

Manual zum Programm für die zustandsbasierte Steuerung mobiler Roboter (ZUSMORO)

Tobias Meyer
Institut für Technische Informatik
Universität zu Lübeck

13. Januar 2010

Inhaltsverzeichnis

1	Vorwort	3
2	Grundlagen	3
2.1	Moore-Automat	3
2.1.1	Zustände	3
2.1.2	Transitionen	3
2.1.3	Bedingungen	4
2.2	Der e-puck	4
2.2.1	Sensoren	4
2.2.2	Aktoren	5
3	Benutzung von ZUSMORO	5
3.1	Die Werkzeuge	5
3.1.1	Erstellen von Zuständen	6
3.1.2	Erstellen von Transitionen	6
3.1.3	Löschen von Zuständen/Transitionen	7
3.1.4	Bewegen von Zuständen/Transitionen	7
3.2	Weitergehende Bedienung	8
3.2.1	Zuständen Aktionen zuweisen	8
3.2.2	Transitionen eine Bedingung zuweisen	8
3.3	Programmsteuerung	9
3.3.1	Verbinden zu einem Roboter	10
3.3.2	Verbindung trennen	10
3.3.3	Programmausführung beginnen	10
3.3.4	Programmausführung beenden	10
4	Schnellstart und Beispielprogramm	11
4.1	Zustände erstellen	11
4.2	Transitionen erstellen	11
4.3	Bedingungen einstellen	12
4.4	Programm sichern	12
4.5	Verbindung herstellen	13
4.6	Programm ausführen	13
4.7	Programmausführung abbrechen	13

1 Vorwort

Das vorliegende Programm (ZUSMORO) wurde im Rahmen der Bachelorarbeit “Entwicklung einer zustandsbasierten Steuerungssoftware für mobile Roboter” an der Universität zu Lübeck am Institut für Technische Informatik von Tobias Meyer geschrieben. Es ermöglicht zustandsbasierte Programme zu erstellen und mit diesen dann einen Roboter zu steuern. Im Folgenden wird die Funktionsweise des Programms anhand des e-puck Roboters exemplarisch beschrieben.

2 Grundlagen

ZUSMORO erlaubt das Erstellen von Programmen durch Angabe eines sog. Moore-Automaten der im folgenden Abschnitt näher erläutert wird. Der Automat wurde nach dem EVA-Prinzip¹ implementiert und arbeitet in der Standardeinstellung mit 20 Hz. Das bedeutet, dass die Sensordaten 20 mal pro Sekunde abgefragt und anhand derer die ggf. neuen Ausgaben bestimmt werden. Die mit ZUSMORO erstellten Programme werden nicht direkt auf dem Roboter ausgeführt, sondern laufen auf dem Rechner an dem sie auch erstellt werden. Der e-puck erhält Befehle per Bluetooth², die entweder Steuerbefehle für die Aktoren sind oder ihn auffordern neue Sensordaten zu übermitteln.

2.1 Moore-Automat

Ein Moore-Automat besteht im Grunde aus Zuständen und Übergängen (Transitionen) die im Zusammenhang mit ZUSMORO ein Programm repräsentieren, mit Hilfe dessen es möglich ist Roboter zu steuern.

2.1.1 Zustände

Ein Zustand bestimmt die Aktionen, die in ihm ausgeführt werden. Bei mobilen Robotern sind dies Befehle wie “vorwärts fahren”, “drehen” oder “LED an”. Das Programm betrachtet diese Aktionen dabei als 2-Tupel, dessen erste Komponente den Aktor und die zweite den Wert für diesen Aktor beschreibt.

Beim e-puck z.B. können Aktionen für die Antriebsmotoren, die LEDs und den “Beeper” spezifiziert werden. Ein Zustand, der “Startzustand”, nimmt dabei eine besondere Stellung ein, er wird ohne Übergang sofort zu Beginn der Ausführung angenommen.

