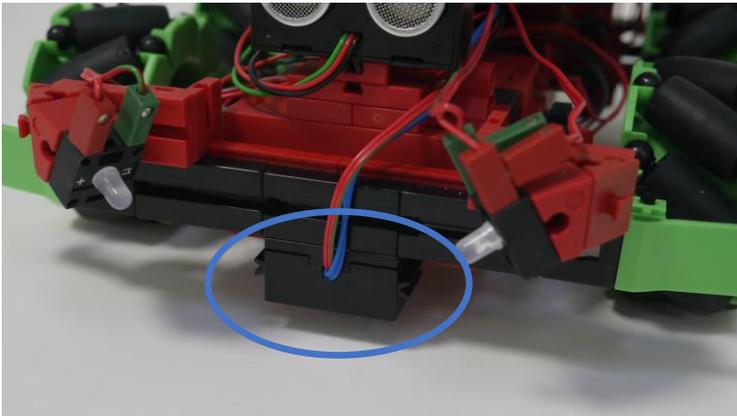


## Omniwheels X4 mit Sensoren – autonomes Fahren

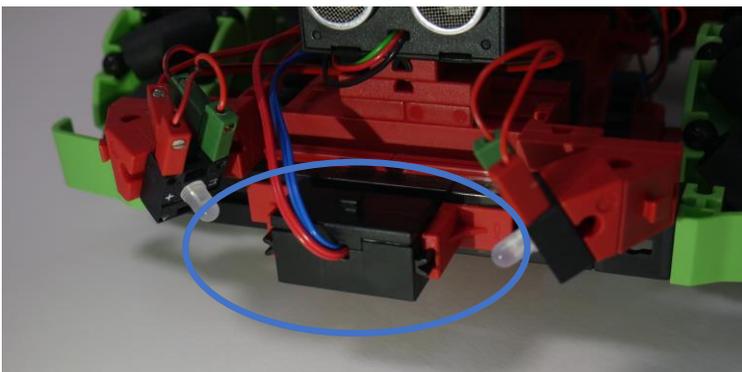
Mit diesem Modell erweiterst du das Modell Omniwheels X4 um die Sensoren IR-Spursensor, Ultraschall-Abstandssensor und Kamera. Dadurch ergeben sich zahlreiche neue Programmiermöglichkeiten rund um das Thema autonomes Fahren.

### **Achtung! Wichtiger Hinweis!**

Ist der IR Spursensor wie in der Bauanleitung abgebildet eingebaut, kann es vorkommen, dass durch zu starke Unebenheiten der Abstand zwischen Sensor und Parcours an manchen Stellen zu gering ist und die Spur nicht mehr richtig erkannt wird. Dieses Problem kann behoben werden durch einen einfachen Umbau: Befestige den Sensor wie in den folgenden Abbildungen gezeigt mit zwei Bausteinen 38423, die im Baukasten enthalten sind. Durch den größeren Abstand zum Parcours wird die Spur auch bei größeren Unebenheiten sicher erkannt.



Originaleinbau nach Bauanleitung



Umbau mit größerem Abstand

Das Modell „Omniwheels x4 mit Sensoren – Autonomes Fahren“ gliedert sich in 6 Programmieraufgaben:

omniwheels_x4_sensor_linefollower.ft	Einer Spur folgen mit Hilfe des IR-Spursensors.
omniwheels_x4_sensor_linefollower_ultrasonic.ft	Eine Spur suchen, der Spur folgen und Hindernisse mit dem Ultraschallsensor erkennen.
omniwheels_x4_sensor_linefollower_ultrasonic_acc.ft	Das Programm entspricht dem vorherigen mit dem Unterschied, dass die Entfernung zum Hindernis gemessen wird und die Geschwindigkeit entsprechend dem Abstand angepasst wird.
omniwheels_x4_sensor_servomove.ft	Testprogramm für die Kamera in Verbindung mit dem Servomotor.
omniwheels_x4_sensor_linefollower_camera_trafficsign.ft	Farberkennung mit der Kamera, Anzeige der Farbflächen auf dem Display, seitliches Einparken, warten am Zebrastreifen, beim Einschalten auf ON im Display nach Farbe Rot fahren
omniwheels_x4_sensor_linefollower_camera_pd.ft	Spurerkennung mit Kamera und PD-Regler. Modell soll an der schwarzen Linie des Parcours entlangfahren.

