Name: \_\_\_\_\_\_\_\_\_\_ Klasse: \_\_\_ Datum: \_\_\_\_\_\_\_\_

# Lösungsblatt Aufgabe 2

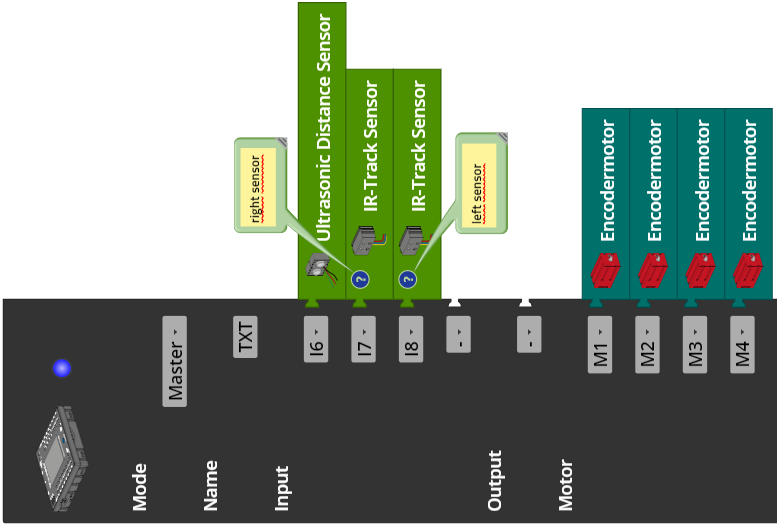
# Spurfolger

*Die Lösungen der Schülerinnen und Schüler sollen für die Steuerung des Fahrzeugs die Unterfunktionen aus der Aufgabe 1 verwenden. Bei der Programmierung des Spurfolgers soll – wie in der Lösung von Aufgabe 8 des Robotics TXT 4.0 Base Set – eine Zustands­variable („state“) zur Unterscheidung der Zustände verwendet werden. Damit werden die Programme sehr klar und übersichtlich.*

*Die beiden Experimentieraufgaben vermitteln das Verständnis für die Entwicklung und Konfiguration eines P- und eines PD-Reglers. Dabei ist vor allem die Ausgabe und grafische Veranschaulichung der Messwerte von besonderer Wichtigkeit.*

## Konstruktionsaufgabe

Anschluss der Sensoren:

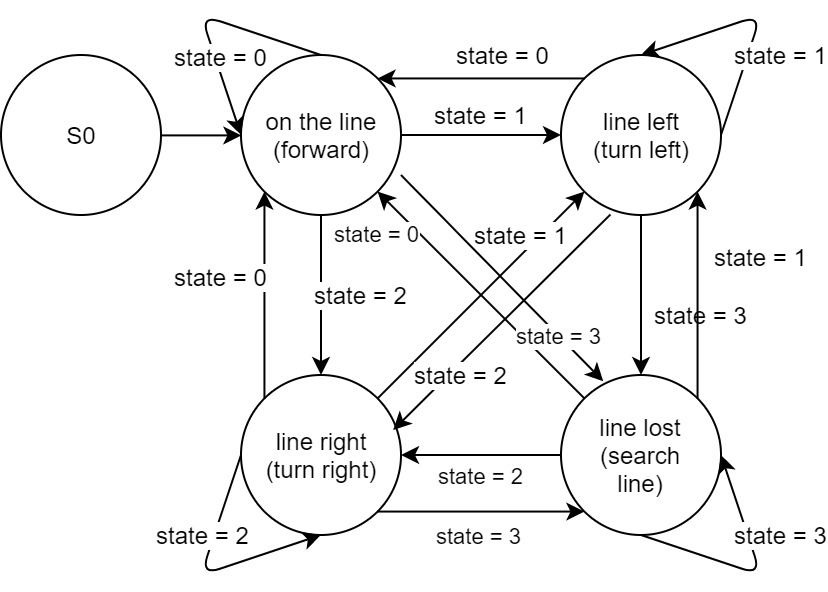


## Programmieraufgaben

**1. Spurfolger mit Spursensor**

Wie der Buggy in Aufgabe 8 des Robotics TXT 4.0 Base Set soll das Mecanum-Omniwheels-Fahrzeug so gesteuert werden, dass die beiden IR-Sensoren des Spursensors mittig über der schwarzen Linie liegen, also beide den Wert 0 liefern.

1a. Zustandsübergangsdiagramm des (digitalen) Spurfolgers:



*State-Transition\_Diagram\_Line\_Follower.drawio*

1b. Abhängig von den Werten des linken und des rechten IR-Sensors wird die Zustandsvariable *state* mit den folgenden Werten belegt:

* *state* = 0 → auf der Spur (beide Sensoren liefern den Wert 0)
* *state* = 1 → Spur liegt links (nur der linke Sensor hat den Wert 0)
* *state* = 2 → Spur liegt rechts (nur der rechte Sensor hat den Wert 0)
* *state* = 3 → Spur verloren (beide Sensoren liefern den Wert 1)

