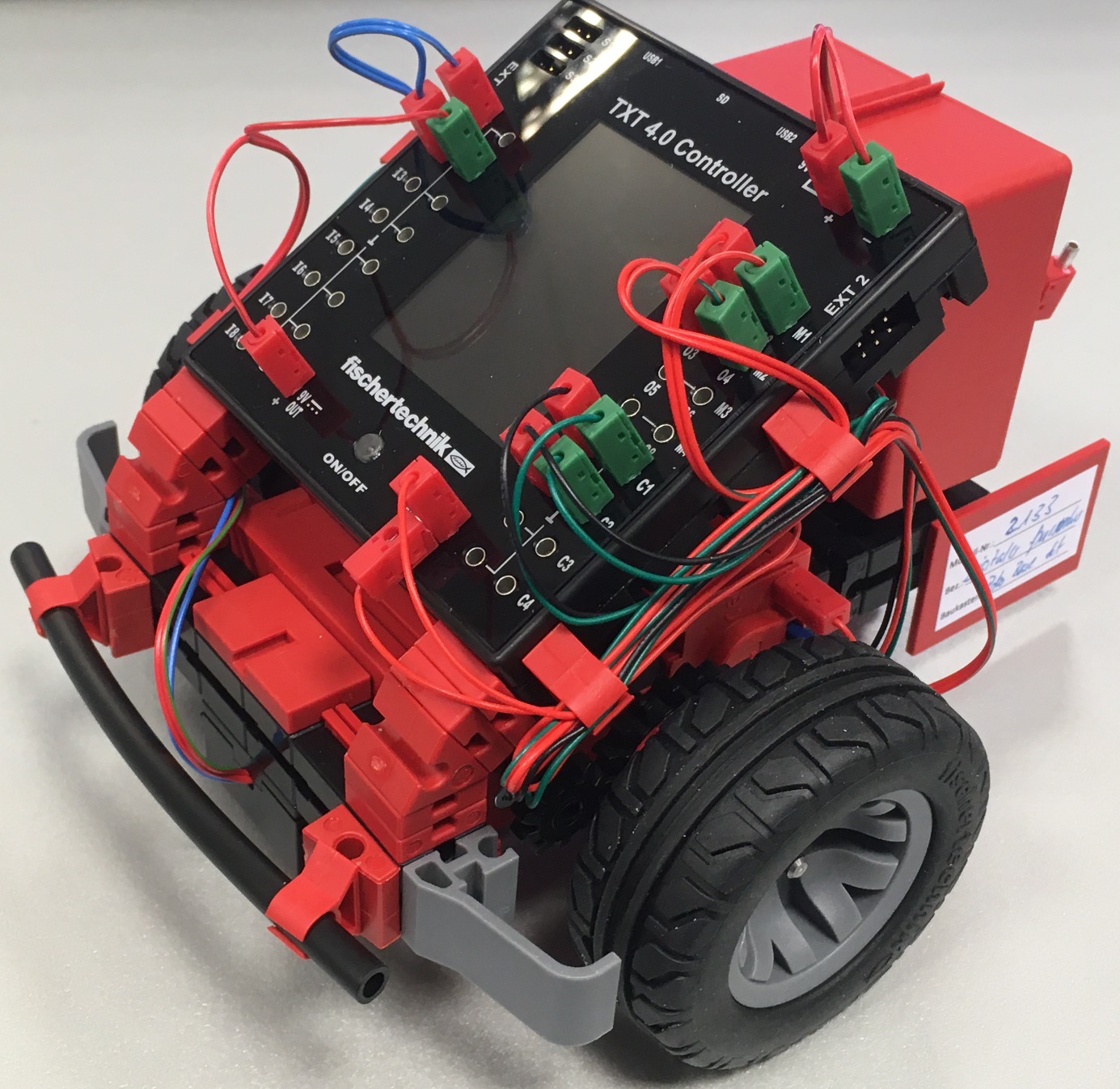
Aufgabe 8:

# Spurfolger

## Konstruktionsaufgabe

Baue den Spurfolger-Buggy nach der Bauanleitung. Schließe die Motoren (M1, M2) und die IR-Sensoren (I1, I2) an den TXT an. Die IR-Sensoren müssen über den 9V-Spannungsausgang des TXT mit Strom versorgt werden (siehe Schaltplan).



Prüfe mit dem Interface-Test, ob die Polung der Motoren korrekt ist. Wenn du den Spursensor von rechts nach links über die schwarze Linie des Parcours führst, müssen die an den Eingängen anliegenden Werte der IR-Spursensoren, die im Interface-Test angezeigt werden, von „1“ (weißer Untergrund) auf „0“ (schwarze Spur) und wieder auf „1“ (weißer Untergrund) wechseln – erst der linke (I1), dann der rechte (I2).

