

Kalis Bastelecke

DIESMAL: ROBOTICS – DER SPURFOLGER

Eine klassische Aufgabe in der Robotik ist ein sogenannter Spurfolger: ein Roboter, der wie von Geisterhand gesteuert einer schwarzen Linie folgt. Trocken ausgedrückt ist er ein einfaches Beispiel eines ersten Regelkreises, also einem System, welches sich selbst reguliert. Hört sich nicht spannend an. Aber wenn Du zum ersten Mal einen selbst gebaut hast und das Kerlchen über den Parcours flitzt, ist das tatsächlich ein Riesenspaß und ganz leicht umzusetzen.

Fischertechnik hat hierfür einen tollen Sensor in einigen Robotics-Kästen, den sog. IR-Spursensor. In diesem Beispiel beschreibe ich, wie man den Spursensor am TXT einsetzen kann. Das Prinzip funktioniert aber an allen Fischertechnik-Controllern gleich, also auf dem BT-Smart, TX oder ftDuino. Was macht unser Sensor eigentlich genau? Wenn man ihn genau betrachtet, stellt man fest, dass es sich eigentlich um zwei Sensoren handelt - zwei Fototransistoren nebeneinander die auf Infrarotlicht schalten. Daneben ist jeweils eine LED, die Infrarotlicht aussendet. Er hat vier Anschlusskabel: ein rotes und ein grünes für die Stromversorgung und zwei Kabel die den Wert „0“ oder „1“ an jeweils einen der Inputs des Controllers liefern (I1-I8).

Für den Test habe ich den Spurfolge-Papierbogen (Abb. 2) von Fischertechnik aus dem Kasten Minibots genommen. Er hat eine schöne schwarze Linie, welche das Licht gut absorbiert. Du kannst Dir die Spur auch schön mit dem Drucker ausdrucken und mehrere Ausdrücke hintereinanderlegen, für eine beliebig lange Spur.

Wird der Sensor am Fahrzeug vorne so angebracht, dass ein „Auge“ rechts und eines links auf der Linie ist, geben die Sensoren jeweils eine „0“ zurück, da die schwarze Linie das IR-Licht absorbiert, also nicht nach oben reflektiert. Die IR-Fototransistoren sperren, es fließt kein Strom. Prüfen kannst Du das in RoboPro mit dem Interface-Test:

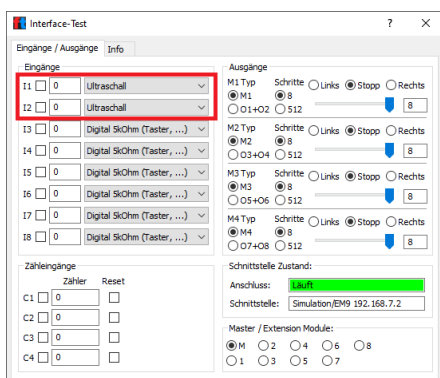


Abbildung 1: Interface-Test

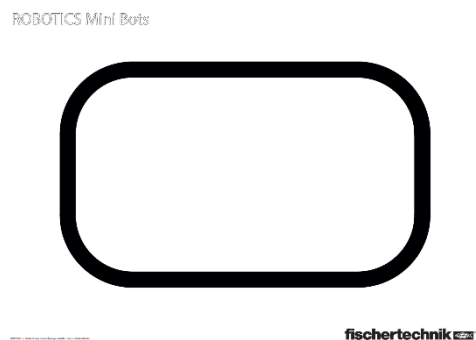


Abbildung 2: DIN A2 Parcours-Mini Bots (156 498)

Wenn Du jetzt dein Fahrzeug von der Linie runternimmst und so platzierst, dass beide Sensoren auf eine weiße Fläche schauen, sollte der Interface-Test zweimal die „1“ zeigen:

