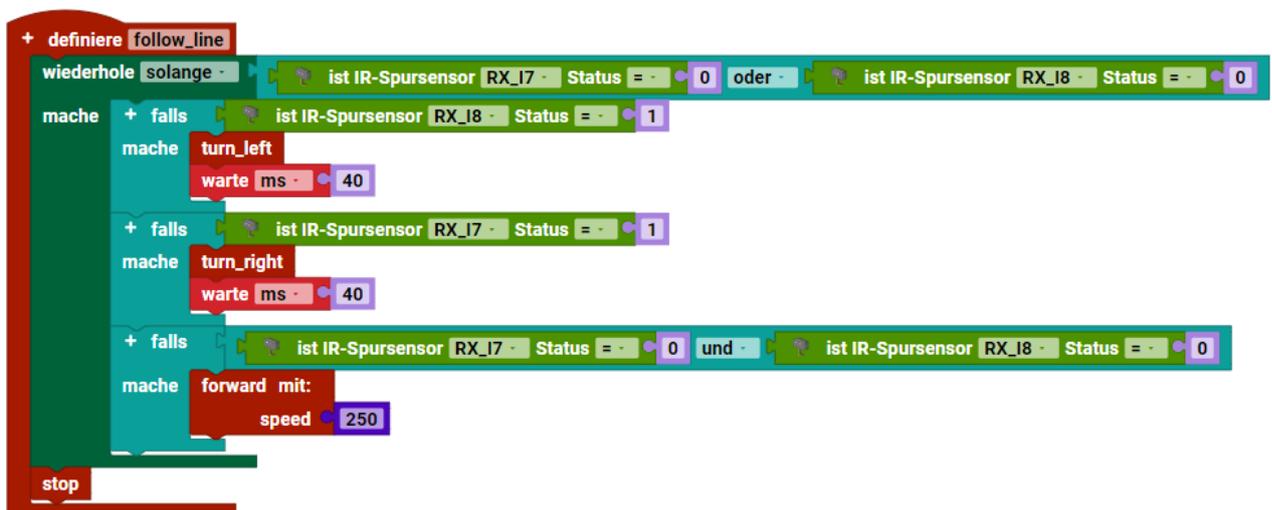


Öffne „Forklift_1“. Wenn du die „rechte Maustaste“ in dem leeren Programmierfensterbereich betätigst, erscheint ein Kontextfenster mit der Auswahl „Hier einfügen“. Klicke den Eintrag an und aus der Zwischenablage wird die Funktion in dein Programm „Forklift_1“ eingebunden.

Jetzt haben wir schon die Funktion "follow_line", aber im vorherigen Übungsbeispiel ging es darum, einer schwarzen Linie zu folgen, die jedoch in diesem Fall endet. Man könnte denken, die Funktion würde so, wie sie ist, funktionieren, und wenn der Roboter das Ende der Linie erreicht, würde er erkennen, dass er am Ende ist, aus der Schleife aussteigen und tun, was wir für ihn vorbereitet haben.

Aber so wird es nicht sein. Normalerweise fährt der Roboter nicht ganz gerade auf der Linie, das wird kaum auffallen, aber dieser kleine Unterschied führt dazu, dass einer der beiden Sensoren zuerst aktiv wird und zu korrigieren beginnt. Da es keine schwarze Linie mehr gibt, wird der Roboter weiterhin im Kreis fahren, bis er die Linie wiederfindet. Und wann wird das sein? Du hast es sicher schon erraten: wenn er eine komplette Runde gedreht hat und zurückkommt, wo er angefangen hat. Lustig, nicht wahr?

Lass uns das ändern, statt darauf zu warten, die Linie wiederzufinden, lassen wir ihn für 40ms oder 0,04 Sekunden korrigieren und dann erneut prüfen, welche Bedingungen vorliegen, das heißt, ob er komplett auf der schwarzen Linie ist, ob er in die andere Richtung korrigieren muss oder ob keine Linie da ist und er daher aufhören muss, der Linie zu folgen. Wir ersetzen die Blöcke des Wartens, bis beide Sensoren auf 0 sind, einfach durch 100 ms warten.



Warum haben wir das nicht von Anfang an so gemacht? Weil diese Lösung nicht so gut ist, um Kurven zu nehmen, da es eine feste Wartezeit gibt, könnte er zu viel korrigieren und müsste dann erneut korrigieren, aber jetzt ist das nicht so wichtig für uns, sondern das Wichtige ist zu erkennen, dass wir das Ende der schwarzen Linie erreicht haben.

Benenne jetzt schon einmal das Programm auf „Fortlift_2“ um und speichere das Teilprogramm auf deinem Rechner mit dem Namen **Forklift_2** ab.

Kopiere nochmals aus dem Programm „Advanced_2 ...robot_2“ die Funktion „find_line“ in „fork_lift_2“.



Definiere eine neue Funktion mit dem Namen „pick_up“. Wird diese im Hauptprogramm aufgerufen, soll die Palette am Punkt A aufgenommen werden. Nach einer Wartezeit von „0,5“ Sekunden fährt das Modell für „1“ Sekunde nach vorne, stoppt und hebt die Gabel über „up“ an.



Definiere eine weitere Funktion mit dem Namen „pick_down“. Wird diese im Hauptprogramm aufgerufen, soll die Palette am Punkt B abgelegt werden. Auch hier fährt die Gabel über „down“ nach unten. Das Modell fährt dann „1“ Sekunde mit „backward“ rückwärts und stoppt. Nach einer kurzen Wartezeit von „0,5“ Sekunden wird die Gabel mit „up“ angehoben. Anschließend folgt nochmals die kurze Wartezeit von „0,5“ Sekunden.