## Programmieraufgaben

**1. Spurfolger mit Spursensor**

Der Buggy soll nun mit Hilfe des Spursensors (zwei nebeneinander liegende Infrarot-Sensoren) der schwarzen Spur des Kreis-Parcours folgen. Verliert der Buggy die Spur, soll er automatisch zurück auf die Spur finden. Die Spur ist durchgehend und einheitlich 2 cm breit.

**Tipp**: Sieh‘ dir die Aufgabe zunächst mit den „Augen“ des Buggys an und mache eine Fallunterscheidung: Welche Fälle muss der Buggy erkennen, und was soll er in dem jeweiligen Zustand tun?

1a. Stelle deine Lösung als Zustandsübergangsdiagramm dar.

1b. Schreibe das Blockly-Steuerungsprogramm für den Buggy und teste es in beiden Fahrtrichtungen.

**2. Optimierter Spurfolger**

Du erkennst, dass dein Buggy stark „zappelt“: Das kostet Geschwindigkeit. Wie kannst du die Geschwindigkeit deines Buggys erhöhen?

Mache verschiedene Versuche, stoppe und notiere jeweils die Zeit für die Bewältigung des Parcours und ziehe Schlüsse aus den Ergebnissen.

|  |  |
| --- | --- |
| Lösungsvariante: Anpassungen, Wahl der Parameter | Zeit |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**3. Spurfolger mit Hinderniserkennung**

Nun wollen wir den Buggy mit zusätzlichen Fähigkeiten ausstatten. Dazu spendieren wir ihm einen Ultraschall-Abstandssensor. Befestige ihn vorne am Buggy und schließe den Datenausgang (schwarzes Kabel) an I3 an, die Stromversorgung an den 9V-Spannungsausgang (siehe Bauanleitung).

Wenn der Buggy mit dem Ultraschall-Sensor ein Hindernis in einer Entfernung von weniger als 10 cm erkennt, soll er ihm ausweichen und anschließend die Spur wiederfinden.

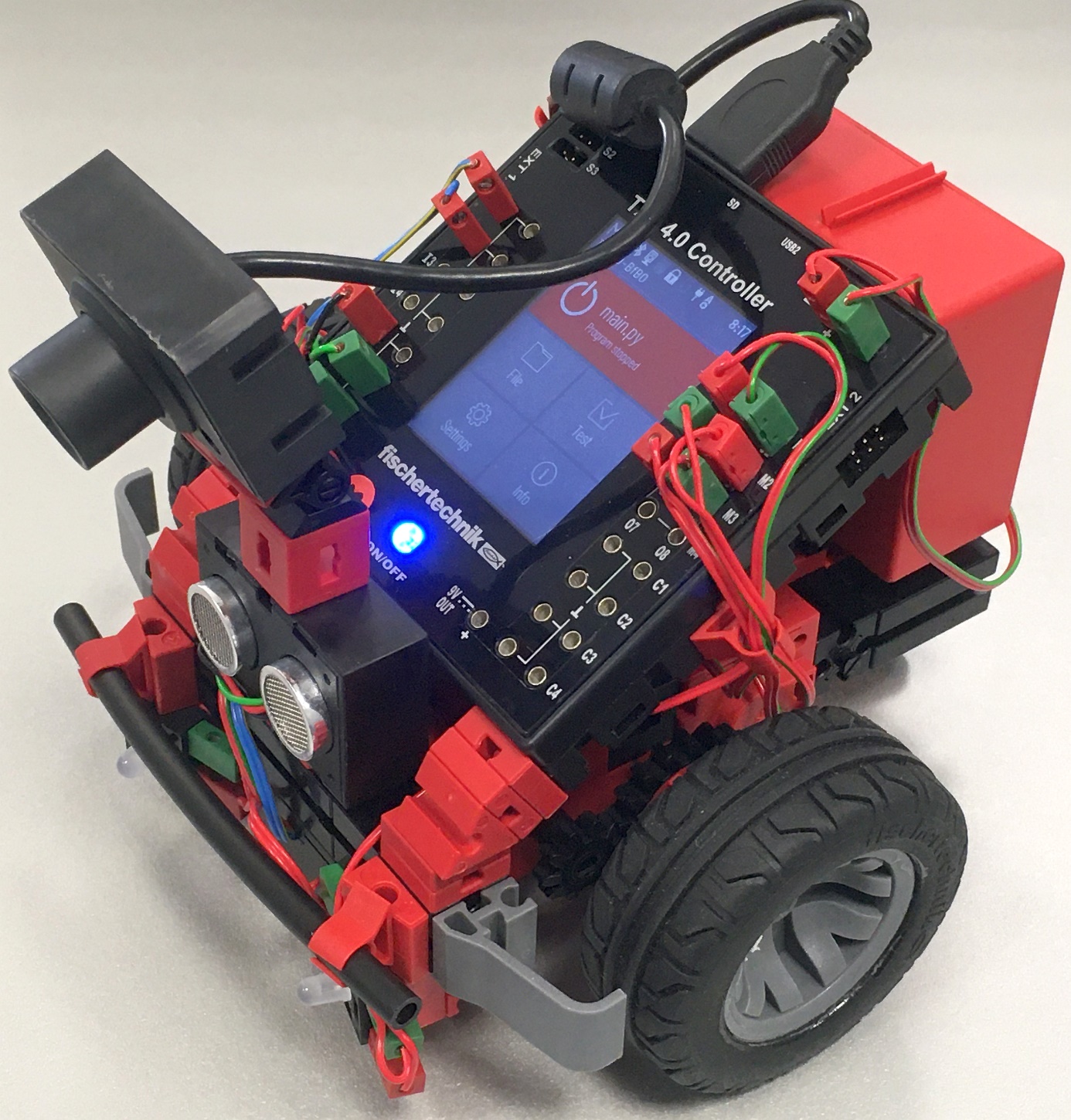
2a. Überlege dir eine „Hindernis-Umfahren-Strategie“ und erweitere zunächst dein Zustandsübergangsdiagramm.