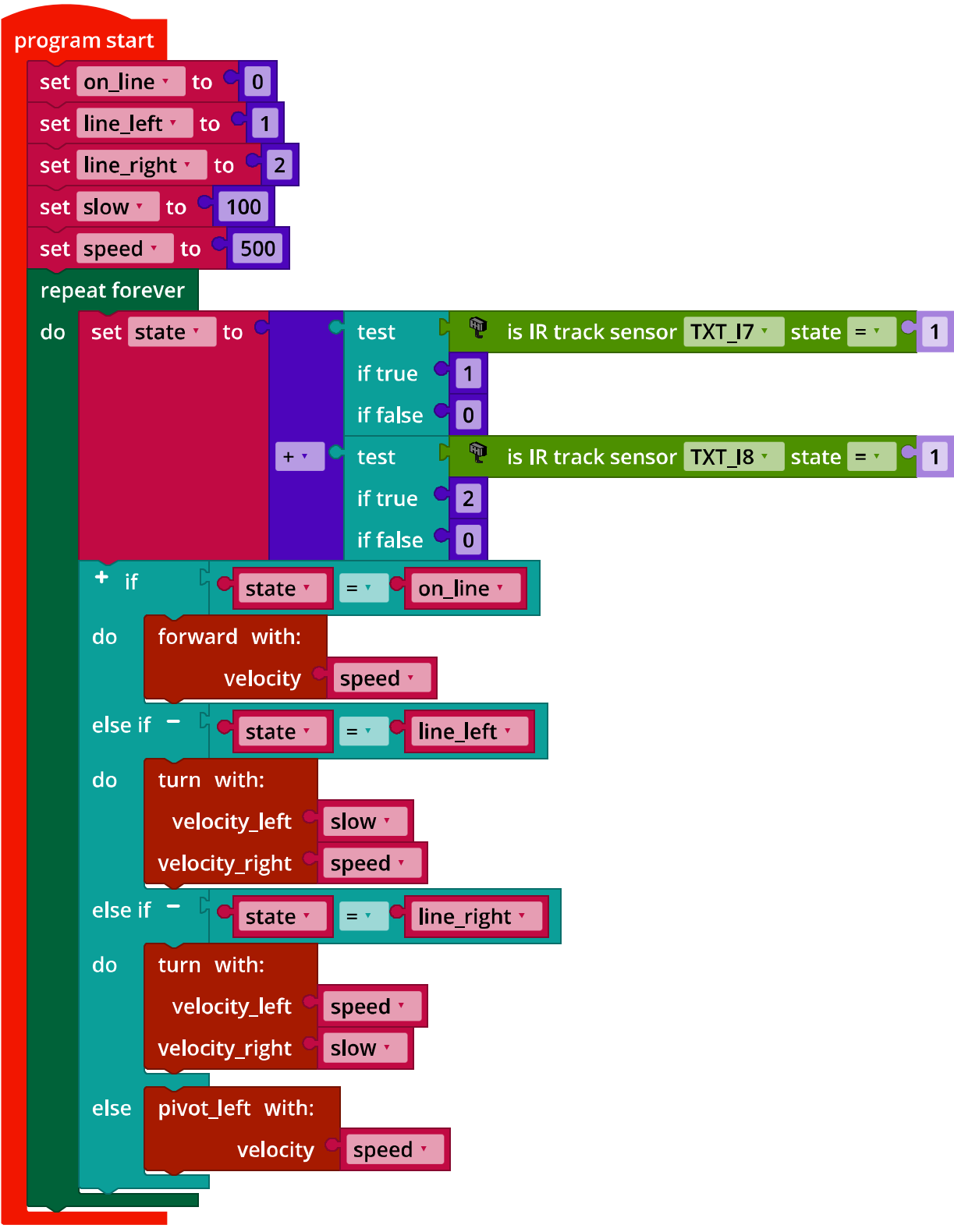
|  |  |  |
| --- | --- | --- |
|  | I8 (linker Sensor ) | I7 (rechter Sensor) |
| Schwarze Linie (Spur) | 0 → *state* += 0 | 0 → *state* += 0 |
| Helle Fläche | 1 → *state* += 2 | 1 → *state* += 1 |

Lesebeispiel: Liefert der linke Sensor (I8) den Wert 0 und der rechte (I7) den Wert 1, dann liegt die Spur links von der Mitte des Fahrzeugs. Die Zustandsvariable *state* erhält den Wert 1, und in diesem Zustand muss das Fahrzeug nach links gesteuert werden, damit der Sensor wieder mittig über der Spur liegt.

Die folgende Beispiellösung verwendet die in Aufgabe 1 programmierten Naviga­tions-Funktionen des Mecanum-Omniwheels-Fahrzeugs. Dadurch wird das Programm sehr über­sichtlich.

Zu Beginn der Schleife wird die Zustandsvariable „state“ abhängig von den Werten der beiden IR-Sensoren auf einen Zustandswert aus {0, 1, 2, 3} gesetzt.

Programmauszug (Beispiel):



*Mecanum\_Line\_Follower\_digital.ft*

1c. Die Geschwindigkeit des Spurfolgers lässt sich erhöhen, indem die Hauptge­schwindigkeit („speed“) möglichst groß gewählt und der Geschwindigkeits­unterschied der Motoren beim Lenken verringert, also die Geschwindigkeit „slow“ so angepasst wird, dass der Spurfolger in Kurven gerade nicht die Spur verliert.

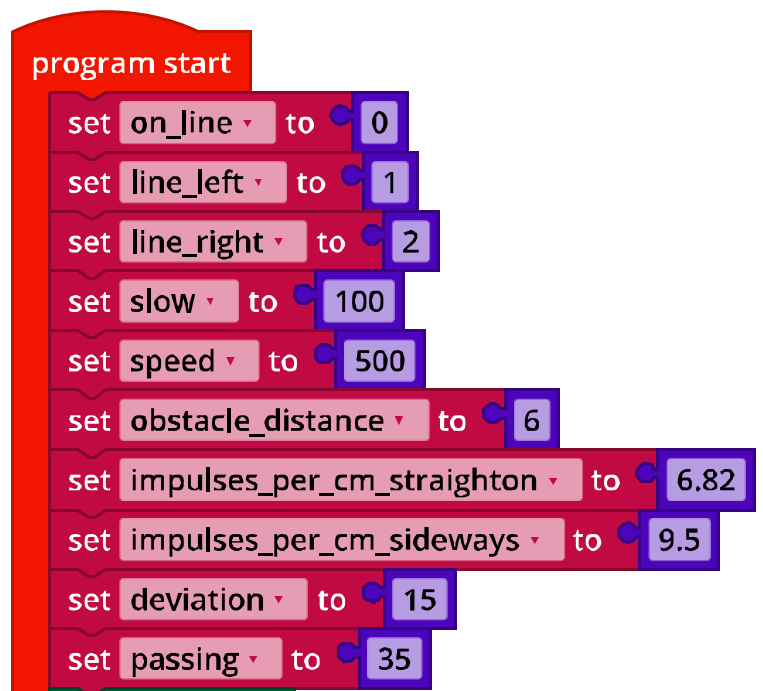
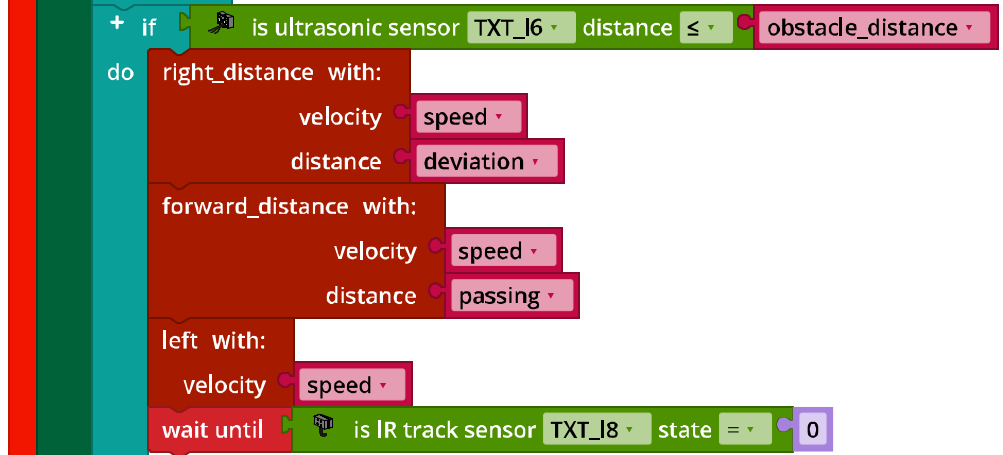
**Hinweis**: Den digitalen Spurfolger kann man auch als Dreipunktregler verstehen, der abhängig von den drei Zuständen „links von der Spur“, „auf der Spur“ und „rechts von der Spur“ die Geschwindigkeit der Motoren regelt. Der Bereich, in dem beide IR-Sensoren den Sollwert „0“ liefern (beide also direkt über der Spur liegen), wird auch als *Hysterese* bezeichnet.