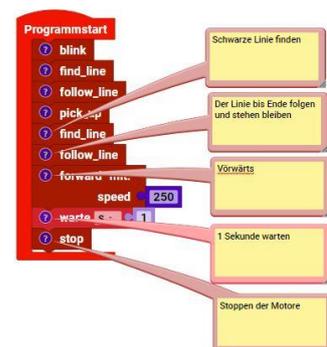
Jetzt sind alle für diese Aufgabe benötigten Funktionen erstellt und du kannst mit dem Hauptprogramm beginnen. Lösche zuerst alle Befehle im Bereich „Programmstart“.

Wichtig: Bei dieser Aufgabe wird das Programm nur einmal durchlaufen. Somit entfällt der schon bekannte Schleifenbefehl „dauerhaft wiederholen“.



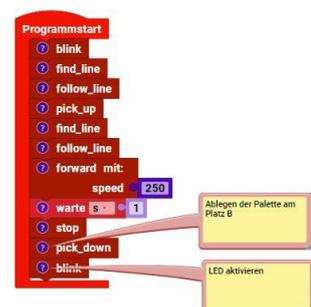
Nach dem Programmstart soll zuerst die LED blinken. Anschließend soll das Modell die schwarze Linie finden bzw. der Linie folgen. Am Endpunkt A der Linie bleibt das Modell über den Stop-Befehl in der Funktion „follow_line“ stehen. Mit der Funktion „pick_up“ wird die Palette aufgenommen.

Da sich die IR-Sensoren jetzt außerhalb der schwarzen Linie befinden, muss diese erneut gesucht werden. Danach fährt das Modell der schwarzen Linie entlang bis zum Endpunkt B. Dort bleibt es mit dem Stop-Befehl von „follow_line“ stehen. Die folgende Funktion „forward ...“ und „warte s ... 1“ sorgt dafür, dass die Palette über dem Ablageort „B“ steht. Mit der Funktion „stop“ bleibt das Modell stehen.



Die letzten beiden Funktionen, die du in dein Hauptprogramm einbindest, ist das Ablegen der Palette am Platz B. Anschließend wird nochmals die Blinkfunktion aufgerufen.

Teste jetzt das fertige Programm. Hat alles funktioniert, kannst du es nochmals auf deinem Rechner abspeichern.



Wenn du dir das letzte Programmbild anschaust, wirst du feststellen, dass die Funktionen und Befehle ein „blaues ?“ haben. Das bedeutet, dass ein „Kommentar“ hinterlegt wurde.

Probiere es einfach einmal aus. Klicke mit der „rechten Maustaste“ z.B. auf die Funktion „blink“. Es erscheint ein Kontextfenster mit der Auswahl „Kommentar hinzufügen“.



Aktiviere den Eintrag – das Kontextfenster wird ausgeblendet. Klicke mit der „linken Maustaste“ auf das „?“

Es erscheint ein „gelbes Eingabefenster“. Mit einem Klick in das Fenster erscheint der „Eingabecursor“. Jetzt kannst du über die Tastatur einen aussagekräftigen Text für diese Funktion eingeben.



Wenn du mit der Maus auf den Rahmen fährst, wird anstelle des Cursors eine Cursorhand dargestellt. Jetzt lässt sich der Rahmen bei „geklickter, linker Maustaste“ an eine beliebige Stelle auf dem Programmierbildschirm verschieben.

Möchtest du den Kommentar wieder ausblenden, klicke das „blaue ?“ mit der linken Maustaste an. Es wird anschließend nur das „blaue ?“ angezeigt.



Möchtest du den Kommentar komplett löschen, klickst du mit der „rechten Maustaste“ auf das „?“ . Es erscheint wieder ein Kontextfenster – diesmal mit der Auswahl „Kommentar entfernen“. Klicke die Auswahl an und das „blaue ?“ wird gelöscht.



~~Ok, beide Aufgaben sind gelöst. Bevor du das Modell demontierst, kannst du dir weitere Fahrmöglichkeiten überlegen und ausprobieren:~~

In dieser Übung haben wir etwas sehr Interessantes entdeckt, nämlich wie zwei scheinbar gleiche Aufgaben nicht auf die gleiche Weise gelöst werden konnten. Und jetzt, warum versuchst du nicht, die Wartezeiten zu ändern? Denn 40 ms sind für uns nichts, aber für einen Roboter ist das sehr viel. Was ist, wenn du 100 ms ausprobierst? Oder 10 ms? Und wenn du diese Änderung mit verschiedenen Geschwindigkeiten kombinierst? Du kannst sicherlich die Genauigkeit des Roboters verbessern, wenn er der Linie folgt, mit diesen Parametern.

Eine andere Lösung für dasselbe Problem wäre gewesen, nachdem wir eine Korrektur starten müssen, statt sofort den Roboter zu drehen, einige Millisekunden zu warten und dann erneut zu überprüfen, wie die beiden Sensoren stehen. Wenn beide weiß erkennen, ist die Linie wahrscheinlich zu Ende... oder vielleicht haben wir die Linie verloren und unser Roboter sucht jetzt dein Mathebuch anstatt des Paletts...

In der Programmierung und in der Robotik gibt es immer verschiedene Wege, um Lösungen für unsere Probleme zu finden... Und viele davon sind gut. Zögere nicht, zu experimentieren.

Tipp: Speichere weitere Programme zu diesem Modell immer mit einem neuen Namen ab.