

## Fussballroboter



Mit diesem Modell werden nochmals die Aktoren wie Encodermotor, Servo und als Sensoren die USB-Kamera und der Fototransistor verwendet

Das Modell Fussballroboter“ gliedert sich in 2 Programmieraufgaben:

omniwheels x4sensor_soccer_ballfollow ower.ft	Das Fußballfeld wird ohne Tor aufgebaut. Der Roboter soll sich im Kreis drehen und einen Ball suchen. Liegt der Ball im Schussbereich der Lichtschranke, wird er weggeschossen.
omniwheels x4sensor_soccer_ballfollow ower_goal.ft	Auf dem Fußballfeld wird ein Tor aufgebaut. Dieses und der Ball wird vom Roboter mit der Kamera erfasst und der Ball wird ins Tor geschossen.

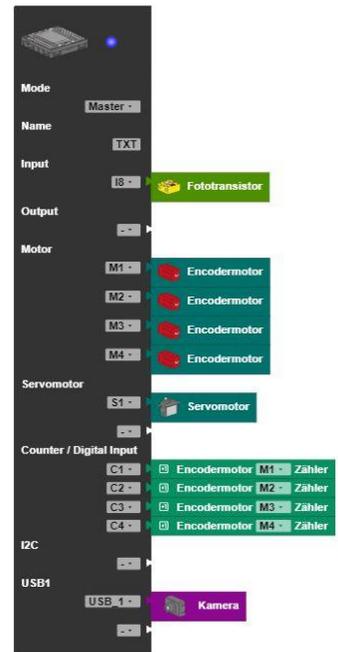
## Aufgabe 1

Starte zuerst das Programm ROBO Pro Coding lade aus den Beispielen für den Baukasten Hightech das Projekt

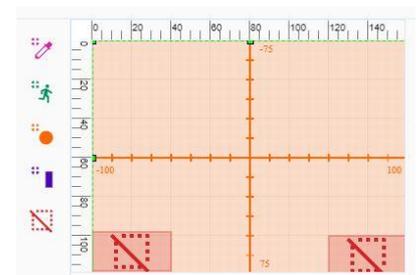
**„omniwheelsx4sensor\_soccer\_ballfollower.ft“**

Schau dir zunächst die Controller Konfiguration an. Sie entspricht dem Schaltplan in der Bauanleitung.

Für die Programme werden die 4 Encodermotoren und die dazugehörigen Zählgänge benötigt. Zusätzlich kommen der Fototransistor, der Servomotor sowie die Kamera zum Einsatz. Die LED für die Lichtschranke kommt in der Konfiguration nicht vor, da sie direkt an die 9V Out Buchse des TXT 4.0 Controllers angeschlossen ist.



Schau dir auch einmal die Kamerakonfiguration an.



Hier sind drei Bereiche definiert:



- Die gesamte Fläche mit dem Befehl „ball\_detector“, der über das gesamte Kamerabild gezogen wird.



- Zwei „block area“, geblockte Bereiche, die nicht ausgewertet werden, da in diesen Bereichen Teile des Modells im Bild zu sehen sind.

Die Controllerkonfiguration ist bei beiden Robotern gleich.

Wenn du dir das Programm anschaust, wirst du wieder bekannte Befehlsblöcke finden. Das sind z.B. alle Fahrbefehle, die der Roboter ausführen kann.

Wird das Programm gestartet, dreht sich der Roboter zuerst im Kreis, bis die Kamera den Ball erkennt. Dies geschieht über den Programmteil „wenn Ball ....“. Wird ein Ball erkannt, werden verschiedene Parameter eingestellt.



Die Werte werden ins Hauptprogramm übergeben und der Roboter fährt auf den Ball zu, bis die Lichtschranke, die am vorderen Teil des Roboters befestigt ist, unterbrochen wird.

```

Starte jedes mal
  ist Fototransistor TXT_M_I8 - dunkel -
    setze b_shot - auf wahr -
    warte s - 1
  
```

Ist die Lichtschranke unterbrochen wird der Variablenwert „b\_shot“ auf wahr gesetzt, ins Hauptprogramm übergeben und ausgewertet.

Im Logikblock „falls – mache – sonst“ wird die Variable „b\_shot“ abgefragt. Ins Lichtschrankenprogramm wurde diese mit „wahr“ übergeben.

```

dauerhaft wiederholen
  mache + falls b_shot -
    mache shotBall
    setze b_shot - auf falsch -
  
```

Anschließend wird die Funktion „shotBall“ aufgerufen. Ist diese abgearbeitet, wird die Variable „b\_shot“ wieder auf „falsch“ gesetzt.