**2. Spurfolger mit Hinderniserkennung**

Nach dem Ausweichen kann die Linie hinter dem Hindernis wiedergefunden werden durch eine

* erneute seitliche Fahrt (Nachteil: die Tiefe des Hindernisses ist nicht bekannt, daher kann das Hindernis gestreift oder aber die Linie verfehlt werden, wenn direkt dahinter eine enge Kurve folgt; Vorteil: gerade Linie wird in richtiger Position gefunden)
* Diagonalfahrt (Nachteil: das Hindernis kann gestreift werden; die Linie wird mit höherer Wahrscheinlichkeit verfehlt; Vorteil: eine gerade Linie wird in richtiger Ausrichtung für die Weiterfahrt gefunden)
* Drehung um 45°-90° mit anschließender Geradeausfahrt (Nachteil: das Hindernis kann gestreift, die Linie verfehlt oder auch eine Fortsetzung der Fahrt in der falschen Richtung erfolgen, wenn die Linie gefunden ist)
* Kurvenfahrt um das Hindernis (Nachteil: es kann eine Fortsetzung der Fahrt in der falschen Richtung erfolgen, wenn die Linie gefunden ist)

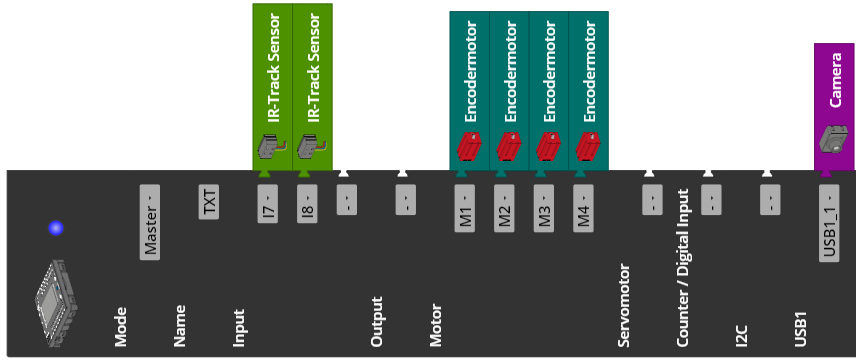
Programmauszug (Beispiel):

 … 

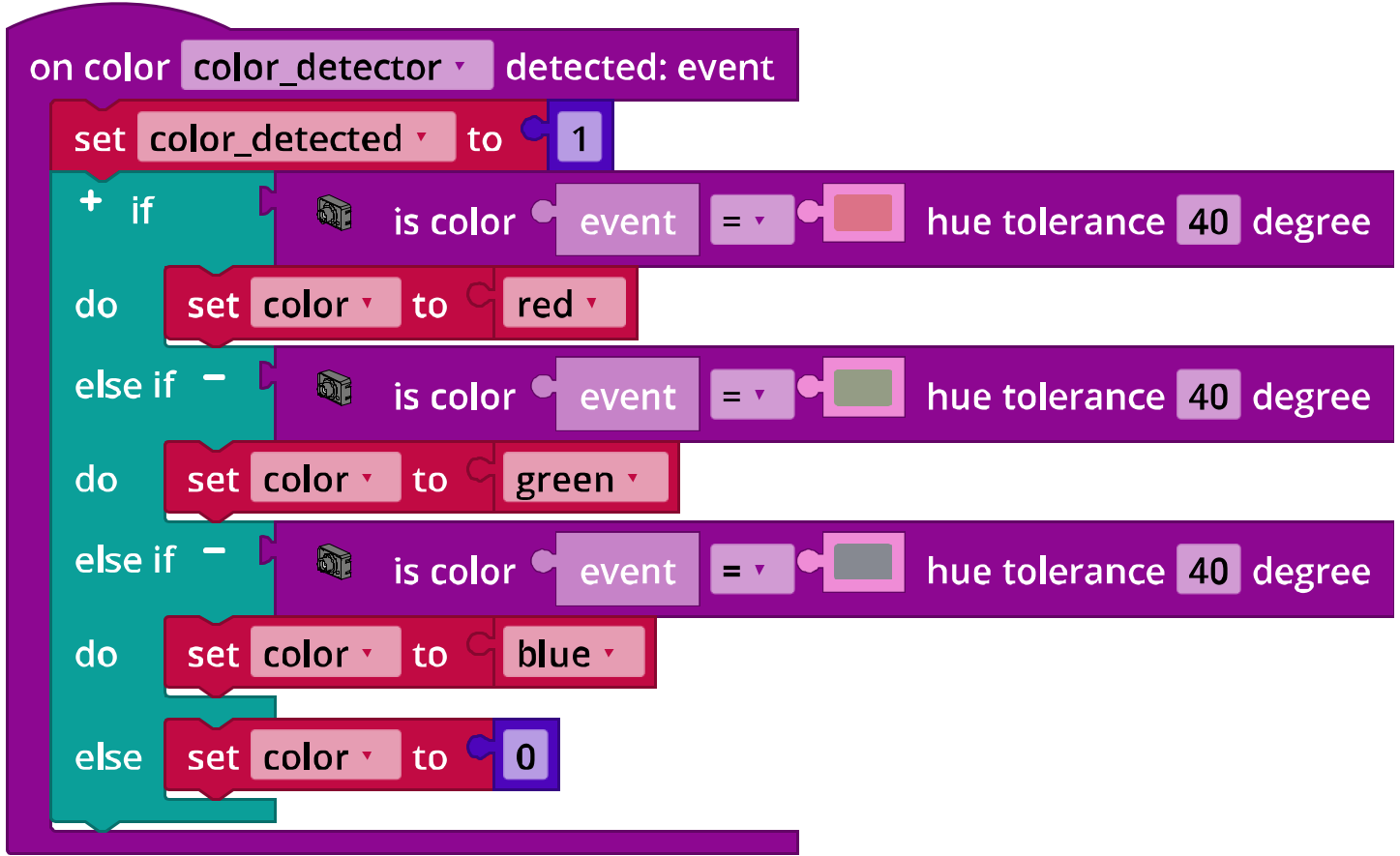
*Mecanum\_Line\_Follower\_Obstacle\_digital.ft*