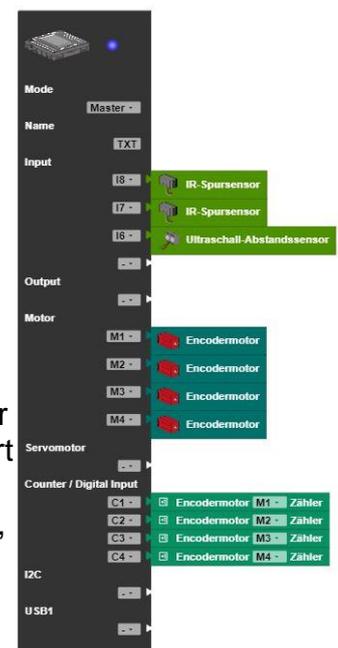
## Controllerkonfiguration

Starte zuerst das Programm „ROBO Pro Coding“ und führe die Controller-Konfiguration durch. Wo die Aktoren und Sensoren angeschlossen werden, entnimmst du wieder der Bauanleitung.

Für die Aufgaben benötigst du die 4 Encodermotoren und die dazugehörigen Zählgänge. Zusätzlich benötigst du zunächst den Ultraschall-Abstandssensor sowie den IR-Spursucher.

Diese Konfiguration wird für die ersten drei Aufgaben benötigt.

Ein ähnliches Programm hast du bei dem 2-rädrigen Roboter in der Aufgabe „omniwheels\_x2\_linefollower.ft“ schon kennengelernt. Dort werden 2 Motoren verwendet. Erstelle das Programm wie in diesem Beispiel schon durchgeführt und ändere das Programm ab, so dass statt 2 alle 4 Motoren entsprechend angesteuert werden.



```

Programmstart
setze speed_fast auf 512
setze speed_slow auf 350
setze state auf 0
dauerhaft wiederholen
mache
  setze state auf 1
  prüfe ist IR-Spursensor TXT_M_I7 Status = 0
  falls wahr 1
  falls falsch 0
  prüfe ist IR-Spursensor TXT_M_I8 Status = 0
  falls wahr 2
  falls falsch 0
+ falls
  state = 3
mache
  + setze Motor TXT_M_M1 links Geschwindigkeit speed_fast
  + setze Motor TXT_M_M2 rechts Geschwindigkeit speed_slow
sonst falls
  state = 0
mache
  + setze Motor TXT_M_M1 rechts Geschwindigkeit speed_slow
  + setze Motor TXT_M_M2 links Geschwindigkeit speed_fast
sonst
  + setze Motor TXT_M_M1 links Geschwindigkeit speed_slow
  sync mit Motor TXT_M_M2 Richtung links
  
```

Somit ist das Programm fertig und du kannst es testen. Speichere es noch auf deinem Rechner unter dem Namen

„omniwheels\_x4\_sensor\_linefollower“

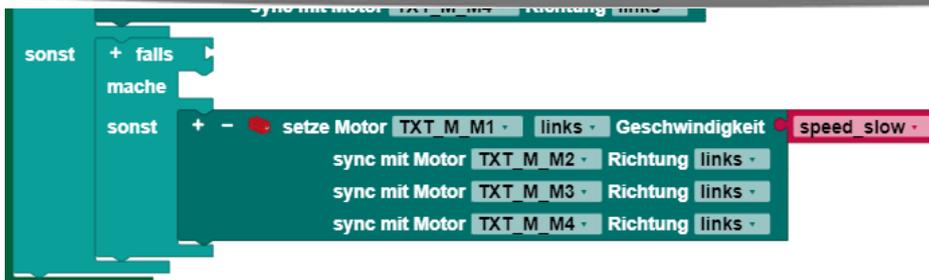
ab.

```

Programmstart
setze speed_fast auf 512
setze speed_slow auf 350
setze state auf 0
dauerhaft wiederholen
mache
  setze state auf 1
  prüfe ist IR-Spursensor TXT_M_I7 Status = 0
  falls wahr 1
  falls falsch 0
  prüfe ist IR-Spursensor TXT_M_I8 Status = 0
  falls wahr 2
  falls falsch 0
+ falls
  state = 3
mache
  + setze Motor TXT_M_M1 links Geschwindigkeit speed_fast
  sync mit Motor TXT_M_M3 Richtung links
sonst falls
  state = 0
mache
  + setze Motor TXT_M_M2 links Geschwindigkeit speed_fast
  sync mit Motor TXT_M_M4 Richtung links
sonst
  + setze Motor TXT_M_M1 links Geschwindigkeit speed_slow
  sync mit Motor TXT_M_M2 Richtung links
  sync mit Motor TXT_M_M3 Richtung links
  sync mit Motor TXT_M_M4 Richtung links
  