2.1.2 Transitionen

Transitionen verbinden Zustände miteinander, sodass ein Wechsel von einem Zustand zu einem anderen stattfinden kann. An jede Transition ist dabei eine Bedingung geknüpft, die erfüllt sein muss, damit der Übergang von dem Ausgangszustand der Transition zum Zielzustand ausgelöst werden kann.

¹EVA ist die Abkürzung für Eingabe, Verarbeitung, Ausgabe

²Bluetooth ist ein Funkübertragungsstandard, der sich besonders für kleine Geräte eignet

2.1.3 Bedingungen

Die Bedingung, die an eine Transition geknüpft ist, wird durch einen Ausdruck der Aussagenlogik spezifiziert. Das Programm erlaubt dabei theoretisch jeden syntaktisch korrekten Ausdruck der Aussagenlogik als Bedingung zu verwenden, ist also bzgl. der Aussagenlogik vollständig. Für den Fall des e-pucks ist dies allerdings bewusst eingeschränkt um den Benutzer die Anwendung zu erleichtern.

2.2 Der e-puck

Der e-puck ist ein von dem Swiss Federal Institute of Technology in Lausanne (EPFL³) entwickelter mobiler Roboter, der speziell für den Einsatz in der Lehre entworfen wurde. Er verfügt über eine Vielzahl von Sensoren und seine Software ist frei verfügbar (open source), wodurch er sich gut auf die individuellen Bedürfnisse zuschneiden lässt. Besonderes Augenmerk verdient dabei die Bluetoothfähigkeit, durch die der e-puck kabellos sowohl gesteuert, wie auch programmiert werden kann.



Abbildung 1: e-puck Detailansicht

2.2.1 Sensoren

Im Folgenden seien kurz die vorhandenen Sensoren genannt, von denen in ZUSMORO allerdings nicht alle genutzt werden können:

- 8 Infrarotabstandssensoren, die rund um den e-puck herum angeordnet sind und auch zur Messung der Helligkeit eingesetzt werden können.
- 3 Infrarotsensoren auf der Unterseite des e-puck zur Linienverfolgung.
- 3 Microphone, die die Navigation nach Geräuschen ermöglichen.
- eine Kamera mit einer Auflösung von 640 x 480 Pixeln.
- ein 3D-Beschleunigungs-/Lagensensor.

³École Polytechnique Fédérale de Lausanne

- und ein Infrarotempfänger mit dem z.B. Signale einer Fernbedienung empfangen werden können.

Verfügbar sind in ZUSMORO zunächst nur die unteren und die ringsherum angeordneten Infrarotsensoren.

2.2.2 Aktoren

Die vorhandenen Aktoren sind:

- zwei Schrittmotoren, die jeweils eines der Räder antreiben. Die Richtung und Geschwindigkeit dieser Motoren kann in Werten von -1000 (rückwärts) bis +1000 (vorwärts) angegeben werden, wobei 1000 in etwa 13 cm/s entspricht
- 9 bzw. 8 ringsherum angeordnete rote LEDs⁴
- eine nach vorn gerichtete, leuchtstarke LED
- ein kleiner Lautsprecher zur Ausgabe von kurzen Audiosequenzen
- sowie kleine LEDs, die das Gehäuse des e-puck beleuchten können.

In ZUSMORO lassen sich davon bisweilen nur die nach vorn gerichtete LED und die sog. Body-LEDs zur Beleuchtung des Gehäuses **nicht** ansprechen.

3 Benutzung von ZUSMORO

Nachdem die grundlegenden Begriffe und Elemente des e-puck erläutert wurden, wird in den folgenden Abschnitten der Umgang und die Benutzung mit ZUSMORO erklärt.

Die Oberfläche (siehe Abbildung 2) von ZUSMORO gliedert sich grob in drei Bereiche:

- Die Werkzeugleiste am linken oberen Fensterrand,
- die Programmsteuerung links unten
- und die weiße “Editorfläche” daneben, auf der das Programm erstellt wird.