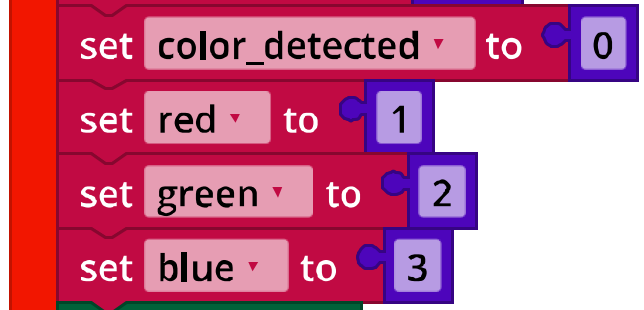
**3. Spurfolger mit Farbsteuerung**

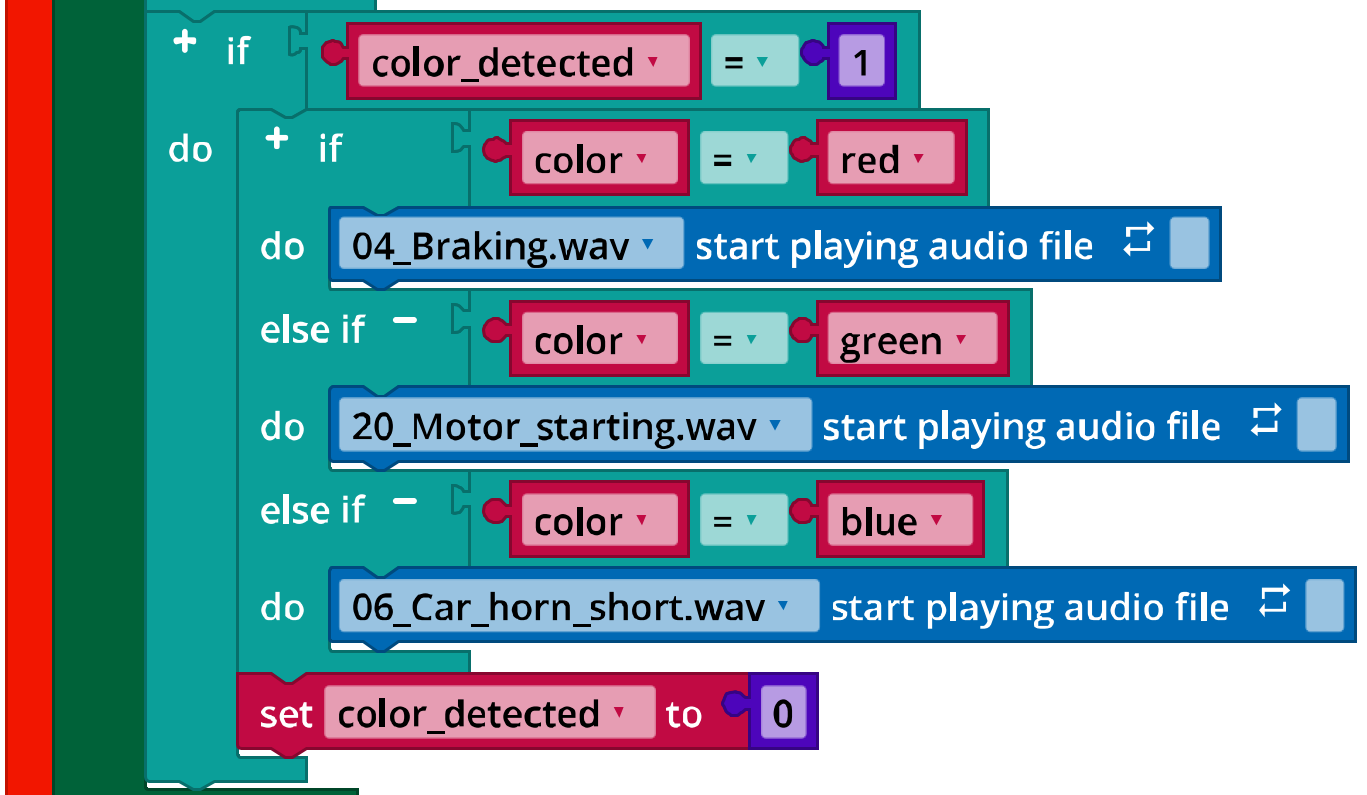
Anschluss der Kamera:



3b. Programmauszug (Beispiel):



