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

# Aufgabe 2

# Spurfolger

## Konstruktionsaufgabe

Für die dritte Programmieraufgabe und die Experimentieraufgaben benötigen wir die Kamera am Mecanum-Omniwheels-Fahrzeug „Basismodell mit Sensoren“. Schließe sie an die USB1-Buchse des TXT an.

**Wichtig**: Wenn der TXT gestartet ist, stellt sich der Servo automatisch auf „Geradeaus-Stellung“. Stecke den Servo-Hebel anschließend so auf, dass die Kamera in einem Winkel von etwa 45° gegenüber der Senkrechten nach vorne geneigt ist.

**Achtung**: Wenn du den Servo bei eingeschaltetem TXT mit der Hand bewegst, kannst du ihn beschädigen.

## Programmieraufgaben

**1. Spurfolger mit Spursensor**

Das Mecanum-Omniwheels-Fahrzeug soll nun mit Hilfe des Spursensors der schwarzen, ca. 2 cm breiten Spur des Parcours aus dem Robotics TXT 4.0 Base Set folgen (siehe auch Aufgabe 8 des Robotics TXT 4.0 Base Set).

1a. Erstelle zunächst ein Zustandsübergangs­diagramm, das die möglichen Zustände und das Verhalten des Mecanum-Omniwheels-Fahrzeugs beschreibt.

1b. Setze dein Zustandsübergangsdiagramm in ein Blockly-Programm um. Verwende dafür eine Zustandsvariable, deren Zustandswert du aus den Werten der IR-Sensoren bestimmst.

Teste das Programm an dem kreisförmigen Parcours desRobotics TXT 4.0 Base Set.

1c. Wie lässt sich die Geschwindigkeit des Mecanum-Omniwheels-Spurfolgers erhöhen? Mache Versuche und messe die Zeit, die der Spurfolger jeweils für die Bewältigung des Parcours benötigt.

|  |  |
| --- | --- |
| Lösungsvariante (Anpassungen, Parameter) | Zeit |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**2. Spurfolger mit Hinderniserkennung**

Auf der Spur können sich auch Hindernisse befinden. Wenn der Ultraschallsensor ein Hindernis in maximal 2 cm Entfernung vor dem Fahrzeug entdeckt, soll sich das Fahr­zeug seitwärts bewegen, am Hindernis vorbeifahren und dann wieder auf die Spur zurückfinden.

Überlege dir verschiedene Lösungsvarianten für das Finden der Linie. Erweitere dein Blockly-Programm ent­sprechend und teste die Varianten. Welche Vor- und Nachteile haben sie?

**3. Spurfolger mit Farbsteuerung**

In den Ecken des Parcours erkennst du vier Farbflächen. Mit diesen Farbflächen kannst du deinem Mecanum-Omniwheels-Fahrzeug Kommandos geben.

3a. Erweitere dein Blockly-Programm um eine Farberkennung. Lass‘ dir zur Kali­brierung der Erkennung den RGB-HEX-Farbcode der erkannten Farbe auf der Konsole ausgeben.

3b. Lege fest, bei welchen Farben (mindestens zwei) das Mecanum-Omniwheels-Fahrzeug wie reagieren soll und passe dein Blockly-Programm entsprechend an.

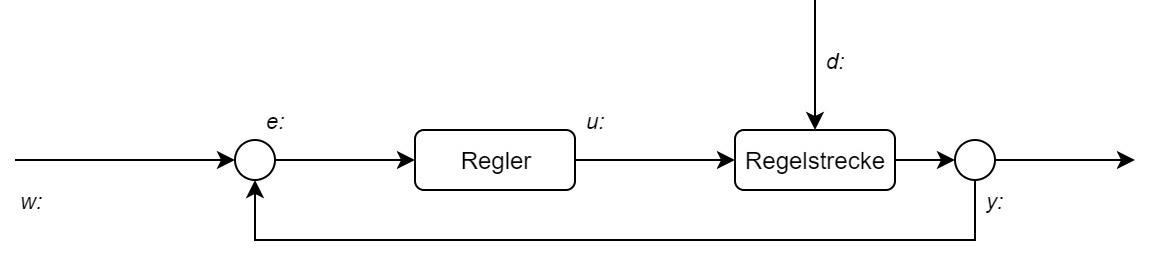
**Hinweis**: Wundere dich nicht, dass der auf der Konsole angezeigte HEX-Wert stark zu schwanken scheint – der Hue-Wert ist stabil. Du kannst jeden der angezeigten HEX-Farbcodes für die Erkennung verwenden.