```

Stößt der Roboter auf ein Hindernis, soll dies vom Ultraschall-Abstandsensoren erkannt werden und der Roboter soll stehen bleiben.

Ersetze den Motorbefehl für Geradeausfahrt durch den Befehl „Führt eine Anweisung aus, falls eine Bedingung wahr ist“ aus der Gruppe „Logik“.



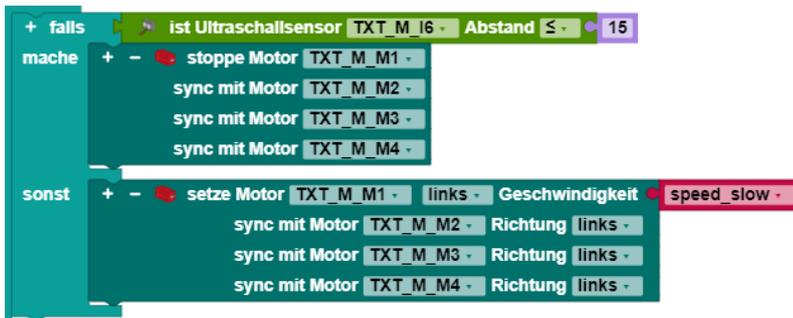
Eingang

Dann folgt  
Die Anweisung ist der Befehl „ist  
Ultraschall-sensor - TXT\_I6 – Abstand 15“ aus  
der Gruppe „Eingang“.