3.1 Die Werkzeuge

Die Werkzeugleiste beinhaltet die “Werkzeuge”, die zum Erstellen eines Programm benötigt werden und von denen jeweils nur eines zur Zeit aktiv ist. Diese sind:

- Das “move”-Werkzeug mit dem Zustände und Transitionen verschoben werden können⁵,
- das “create state”-Werkzeug mit dem Zustände erzeugt werden können,
- das “Transitions”-Werkzeug zum Erstellen von Transitionen,
- und das “delete”-Werkzeug zum Löschen von Zuständen und Transitionen.

⁴die beiden hintersten LEDs lassen sich nur gemeinsam ansteuern

⁵dies ist zum Teil auch möglich, wenn andere Werkzeuge ausgewählt sind

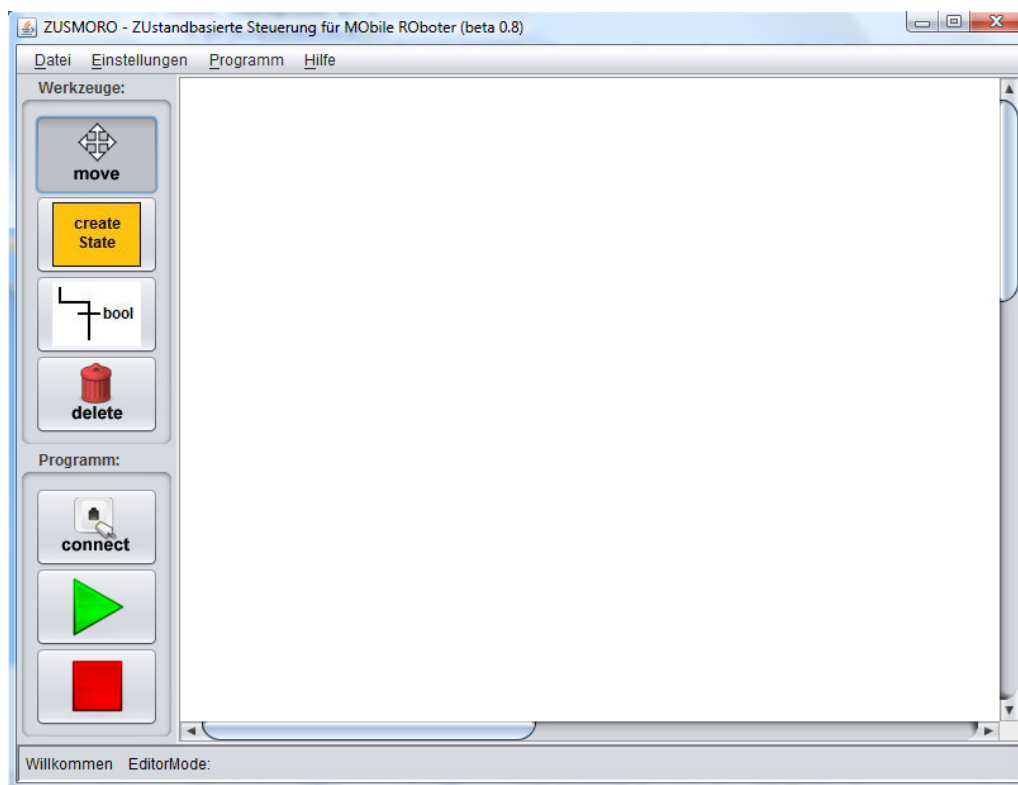


Abbildung 2: Das Hauptfenster von ZUSMORO

3.1.1 Erstellen von Zuständen

Um einen Zustand zu erstellen wählen Sie das “create state”-Werkzeug durch einfachen Klick aus und klicken anschließend auf die Editorfläche. Man kann mehrere Zustände nacheinander erstellen, indem man nach erstellen eines Zustandes die Maus wieder auf eine freie Fläche bewegt und erneut klickt. Der erste erstellte Zustand ist automatisch der Startzustand, zu erkennen an der doppelten Umrandung. Sie können den Startzustand selbstverständlich noch ändern. Nach Erstellen eines Zustands besitzt dieser noch keine Aktionen, das Programm behandelt dies allerdings als ob der Zustand die Aktionen “beide Motoren stop” “alle LEDs aus” sowie “Beeper aus” besäße.