2b. Ergänze anschließend dein Blockly-Programm entsprechend.

**4. Steuerung mit Farbflächen**

Jetzt statten wir den Buggy außerdem mit der Kamera und zwei Frontstrahlern aus. Die Kamera wird an den USB1-Anschluss und die beiden LED werden parallel an Ausgang O5 angeschlossen.

Befestige außerdem den Fototransistor seitlich am Buggy und schließe ihn an I4 an.



Links und rechts des Parcours erkennst du mehrere Farbflächen. Mit Hilfe der Kamera kannst du dem Buggy über diese Farbflächen Kommandos geben – ihn wenden, blinken oder ein Geräusch von sich geben lassen.

4a. Lege fest, bei welchen Farbflächen der Buggy wie reagieren soll.

4b. Schreibe ein kleines Hilfsprogramm, mit dem du den HEX-Code der Farben der Farbflächen bestimmen kannst.

4c. Erweitere anschließend dein Blockly-Programm so, dass der Buggy, während er der Spur folgt, die Farbe der Farbflächen erkennt und darauf entsprechend reagiert.

**Tipp**: Stelle die Farberkennung auf die ermittelten HEX-Codes ein und wähle eine Toleranz von 1. Wenn es bei deinen Tests zu Fehlerkennungen kommt, musst du diesen Wert verringern.