In dieser Funktion findest du auch die Steuerung des Servomotores, der den Ballabschuss realisiert. Gleich zu Beginn der Funktion wird der Befehl „setze Servomotor ... Position 400“ aufgerufen. Der Servo fährt auf die Position 400 und schießt somit den Ball weg. Mit dem Befehl „setze Servomotor ... Position 250“ wird der Servo wieder auf die Startposition bewegt. Gleichzeitig werden noch zwei Fahrbefehle für den Roboter aufgerufen.

```

+ definiere shotBall
  setze Servomotor TXT_M_S1 - Position 400
  + - setze Motor TXT_M_M1 - links - Geschwindigkeit 512
    Schrittweite 50
    sync mit Motor TXT_M_M2 - Richtung links -
    sync mit Motor TXT_M_M3 - Richtung links -
    sync mit Motor TXT_M_M4 - Richtung links -
  warte ms - 50
  setze Servomotor TXT_M_S1 - Position 250
  warte bis hat Motor TXT_M_M1 - Position erreicht
  + - setze Motor TXT_M_M1 - rechts - Geschwindigkeit 512
    Schrittweite 20
    sync mit Motor TXT_M_M2 - Richtung rechts -
    sync mit Motor TXT_M_M3 - Richtung rechts -
    sync mit Motor TXT_M_M4 - Richtung rechts -
  warte bis hat Motor TXT_M_M1 - Position erreicht
  
```

Wenn du das Programm testest, werden auf deinem Programmbildschirm im Bereich „Konsole“ die im Block „wenn Ball ...“ ermittelten Parameter angezeigt.

```

gib aus formatiere text " ts:{.1f} ball:{} {} "
+ - mit ts_delta -
ball_pos_x -
ball_d -
  
```

Bevor du das zweite Programm öffnest, speichere das Programm unter dem Namen

**„omniwheels\_x4\_soccer\_ballfollower.ft**

auf deinem Rechner ab.

## Aufgabe 2

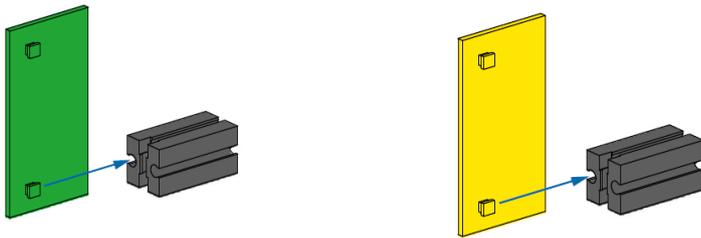
Lade für die neue Aufgabe das Programm

**„omniwheels\_x4\_soccer\_goal.ft“**

Was soll geschehen, wenn das Programm gestartet wird?

Der Roboter erfasst das Tor (beide Pfosten) und den Ball auf dem Fußballfeld und positioniert sich in Richtung Tor, so dass der Ball schließlich ins Tor geschossen werden kann. T

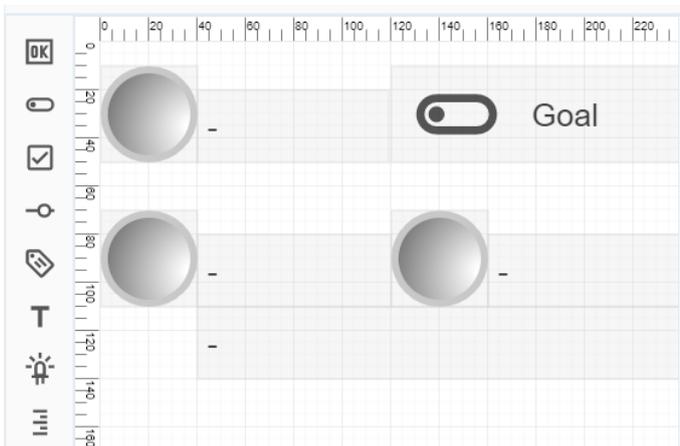
Das Tor besteht aus 2 Farbtafeln, die auf der Torlinie aufgestellt werden.



Tormarkierungen: **links: Grün, rechts: Gelb**

Fußballroboter und Ball werden im Parcours zufällig positioniert, der Fußballroboter steht innerhalb der schwarzen Markierung.