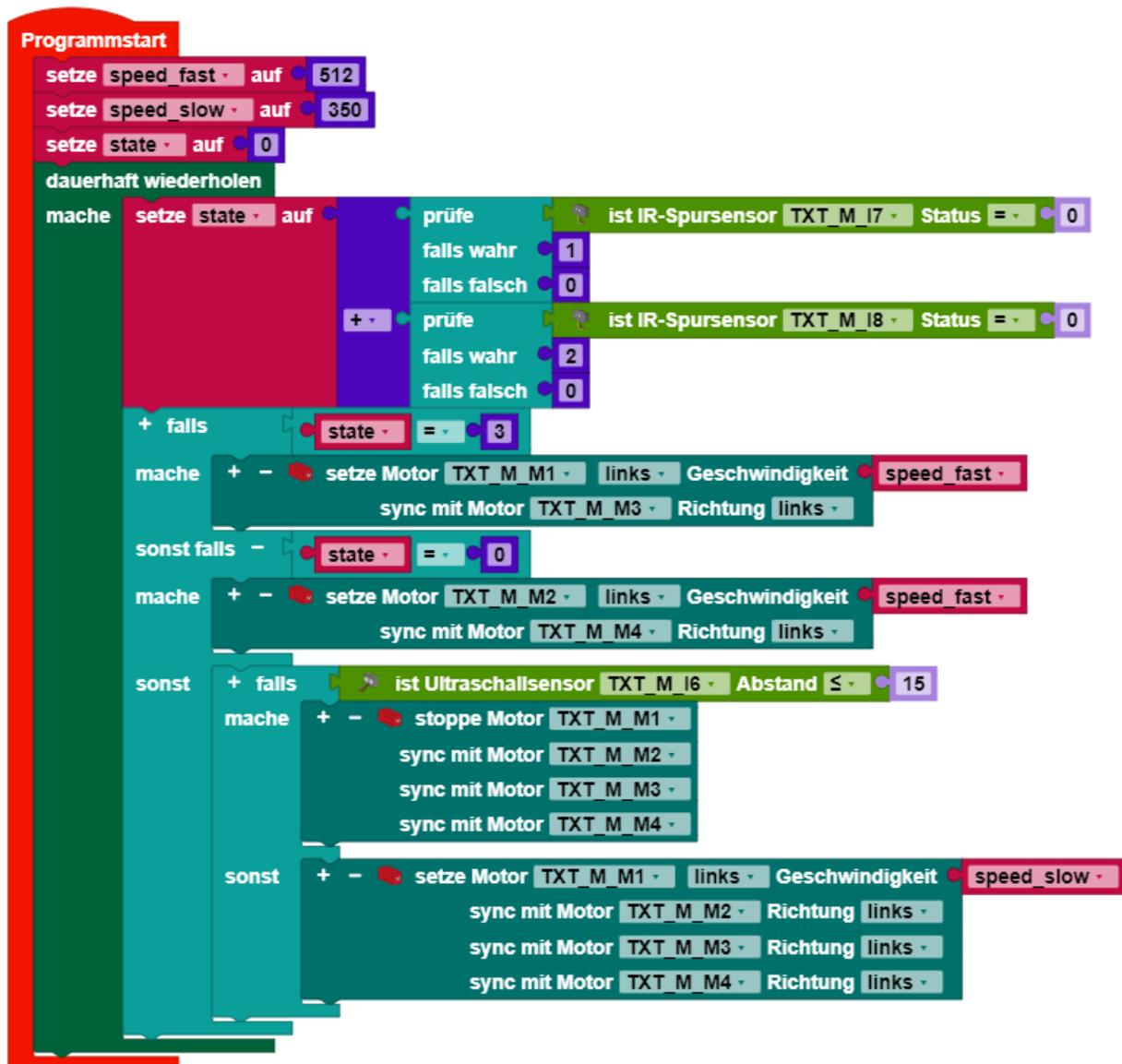


Ändere den Abstand auf „kleiner gleich“ um.

In den Abfragebereich „mache“ fügst du die Motorbefehle zum Stoppen der vier Motoren ein.



Das komplette Programm sieht dann folgendermaßen aus.



Teste nun dein Programm anhand der Aufgabenstellung. Funktioniert alles, kannst du es unter dem Namen

**„omniwheels\_x4\_sensor\_linefollower\_ultrasonic“**

auf deinem Rechner abspeichern und mit der nächsten Aufgabe beginnen.

Die nächste Aufgabe entspricht im Wesentlichen der letzten Aufgabe nur, dass die Geschwindigkeit in Abhängigkeit der Entfernung zu einem Hindernis verlangsamt wird. Wenn der Abstand zum Ultraschall-Abstandsensoren größer 10 cm ist, soll der Abstand gemessen werden und der Wert einer Variablen „speed\_slow“ übergeben werden. Ansonsten fährt der Roboter der Linie entlang und es wird ggf. die Fahrt nachgeregelt.

Ist die Entfernung kleiner/gleich 10 soll der Roboter stehen bleiben.

Baue den folgenden Befehl in dein Programm ein:

```

setze speed_slow - auf 512
begrenze zwischen 200 und 512
  * - 5
  begrenze zwischen 10 und 100
hole Ultraschallsensor TXT_M_I6 - Abstand

+ - setze Motor TXT_M_M1 - links - Geschwindigkeit speed_slow -
  sync mit Motor TXT_M_M2 - Richtung links -
  sync mit Motor TXT_M_M3 - Richtung links -
  sync mit Motor TXT_M_M4 - Richtung links -
  