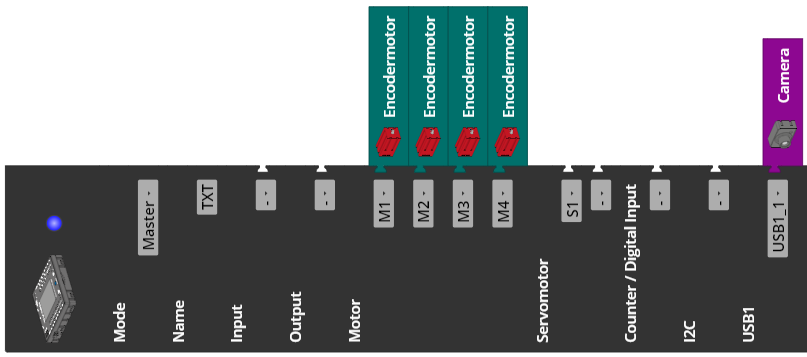




*Mecanum\_Line\_Follower\_with\_Color\_Detection\_digital.ft*

## Experimentieraufgaben

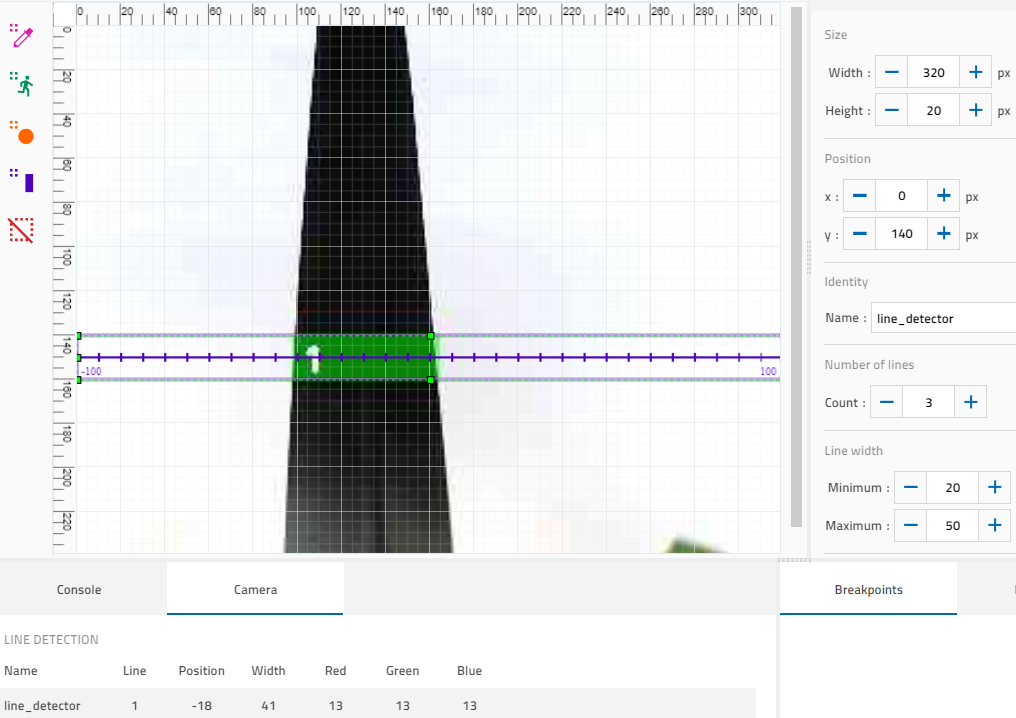
Anschluss der Sensoren:



**1. Spurfolger mit Proportionalregler**

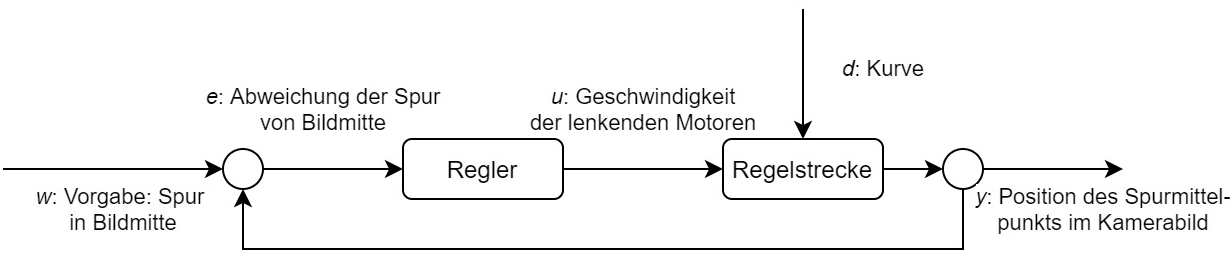
Der Liniendetektor der Kamera liefert die Abweichung von der Mitte des Erkennungs­fensters.

Konfiguration der Linienerkennung:



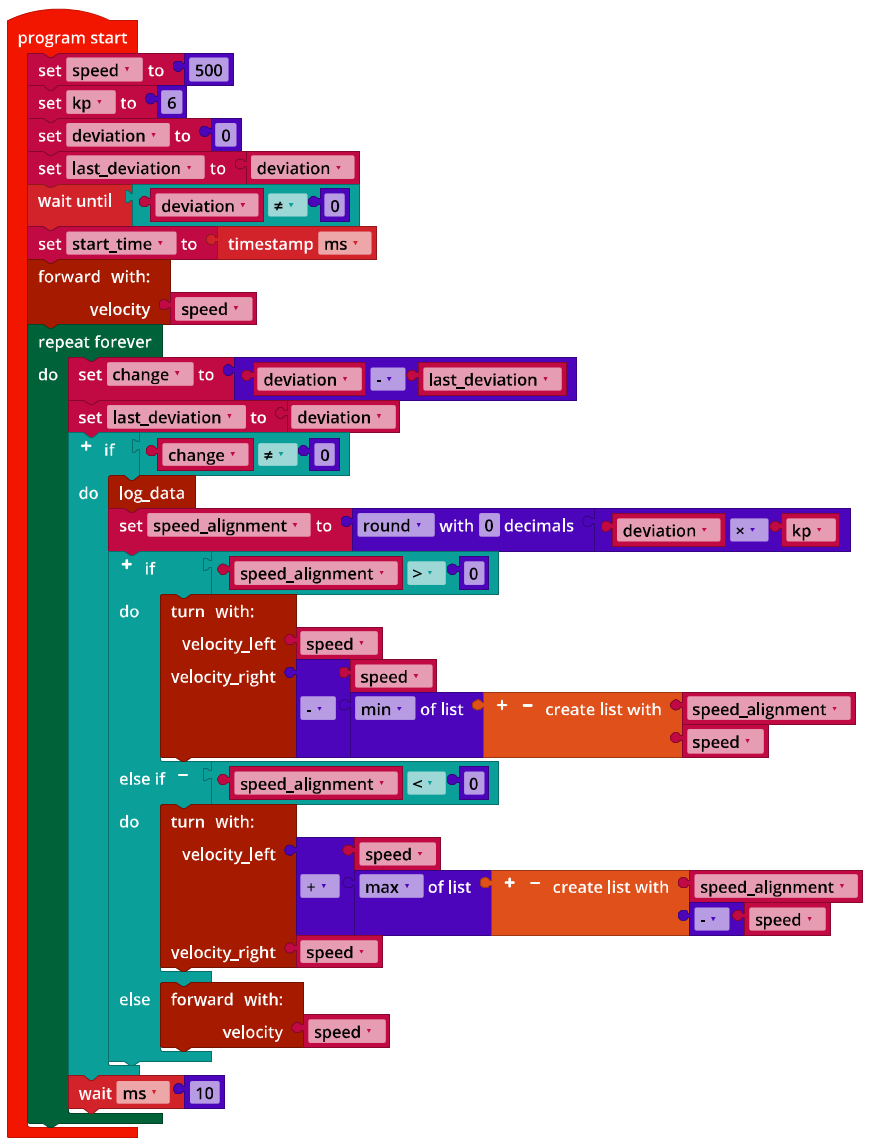
Da Farbflächen als Linie erkannt werden können, sollte die Linienerkennung auf mindestens zwei Linien konfiguriert werden.