Der Roboter sucht nach dem Ball, fährt in Position und schießt ins Tor.

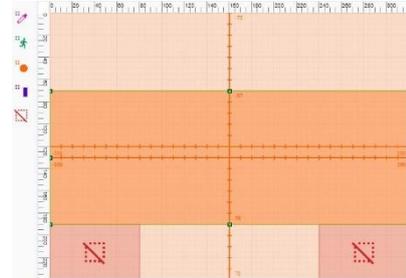


Die detektierten Objekte werden am Display angezeigt:

- Position des Balls,
- linker und rechter Torpfosten

Sie werden mit 3 Anzeigeelementen in unterschiedlichen Farben dargestellt.

Schau dir auch einmal die Kamerakonfiguration an. Hier findest du geblockte Bereiche, Bereiche zur Ballerkennung, und übereinander gelagert die Erkennung der Torpfosten links und rechts



Wie ist nun der Ablauf, wenn du den Roboter in Betrieb nimmst?

1. START Knopf am Display drücken
2. Falls kein Ball erkannt wird sucht der Roboter den Ball und fährt vor ihn
3. Falls der Roboter in der Nähe des Balls ankommt, dreht sich der Roboter mit einer seitlichen Bewegung um den Ball („Roboter tanzt um den Ball herum“) bis sich der Ball im Kamerabild genau in der Mitte der beiden Torpfosten befindet
4. Anschließend Torschuss mit Soundausgabe
5. Der Roboter bleibt stehen, bis wieder START gedrückt wird
6. Der Ball darf sich nicht im Tor befinden, sonst wird das Programm sofort abgebrochen.

Findet die Kamera die Torpfosten so wird dies im Display farblich und mit den x/y-Koordinaten angezeigt. Dies gilt ebenfalls für den Ball.

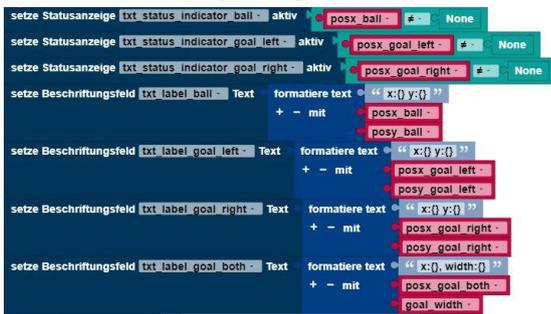


Wie schon erwähnt, wird das Programm auf dem Display mit der folgenden Befehlsabfrage gestartet.

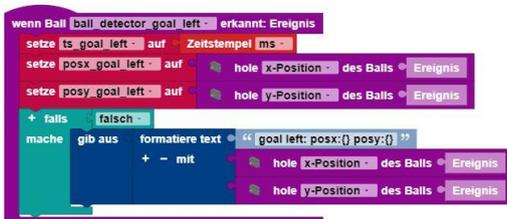
```

dauerhaft wiederholen
mache + falls
  ist Schalter txt_switch_ball - checked -
mache
  setze Schalter txt_switch_goal - checked - falsch -
  setze state - auf - " NONE "
  
```

Die ermittelten Werte der Kamera-Bilddaten werden in einer Befehlsfolge verarbeitet und auf dem Display dargestellt.



Im Unterprogramm „wenn Ball balldedector\_goal\_left“ wird das linke Tor erkannt und die ermittelten Daten weiterverarbeitet. Dies gilt auch für das Unterprogramm „wenn Ball balldedector\_goal\_right“.



Schaue dir einfach auch die anderen Unterprogramme an. Du wirst erkennen, dass einige dir aus dem ersten Programm bekannt sind.

Probiere einfach das fertige Programm aus und teste damit deinen Fußballroboter.

Speichere auch dieses Programm auf deinem Rechner ab.

**Wichtiger Hinweis:** Falls der Roboter die Torpfosten nicht zuverlässig erkennt, kann das an den Lichtverhältnissen im Raum und den in der Kamerakonfiguration eingestellten Farbwerten für den gelben und grünen Torpfosten liegen. In diesem Fall rufst du in dem ROBO Pro Coding Programm die Kamerakonfiguration auf und änderst in den Feldern „ball\_detector\_goal\_left“ und right die Farbwerte so, dass sie korrekt erkannt werden.

Viel Erfolg beim Fußball spielen!