```

Verwende den Befehl „setze speed\_slow auf“ gefolgt vom Befehl „Begrenze zwischen und“. Diesen findest du in der Gruppe „Mathe“, ebenso die Befehle zur Abstandsberechnung.

```

Programmstart
setze speed_fast - auf 512
setze speed_slow - auf 350
setze state - auf 0
dauerhaft wiederholen
mache
  setze state - auf 0
  prüfe ist IR-Spursensor TXT_M_I7 - Status == 0
  falls wahr 1
  falls falsch 0
  + - prüfe ist IR-Spursensor TXT_M_I8 - Status == 0
  falls wahr 2
  falls falsch 0
+ falls state - == 3
mache
  + - setze Motor TXT_M_M1 - links - Geschwindigkeit speed_fast -
    sync mit Motor TXT_M_M3 - Richtung links -
sonst falls state - == 0
mache
  + - setze Motor TXT_M_M2 - links - Geschwindigkeit speed_fast -
    sync mit Motor TXT_M_M4 - Richtung links -
sonst
  + falls ist Ultraschallsensor TXT_M_I6 - Abstand <= 10
  mache
    + - stoppe Motor TXT_M_M1 -
      sync mit Motor TXT_M_M2 -
      sync mit Motor TXT_M_M3 -
      sync mit Motor TXT_M_M4 -
  sonst
    setze speed_slow - auf 512
    begrenze zwischen 200 und 512
      * - 5
      begrenze zwischen 10 und 100
    hole Ultraschallsensor TXT_M_I6 - Abstand
    + - setze Motor TXT_M_M1 - links - Geschwindigkeit speed_slow -
      sync mit Motor TXT_M_M2 - Richtung links -
      sync mit Motor TXT_M_M3 - Richtung links -
      sync mit Motor TXT_M_M4 - Richtung links -
  
```

Teste anschließend dein Programm. Funktioniert alles, speichere das Programm unter dem Namen

**„omniwheels\_x4\_sensor\_linefoller\_ultrasonic\_acc“**

auf deinem Rechner ab.

Das folgende Programm kannst du einmal als Programmierübung eingeben. Es dient

dazu, dass du den Umgang mit dem Setzen einer Position des Servos kennenlernst.

```

Programmstart
setze posUp auf 250
setze posLine auf 160
setze posDown auf 120
setze Servomotor TXT_S1 Position posUp
warte s 1
setze Servomotor TXT_S1 Position posLine
warte s 1
setze Servomotor TXT_S1 Position posDown
warte s 1
setze Servomotor TXT_S1 Position posLine
warte s 1
setze Servomotor TXT_S1 Position posUp
    
```

Definiere zuerst die drei Variablen. Die Werte sind die Positionen des Servos von 0 bis 512 wobei der Wert 256 die Mittelstellung des Servos ist.



Füge aus der Gruppe „Motor“ den Befehl „setze Servomotor TXT\_S1 Position“ fünfmal unter die Variablendefinition ein, getrennt jeweils durch einen „warte“ Befehl mit 1s.

```

setze Servomotor - Position 256
    
```

Füge in die Positionsangabe der Servomotoren jeweils aus der Gruppe „Variablen“ die entsprechenden Variablen ein.

```

warte s 1
    
```

Hast du das Programm fertig geschrieben, teste es und speichere es wie gewohnt unter dem Namen

## „omniwheels\_x4\_sensor\_servomove“

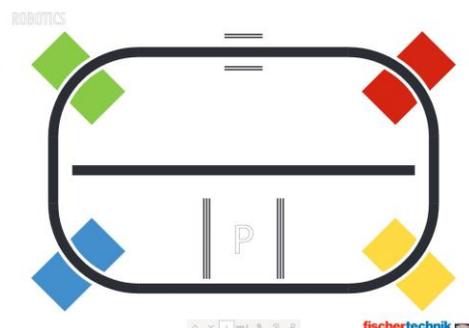
auf deinem Rechner ab.

Mit diesem Wissen kannst du dich an die nächste Aufgabe wagen.