3.1.2 Erstellen von Transitionen

Um eine Transition zu erstellen müssen zumindest zwei Zustände vorhanden sein, die miteinander verbunden werden, da Schleifen⁶ bei den hier verwendeten Moore-Automaten praktisch keine Funktion haben, werden diese von ZUSMORO nicht erstellt. Sollte dies jedoch dennoch gewünscht sein, kann dieses Verhalten durch eine Option in den Einstellungen deaktiviert werden. Wählen Sie also das “Transitions”-Werkzeug aus, bewegen Sie den Mauszeiger über den Ausgangszustand, drücken und halten Sie die linke Maustaste, bewegen Sie dann den Mauszeiger über den gewünschten Zielzustand und lassen Sie die Taste über diesem los. Die Transition wird

⁶Transitionen die vom Ausgangszustand wieder zu diesem zurückführen

nun durch eine schwarze Linie zwischen den beiden Zuständen angezeigt. Transitionen gehen immer von der Unterseite eines Zustandes aus und enden an der Oberseite des Zielzustands. Die Standardbedingung für neu erstellte Transitionen ist “true”, d.h. die Bedingung ist stets erfüllt.

3.1.3 Löschen von Zuständen/Transitionen

Zum Löschen von Zuständen oder Transitionen wählen Sie das “delete”-Werkzeug aus und klicken anschließend auf den Zustand oder die schwarze Linie einer Transition. **Beachten Sie, dass beim Löschen von Zuständen auch alle mit diesem Zustand verbundenen Transitionen gelöscht werden!** Neben der Möglichkeit Zustände und Transitionen mit dem “delete”-Werkzeug zu löschen, können Sie auch einfach die Maus über diese bewegen und dann die ENTF-Taste drücken.

3.1.4 Bewegen von Zuständen/Transitionen

Um Zustände oder Transitionen zu bewegen, wählen Sie das “move”-Werkzeug aus, klicken Sie dann auf das zu verschiebende Objekt, halten Sie die Taste gedrückt, bewegen Sie die Maus zu dem gewünschten Ort und lassen Sie dann die Taste los. Wenn Sie mehrere Zustände auf einmal verschieben wollen, halten Sie die STRG-Taste gedrückt und klicken Sie nacheinander die gewünschten Zustände an. Anschließend werden, wenn Sie einen der markierten Zustände verschieben, alle anderen Zustände in gleicher Weise verschoben. Alternativ können Sie auch einen Auswahlrahmen um die Zustände ziehen. Klicken und halten Sie dafür die linke Maustaste auf einer freien Fläche und ziehen Sie den Rahmen soweit bis alle gewünschten Zustände markiert erscheinen.

3.2 Weitergehende Bedienung

Nachdem Sie nun wissen, wie Sie Zustände und Transitionen erstellen, verschieben und löschen können, wird es Zeit diesen sinnvolle Aktionen und Bedingungen zuzuweisen.

3.2.1 Zuständen Aktionen zuweisen

Um einem Zustand Aktionen zuzuweisen wählen Sie eines der Werkzeuge mit Ausnahme des “delete”-Werkzeugs aus und Doppelklicken Sie auf einen Zustand. Es öffnet sich ein Fenster, dass es Ihnen erlaubt einzustellen, auf welche Werte die Aktoren, während dieser Zustand später während der Programmausführung aktiv ist, eingestellt werden sollen. Die Motoren können grob über die beiden Schieberegler und fein über die Pfeil-hoch- bzw. Pfeil-runter-Schaltflächen oder Tasten eingestellt werden. Die LEDs lassen sich durch klicken an- oder ausschalten, ebenso lässt sich der Beeper aktivieren oder deaktivieren. Zudem kann im oberen Bereich des Fensters die Bezeichnung eingegeben und ob dieser Zustand der Startzustand sein soll, eingestellt werden. Ein Klick auf “Übernehmen” setzt die Aktionen und schließt das Fenster, bei Klick auf Abbrechen werden alle Änderungen verworfen und das Fenster ebenfalls geschlossen.