## Experimentieraufgaben

**1. Spurfolger mit Proportionalregler**

Der Mecanum-Omniwheels-Spurfolger soll nun wie der Spurfolger-Buggy aus Aufgabe 8 des Robotics TXT 4.0 Base Set mit einem Proportionalregler (P-Regler) ausgestattet werden. Dabei soll die Stärke der Richtungskorrektur von der Größe der Linienabwei­chung abhängen. Auch hier soll mit der Linienerkennungsfunktion der Kamera die Größe der Abweichung von der Spur bestimmt werden.

1a. Beschrifte zunächst den folgenden Regelkreis mit den entsprechenden Werten deines Mecanum-Omniwheels-Spurfolgers:



1b. Konzipiere und programmiere eine Proportionalregelung (P-Regler) für den Mecanum-Omniwheels-Spurfolger. Dabei kannst du das Erkennungsverfahren für die schwarze Linie aus der Experimentier­aufgabe 1 von Aufgabe 8 des Robotics TXT 4.0 Base Set verwenden.

Teste dein Programm an dem einfachen geraden Linien-Parcours, indem du den Mecanum-Omniwheels-Spurfolger parallel zur Spur so startest, dass der Mittelpunt der Spur möglichst weit links im Bild erscheint.

**Tipp**: Beginne mit dem Proportionalitätsfaktor . Erhöhe den Wert schrittweise um 1, bis der Mecanum-Omniwheels-Spurfolger sich zügig „einschwingt“, d. h. die Abweichung von der Spur sich schnell verringert, ohne dass der Regler auf- oder stark überschwingt.

1c. Ergänze in deinem Programm eine Textausgabe, die nach jeder Änderung der Abweichung von der Spur

* die Zeit (in ms), die seit dem Start des Programms verstrichen ist, und
* den Wert der aktuellen Abweichung

durch ein Leerzeichen getrennt auf der Konsole ausgibt.

Kopiere nach jeder Testfahrt mit einem anderen Wert für die Konsolen-Ausgaben in eine Tabellenkalkulation und lass‘ dir die Werte grafisch in einem Diagramm (x: Zeit, y: Abweichung von der Spur) anzeigen. Passe den Proportionalitätsfaktor so lange an, bis sich der Kurven­verlauf schnell ein­schwingt.

Teste den Spurfolger mit dem kreisförmigen Parcours aus dem Robotics TXT 4.0 Base Set. Eventuell musst du dafür die Höchstgeschwindigkeit etwas senken.

**2. Spurfolger mit PD-Regler**

Wie beim Buggy in Aufgabe 8 des Robotics TXT 4.0 Base Set kannst du durch eine Erweiterung des Reglers um ein „D“-Glied (Differential-Anteil), das die Größe der Veränderung der Abweichung von der Spur bei der Geschwindigkeitskorrektur berück­sichtigt, das Überschwingen weiter dämpfen.

Erweitere den P-Regler deines Mecanum-Omniwheels-Spurfolgers aus Experimentieraufgabe 1 entsprechend zu einem PD-Regler. Führe Testfahrten mit unterschiedlichen Werten für den Differential-Faktor durch und lass‘ dir die Daten in einem Tabellen­kalku­lationsprogramm grafisch anzeigen.

**Tipp**: Beginne deine Tests mit dem Differential-Faktor und erhöhe dessen Wert so lange in Schritten von 0,25, bis das Überschwingen des P-Reglers gut gedämpft wird..

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.