Lade dazu das Programm

## „omniwheels\_x4\_sensor\_linefollower\_camera\_trafficsign“

Was soll mit dem Modell geschehen, wenn das Programm gestartet wird und das Modell auf dem Parcours steht? Die Abbildung zeigt dir den Parcours mit verschiedenen Farbfeldern und Strichen (Zebrastreifen, Einparkzone) sowie der Fahrstrecke. Das Fahrzeug soll die verschiedenen Farbflächen

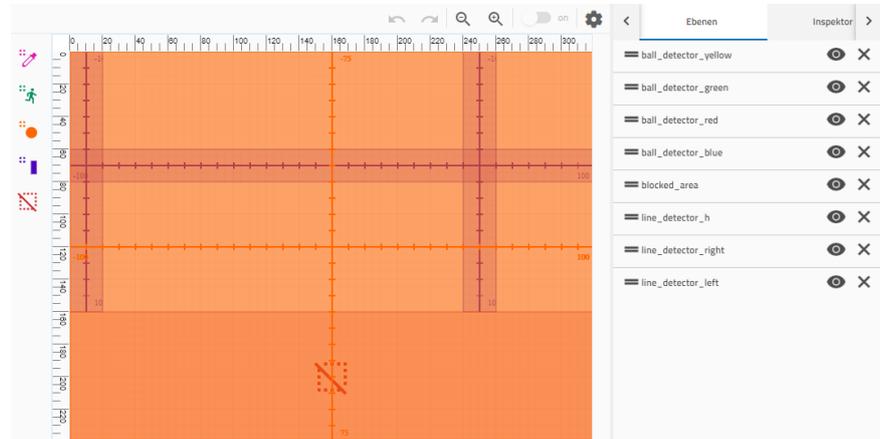


erkennen und jeweils eine Aktion ausführen.

Programmtechnisch wird die Kamera mit Servo, der IR-Farbsensor und das Display benötigt.

Schau dir bitte einmal die Konfiguration der Kamera an. Schalte dazu auf „Kamerakonfiguration“ um. Es erscheint folgendes Bild.

Die orangenen Rechtecke in der Mitte ermitteln über das Kamerafenster die einzelnen farbigen Flächen auf dem Parcours.

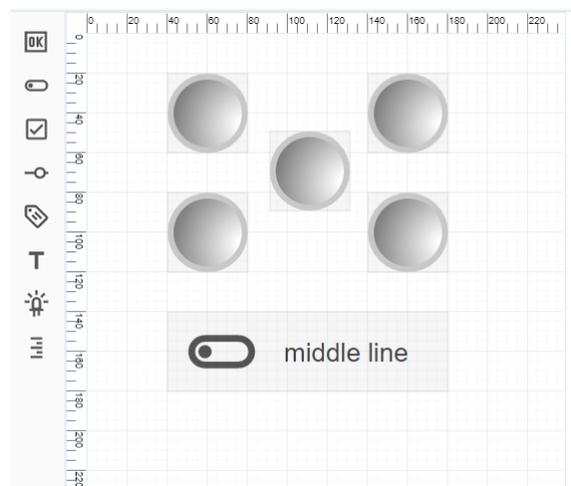


Die obere Linie ermittelt die Position der Einparkfläche. Die beiden senkrechten Linien ermitteln die Positionen des Zebrastrreifens.

Die einzelnen Sensorfelder und die Ebenen werden rechts neben der Kamerakonfiguration eingeblendet. Die einzelnen Ebenen können über das Symbol aus- und eingeblendet werden.

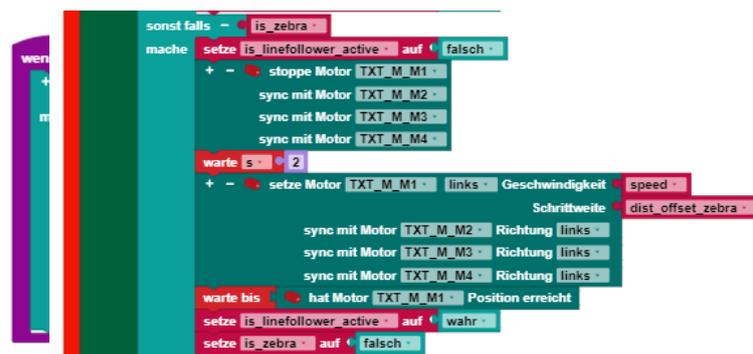


Öffne die „Anzeigekonfiguration“. Auf dem Display des TXT 4.0 Controllers wird angezeigt, wann eine der 4 Farbmarkierungen erkannt wurde. Wird der Schalter „middle line“ aktiviert, biegt der Roboter nach der roten Farbfläche ab und nimmt die „Abkürzung“ an der mittleren schwarzen Linie entlang.