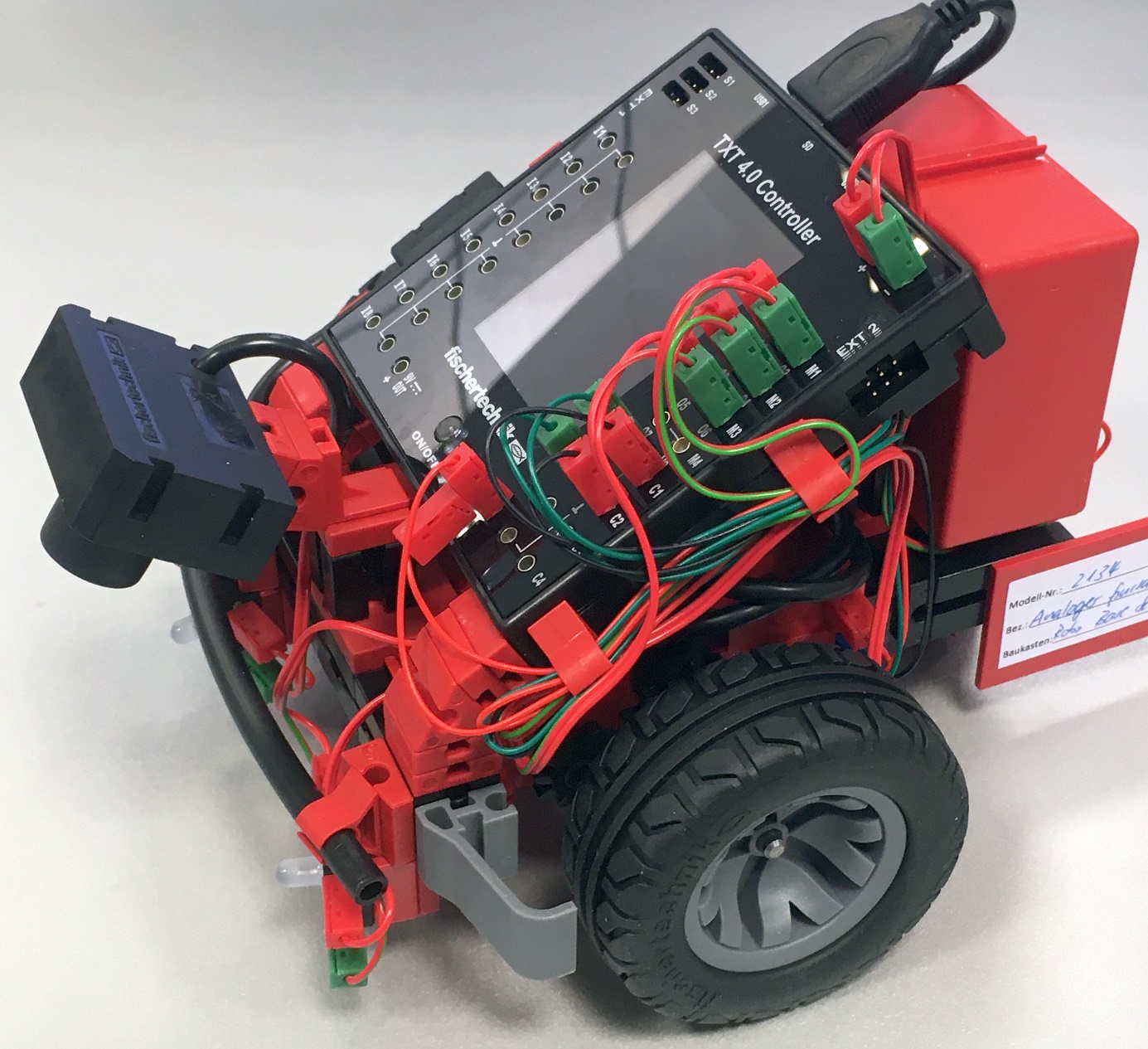
4d. Ergänze den Buggy um einen „Dämmerungsschalter“, sodass abhängig von dem Fototransistor die beiden Frontstrahler automatisch aktiviert bzw. deaktiviert werden.

## Experimentieraufgaben

Wir können die Steuerung des Buggys auch ausschließlich mit der Kamera vorneh­men. Dazu verzichten wir nun bei unserem „analogen Spurfolger“ auf alle anderen Sensoren (Ultraschall, Spursensor).

Baue den Spurfolger-Buggy entsprechend um.

Mit der Linienerkennungsfunktion der Kamera können wir den Buggy nun mit einem Regler ausstatten.



**1. Spurerkennung mit Kamera**

Mit der Linienerkennungsfunktion der Kamera können wir die Lage der Spur fest­stellen. Da sich links und rechts der Spur Farbflächen befinden, die von der Kamera ebenfalls als Linie erkannt werden können, müssen wir die Farbe der von der Kamera erkannten Spur auswerten.

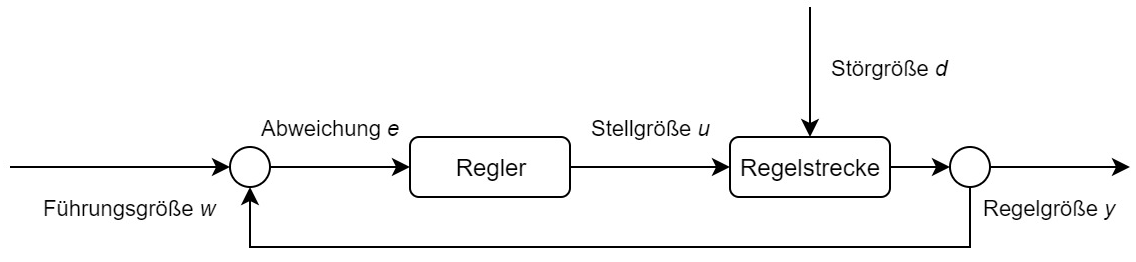
Schreibe ein Blockly-Programm, das mehrere Linien erkennen kann und die RGB-Farben der erkannten Linien auf der Konsole ausgibt.