1a. Regelkreis:

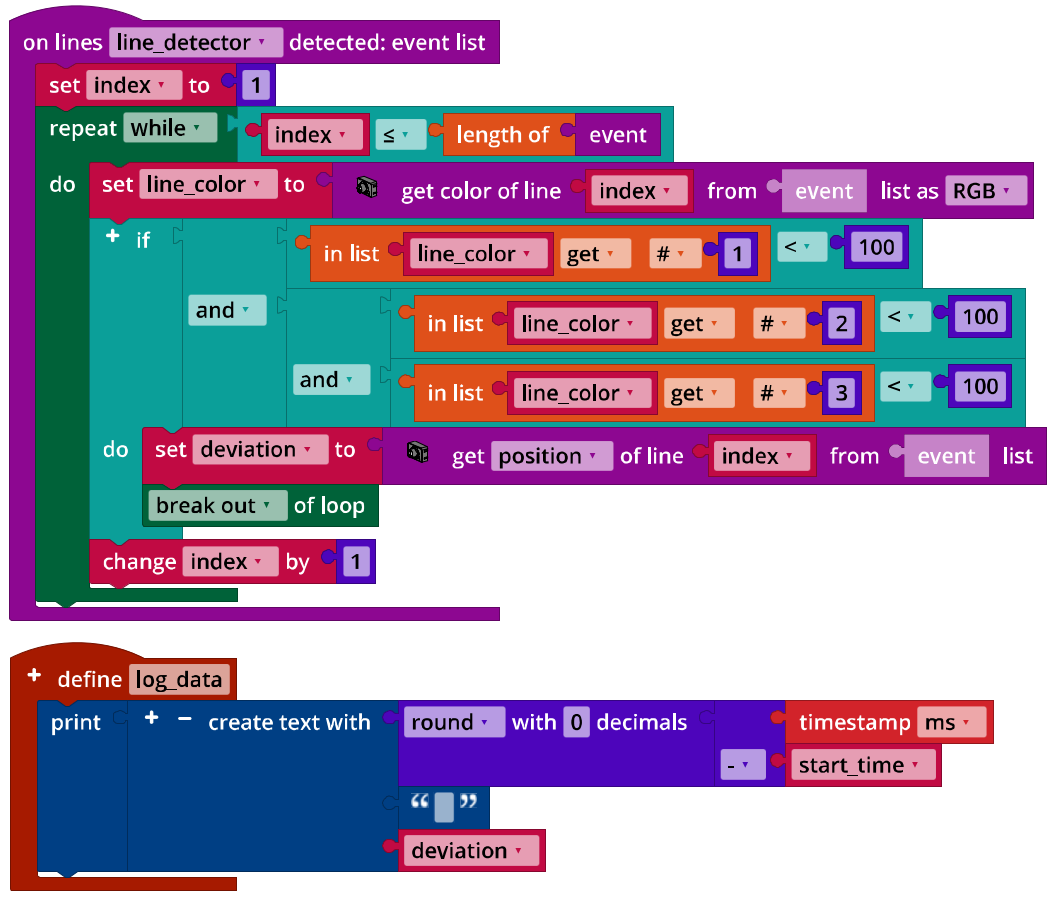


*Regelkreis\_Mecanum\_Spurfolger.drawio*

1b. Programmauszug (Beispiel):



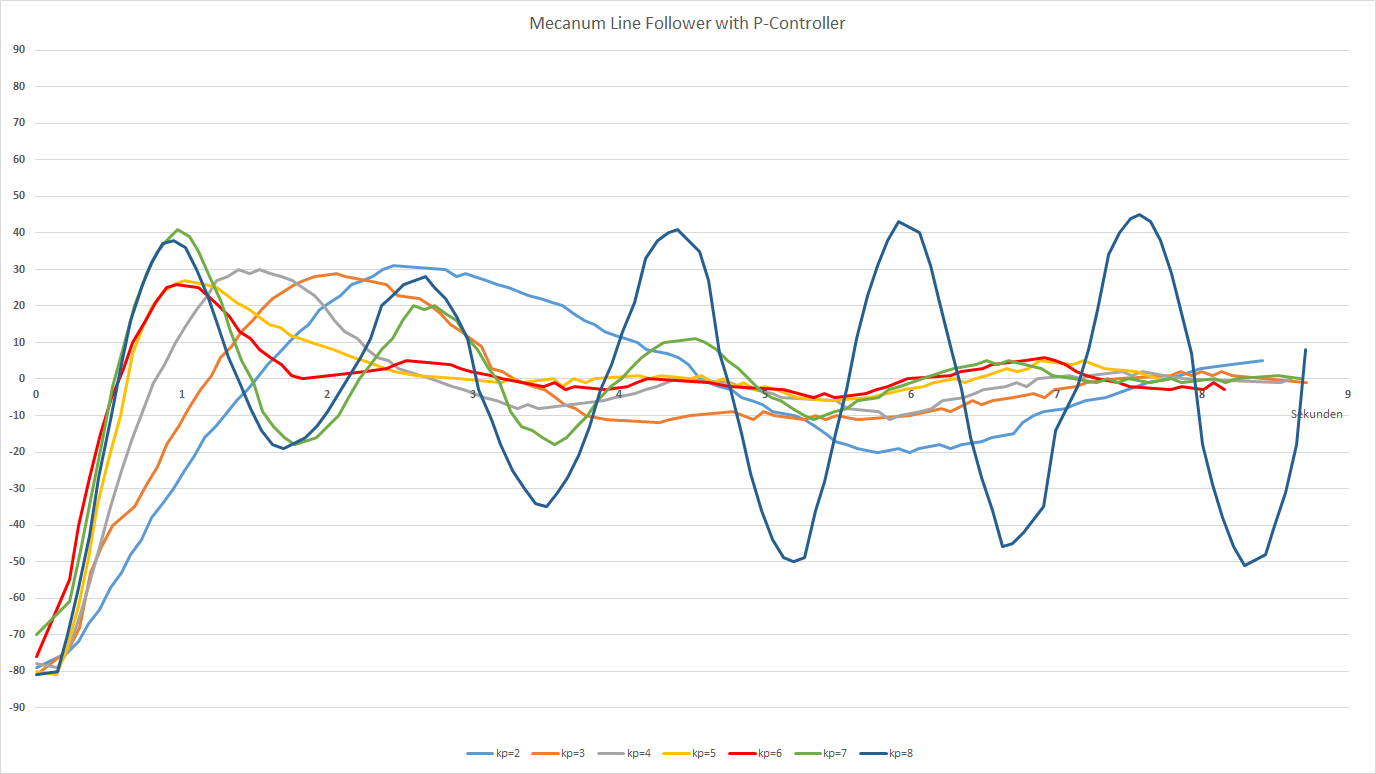
*Mecanum\_Line\_Follower\_P\_Controller.ft*