Wirf noch einen Blick auf die Abfrage „wenn Linien line\_detector\_h erkannt: Ereignisliste“. Wenn der Roboter über den Parcours fährt, soll er am Beginn des Zebrastrreifens stehen bleiben, 2 Sekunden warten und dann weiterfahren.

Anschließend fährt der Roboter weiter bis die Position von „TXT\_m\_M1“ erreicht wurde



(Schrittweite „dist\_offset\_zebra“).

Noch ein Hinweis: Wird von der Robotersensorik die Parkposition erkannt, soll hier eingeparkt werden.

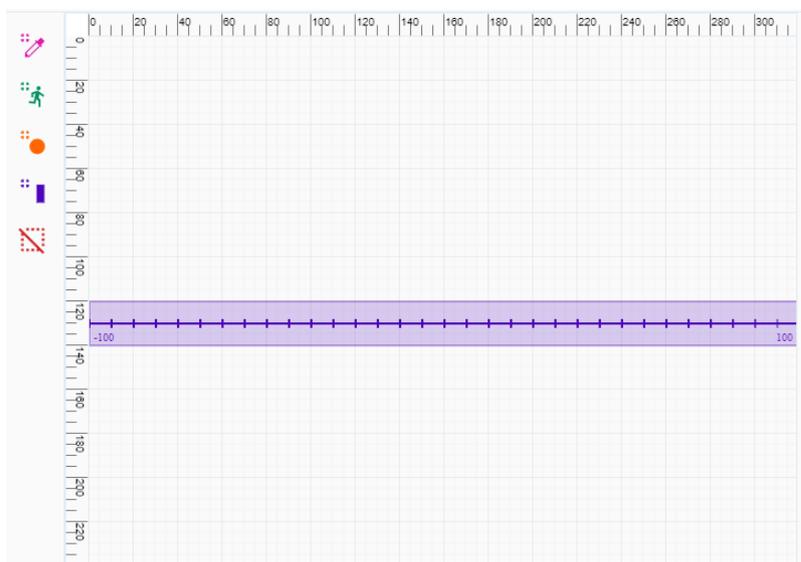
Das sollte für´s Erste reichen. Weitere Erklärungen folgen im dann im letzten Programm.

Speichere sicherheitshalber das Projekt auf deinem Rechner und teste es bevor du das nächste Projekt

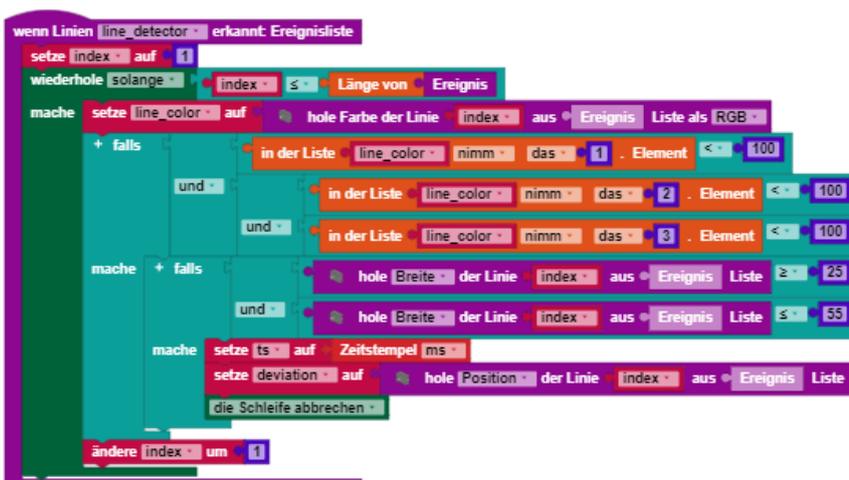
## „omniwheels\_x4\_sensor\_linefollower\_camera\_pd“

öffnest.