Platziere den Buggy nun an verschiedenen Stellen des Parcours. An welchen Eigen­schaften lässt sich die schwar­ze Spur einfach und zuver­lässig erkennen?

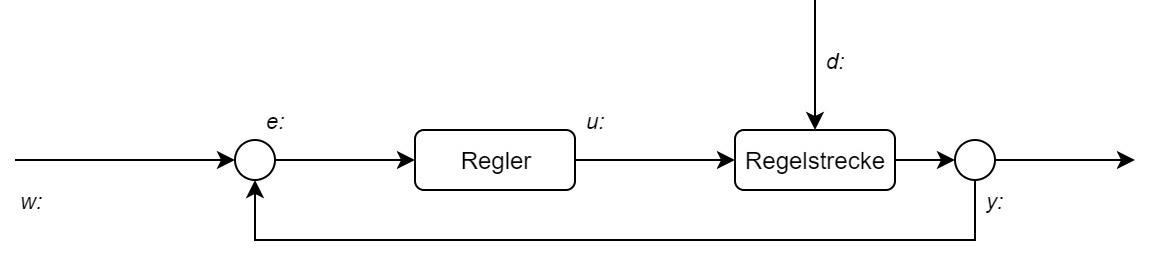
**2. Spurfolger mit P-Regler**

Jetzt wollen wir den Buggy mit einem einfachen Proportional-Regler automatisch der Spur folgen lassen.

Betrachte dazu den Spurfolger als Regelkreis:



2a. Welche Größen des Spurfolgers entsprechen den Werten *w*, *e*, *u*, *d* und *y* im Regelkreis? Beschrifte die folgende Abbildung entsprechend:



2b. Schreibe ein Blockly-Programm, das den Buggy mit einem Proportional­regler (Pro­portionalitätsfaktor ) entlang der Spur des Parcours steuert.

Verwende dafür die in Experimentieraufgabe 1 entwickelte Spurerkennung.

**Tipp**: Wähle als Maximalgeschwindigkeit der Motoren 350. Reduziere die Geschwin­digkeit des jeweils für die Richtungskorrektur zuständi­gen Motors (wenn links von der Spur: rechter Motor, wenn rechts von der Spur: linker Motor) alle 20 ms um .

Beginne deine Experimente zur Einstellung des Proportio­nalitätsfaktors an einem geraden Abschnitt der Spur mit und vergrößere den Wert in kleinen Schritten so lange, bis der Regler ohne zu oszillieren der gesamten Spur folgt.

**3. Spurfolger mit PD-Regler**

Der Proportionalregler korrigiert die Fahrtrichtung relativ langsam, daher kann es in einer engen Kurve dazu kommen, dass der Buggy die Spur verlieren.

Mit einem „D“-Glied (Differential-Faktor ), das die Größe der Veränderung der Spurabweichung misst, kann der Regler verbessert werden: Auf plötzliche starke Veränderungen der Richtung der Spur (enge Kurve) reagiert der PD-Regler schneller und dämpft gleichzeitig das Überschwingen.

3a. Lass‘ dir die Regler-Daten (Abweichung von der Spur, proportionaler Korrektur­wert) deines Proportionalreglers aus Experimentieraufgabe 2 auf einer Fahrt entlang der Spur auf der Konsole ausgeben und kopiere diese Daten anschließend in ein Tabellenkal­kulationspro­gramm:

* Lass‘ dir die Daten des Reglers grafisch anzeigen.
* Ergänze in der Tabelle eine Spalte, die die Differenz der Spurabweichung (aktuelle Abweichung minus letzte Abweichung) bestimmt.
* Berechne in einer weiteren Spalte den Korrekturwert eines PD-Reglers zu einem vorgegebenen Differential-Faktor . In der Grafik erkennst du den Einfluss des Faktors auf die Richtungskorrektur.
* Wähle einen Wert für deinen PD-Regler, der dir geeignet erscheint.

3b. Erweitere den P-Regler deines Spurfolgers zu einem PD-Regler. Überprüfe durch Testläufe, ob der in der Simulation in Aufgabenteil 3a. gewählte Wert geeignet ist. Korrigiere ihn, wenn nötig, in kleinen Schritten.

**Tipp**: Auch kleine „Ausschläge“ des Reglers kannst du gut erkennen und bei der Einstellung von berücksichtigen, wenn du die Werte des Reglers auf der Konsole ausgibst und in dem Tabellenkalkulationsprogramm auswertest.

Anlagen

# Aufgabe 8: 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 geschlossener Kreislinie und Farbflächen
* Hindernis (Karton, Dose, …)

## Weiterführende Informationen

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

[2] 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.

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

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

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

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

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