*Mecanum\_Line\_Follower\_P\_Controller.ft*

Multipliziert mit dem Proportionalitätsfaktor wird die Position so zu der Motorge­schwindigkeit des Links- bzw. Rechtsabbiegens (Funktion *turn*) addiert bzw. sub­trahiert, dass mit der Änderung der Fahrtrichtung die Abweichung von der Spur abnimmt. Je nach vertikaler Positionierung des Liniendetektors in der Kamera-Konfiguration müssen ggf. die minimale und maximale Linienbreite im Programm angepasst werden.

1c. Messergebnisse des P-Reglers (mit :



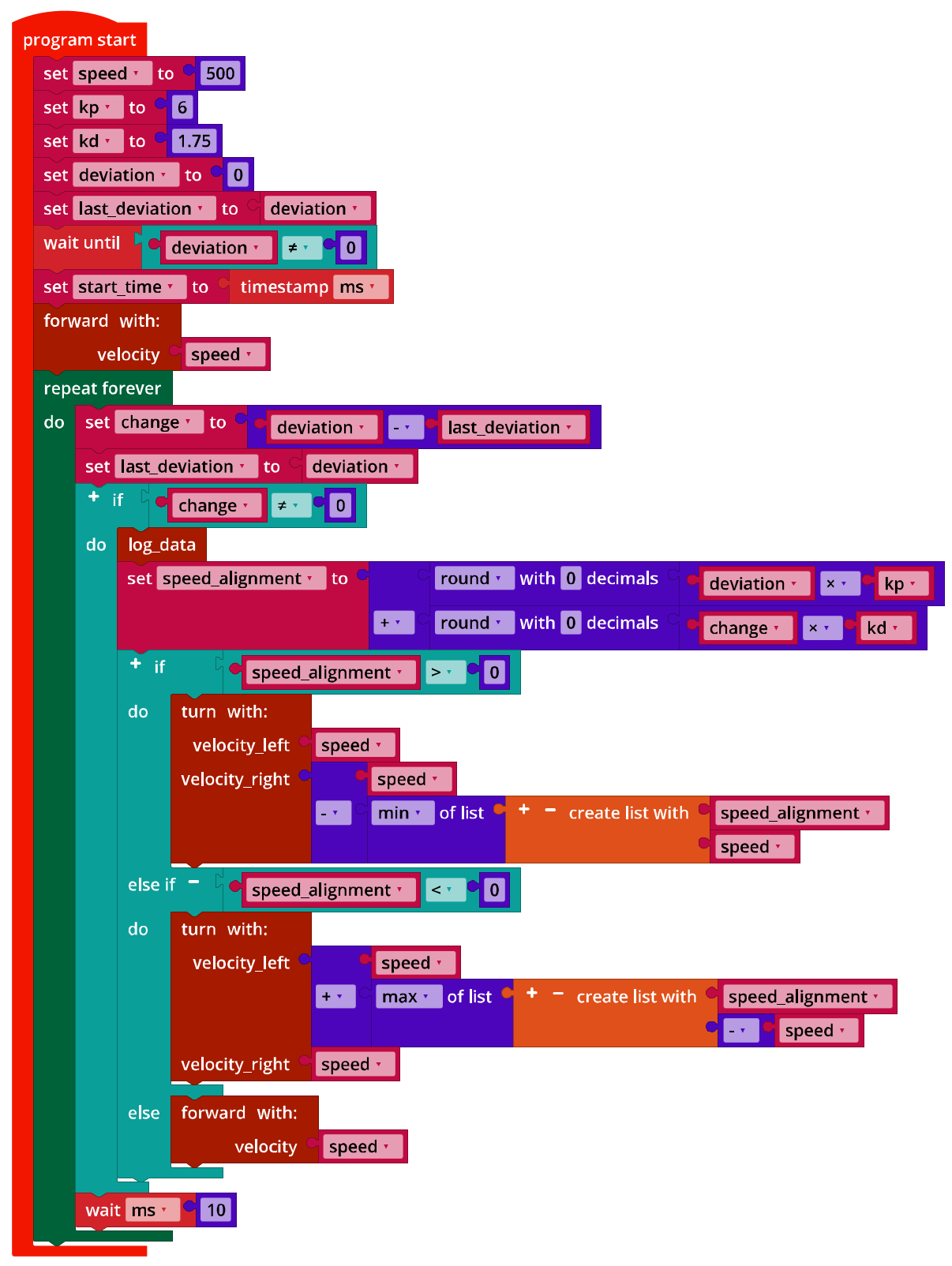
*Mecanum\_Line\_Follower\_with\_P\_Controller\_Results.jpg*

Die rote Linie mit schwingt nach ca. 1,7 Sekunden am schnellsten ein, ohne ein zweites Mal überzuschwingen. Mit (dunkelblaue Linie) oszilliert der Regler.