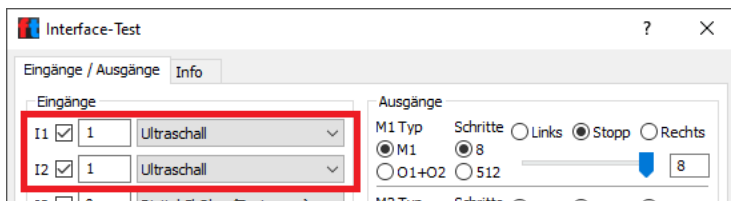


Abbildung 3: Sensor auf weißer Fläche

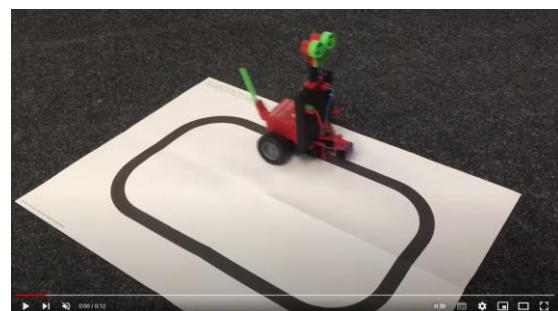
Ach ja, das Fahrzeug: Es gibt zwar viele großartige Varianten, die Du ausprobieren kannst! Für den Anfang empfiehlt sich aber erstmal ein Fahrzeug mit einem Differentialantrieb: also zwei unterschiedlich ansteuerbare Räder rechts und links auf einer gemeinsamen geometrischen Achse. Oder gleich ein Raupenantrieb, also Raupenkettens rechts und links am Fahrzeug.

Damit das mit dem Spurfolgen klappt, schauen wir uns die Fälle an, die das System annehmen kann. Der Informatiker nennt das auch gerne „Zustände“, wenn er das Modell als einen „endlichen Zustandsautomaten“ betrachtet:

1. Zustand ($S=Z_0$): beide Sensoren liefern „0“. Das heißt der rechte und der linke Sensor sehen schwarz, somit ist das Fahrzeug auf der Linie. Alles gut, darum schalten wir beide Motoren auf vorwärts, mit der gleichen Geschwindigkeit. Das Fahrzeug fährt gerade vorwärts.
2. Z_1 : Der linke Sensor liefert „0“ und der rechte „1“: jetzt ist der Roboter zu weit rechts und muss nach links korrigieren. Also links Stopp und rechts mit normaler Geschwindigkeit vorwärts.
3. Z_2 : Der rechte Sensor liefert „0“ und der linke „1“: das Fahrzeug ist leicht von der Linie nach links abgekommen. Wir schalten den linken Motor mit der Standardgeschwindigkeit wieder nach vorne, den rechten Motor stoppen wir. Dadurch dreht sich der Roboter nach rechts.
4. Z_3 : Der letzte Fall ist, dass beide Sensoren weiß – also „1“ sehen: das kommt vor, wenn der Roboter die Spur verliert, z.B. wenn er zu schnell ist und somit aus der Kurve fliegt. Eine Lösung kann sein, einen „Suchmodus“ aufzurufen, also den kleinen Kerl beispielsweise auf der Stelle drehen lassen. Also die Motoren in gegenläufiger Richtung anzutreiben.

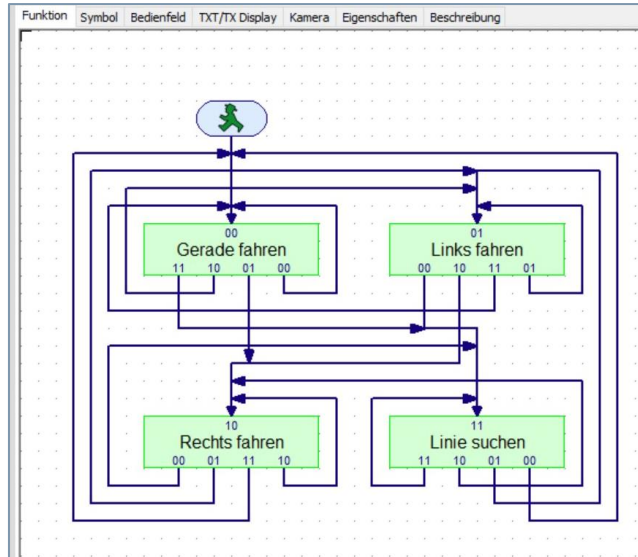
Wie das in der Praxis aussehen kann, sieht man hier:

<https://youtu.be/thKPlhrM57I>

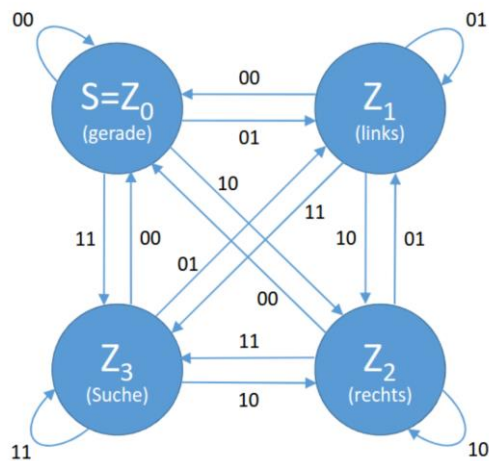


Wichtig ist, dass die Zustände sich sehr schnell ändern können, je nachdem, was sie Sensoren ausspucken. In jedem Zustand müssen ALLE Sensoren abgefragt werden und ggf. direkt in den entsprechenden anderen Zustand wechseln.