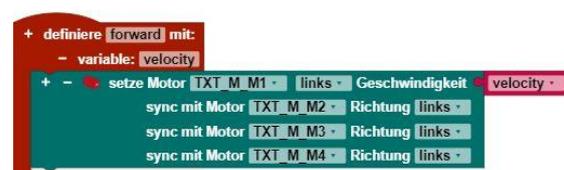
Das Robotermodell hat die Aufgabe mit Hilfe der Kamera an der schwarzen Linie entlang zu fahren. Schau dir zuerst die Kamerakonfiguration an. Das Bild kennst du schon aus der vorherigen Aufgabe. Die Kamera erkennt durch den „LineDetektor“ die schwarze Linie.



Im Hauptprogramm steht ein Befehlsblock, in dem abgefragt wird, ob im Kamerabild ein bestimmtes Ereignis erkannt wird: „wenn Linien ... erkannt: Ereignis“.



Liegt die Linie außerhalb des Abtastbereiches, müssen die Motoren nachgeregelt werden. Dies geschieht über den Befehlsblock „definiere turn mit.“



Mit dem Befehlsblock „definiere forward“ wird die



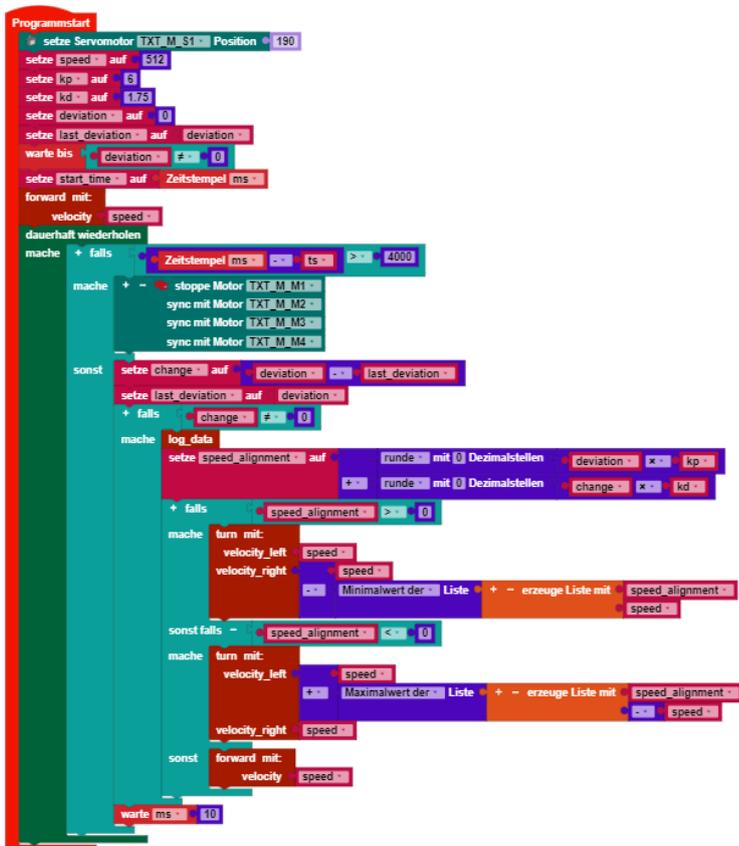
Vorwärtsbewegung festgelegt.

Mit Hilfe des Befehlsblocks „definiere log\_date“ werden in der Konsole von ROBO Pro Coding verschiedene Steuerdaten unten in der Konsole angezeigt.



Zum Schluss noch das Hauptprogramm. In ihm werden alle benötigten Variablen definiert und die Kameradaten weiterverarbeitet.

Teste auch dieses Modell und speichere das Projekt auf deinem Rechner ab.



Da mit der Kamera genau erkannt wird an welcher Stelle im Bild sich die Spur befindet, kann der PD-Regler mehr oder weniger die Fahrtrichtung korrigieren, abhängig davon, wie weit außerhalb der Mitte sich die Spur befindet. Dadurch ergibt eine sehr elegante Bewegung des Modells entlang der schwarzen Linie.

Somit hast du auch diese Aufgaben durchgeführt und kannst den Roboter für das nächste Modell umbauen.