**2. Spurfolger mit PD-Regler**

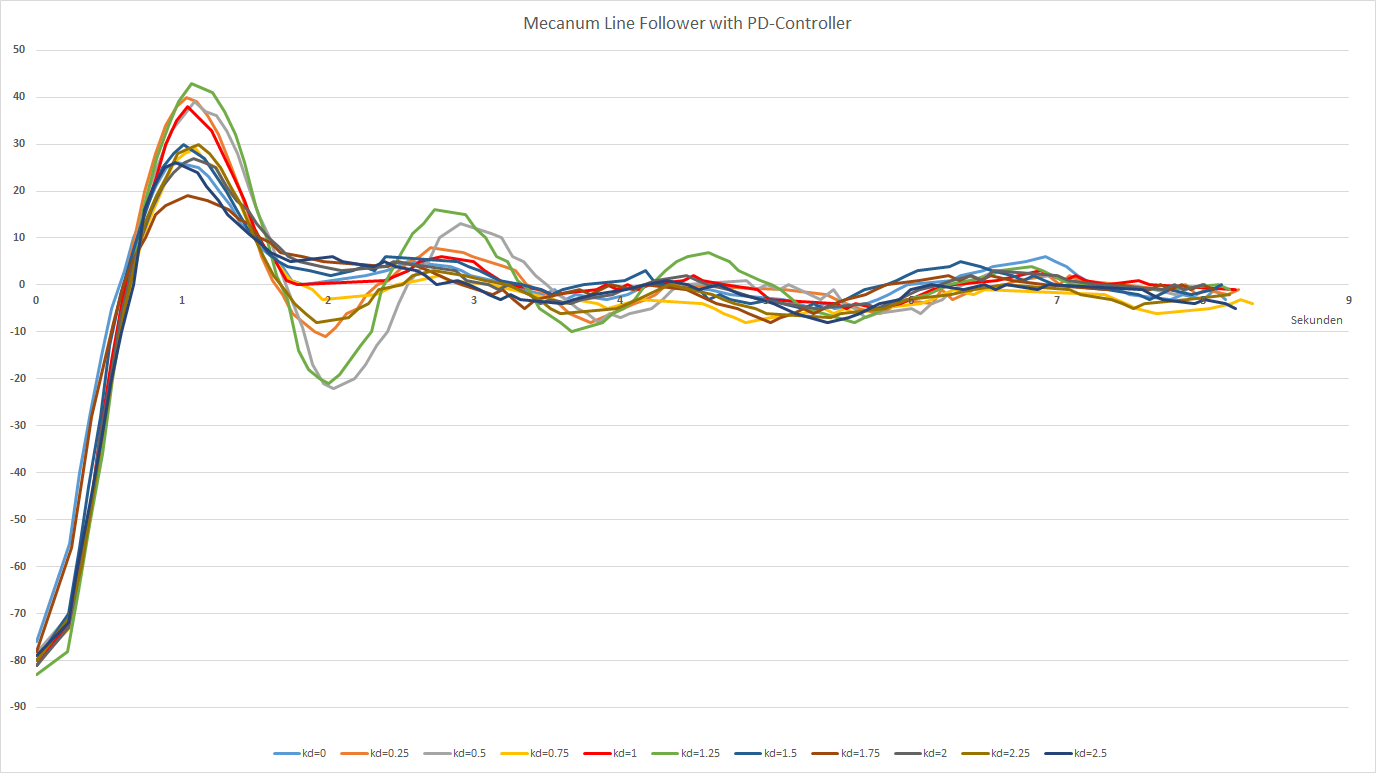
Das D-Glied des PD-Reglers ist die Änderung der Abweichung multipliziert mit dem Differenzial-Faktor .

Programm (Beispiel):



*Mecanum\_Line\_Follower\_with\_PD\_Controller.ft*

Messergebnisse des PD-Reglers  
(mit :



*Mecanum\_Line\_Follower\_with\_PD\_Controller\_Results.jpg*

Mit dem Differential-Faktor war die Dämpfung des Überschwingens in den durchgeführten Tests am wirksamsten. Der Wert kann abhängig von der Position der Linienerkennung im Kamerabild und kleinen konstruktiven Unterschieden von Modell zu Modell unterschiedlich ausfallen.

Anlagen

# Aufgabe 2: Spurfolger

## Erforderliches Material

* PC für Programmentwicklung, lokal oder über Web-Schnittstelle.
* USB-Kabel oder BLE- bzw. WLAN-Verbindung für die Übertragung des Programms auf den TXT4.0.
* Parcours-Bogen mit schwarzer, 2 cm breiter, gerader Linie
* Parcours-Bogen mit schwarzer, 2 cm breiter, geschlossener Kreislinie   
  (aus Robotics TXT 4.0 Base Set)
* Hindernis (Karton, Dose, …)

## Weiterführende Informationen

[1] FRC Team 2605 (Bellingham, WA): [*How a Mecanum Drive Works*](https://seamonsters-2605.github.io/archive/mecanum/). github.io

[2] Wikipedia: [*Endlicher Automat (Zustandsautomat)*](https://de.wikipedia.org/wiki/Endlicher_Automat)

[3] Ferdinand Wagner, Ruedi Schmuki, Thomas Wagner, Peter Wolstenholme: [*Modeling Software with Finite State Machines. A Practical Approach*](http://is.ifmo.ru/download/modelingsoftwarewithfinitestatemachinesapracticalapproach.pdf). Auerbach Publications, 2006.

[4] Online-Diagrammeditor zur Erstellung von Zustandsübergangsdiagrammen (Format drawio): <https://www.diagrammeditor.de/>

[5] Wikipedia: [*Regelungstechnik*](https://de.wikipedia.org/wiki/Regelungstechnik).

[6] Wikipedia: [*Regler*](https://de.wikipedia.org/wiki/Regler).

[7] RN-Wissen: [*Regelungstechnik*](https://rn-wissen.de/wiki/index.php/Regelungstechnik).

[8] Tim Wescott: [*PID without a PhD*](https://www.magentacloud.de/lnk/oyB5kKLU). Embedded Systems Programming, 10/2000, S. 86-108.