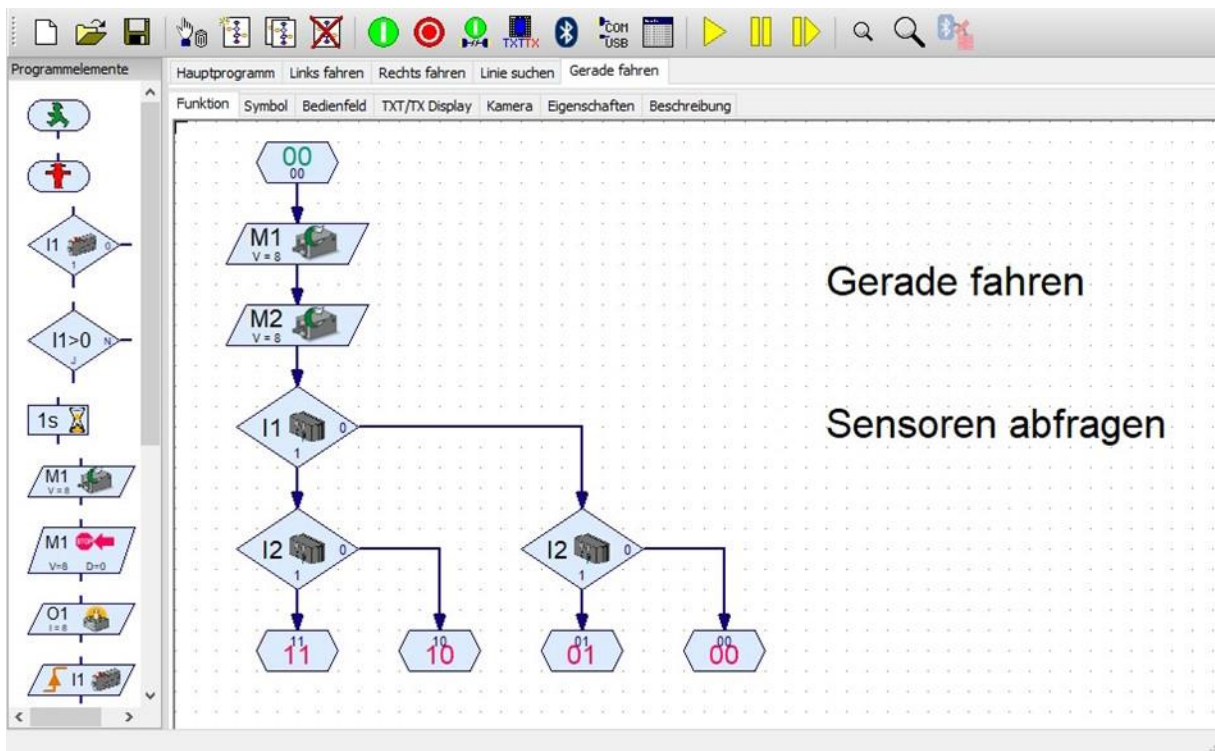
Das RoboPro-Programm kann dann so aussehen:



Auch wenn das Programm sehr kompliziert wirkt, ist es doch recht einfach aufgebaut. Das Zustandsdiagramm von Dirk Fox zeigt die (einfachen) Zusammenhänge:



Hinter jedem der grünen Blöcke versteckt sich ein Unterprogramm. Hier exemplarisch das Unterprogramm für „Gerade fahren“.



Alle Unterprogramme werde ich nicht zeigen, ein bisschen musst Du schon selbst nachdenken. Aber solltest Du nicht weiterkommen oder Fragen haben: einfach eine kurze Mail an technika@cyberforum.de – wir helfen gerne weiter :-)

Und jetzt ran an die Kästen und Spurfolger bauen!

Euer Kali