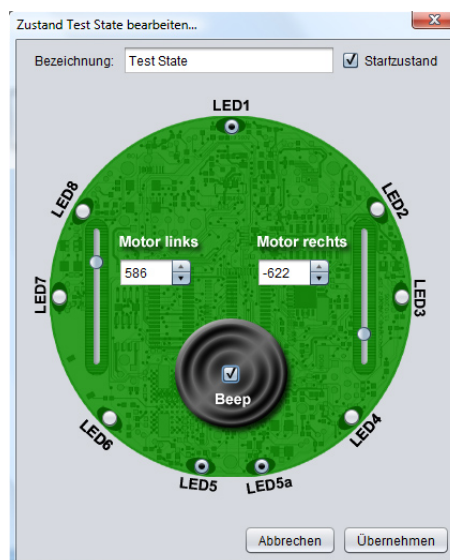


Abbildung 3: Fenster zum Zuweisen von Aktionen

3.2.2 Transitionen eine Bedingung zuweisen

Analog öffnet sich auch nach Doppelklick auf eine Transition ein Fenster. Auch hier kann im oberen Bereich eine Bezeichnung vergeben werden. Die Sensoren sind durch die schwarzen Rechtecke angedeutet, durch Klick auf einen dieser Sensoren erscheinen zwei Eingabelemente über dem Sensor; ein “Drop down”-Menü und ein Textfeld. Im “Drop down”-Menü kann der Operator und durch das Textfeld ein Vergleichswert eingestellt werden. Soll ein Sensor nachträglich doch nicht Bestandteil einer Bedingung sein, entfernen sie den Vergleichswert wieder. Um die Eingabe zu beenden bestätigen Sie sie mit der Return-Taste oder klicken Sie einen anderen Sensor

an. Die Eingabeelemente verschwinden und der Operator wird mit dem Vergleichswert unterhalb bzw. neben dem Sensor angezeigt. Verfahren Sie so weiter mit allen gewünschten Sensoren. Neben den Infrarotsensoren steht auch ein Timer zur Verfügung, dieser wird auf die gleiche Weise eingestellt.

ACHTUNG!:

Die Genauigkeit des Timers ist direkt von der eingestellten EVA-Frequenz abhängig. Der Zustandsübergang kann dementsprechend bei einer Frequenz von 20 Hz bis zu 50 ms später als eingestellt ausgelöst werden. Aus dem gleichen Grund sollte er nicht mit dem “==”-Operator verwendet werden.

Alle Vergleiche von Sensoren mit Vergleichswerten werden um einen aussagenlogischen Ausdruck zu formen mit dem logischen UND verknüpft. Um mehrere Bedingungen mit dem logischen ODER zu verknüpfen, erstellen Sie mehrere Transitionen, die die gleichen Zustände verbinden.

Während der Programmausführung werden die Transitionen in der Reihenfolge wie sie erstellt wurden daraufhin überprüft, ob ihre Bedingung erfüllt ist. In den Programmeinstellungen können Sie zudem die Option “Transitionenreihenfolge einstellbar” aktivieren, anschließend können Sie die Reihenfolge im Transitionsfenster manuell einstellen. Dort finden Sie ein Eingabefeld mit der Bezeichnung “Auswertungsrang”, eine Null in diesem Feld bedeutet, dass diese Transition zuerst evaluiert wird. Die erste Transition deren Bedingung während der Auswertung erfüllt ist, löst dann einen Zustandsübergang aus. Das bedeutet, dass die anderen Transitionen nicht mehr überprüft werden und somit auch keinen Zustandsübergang mehr auslösen können.

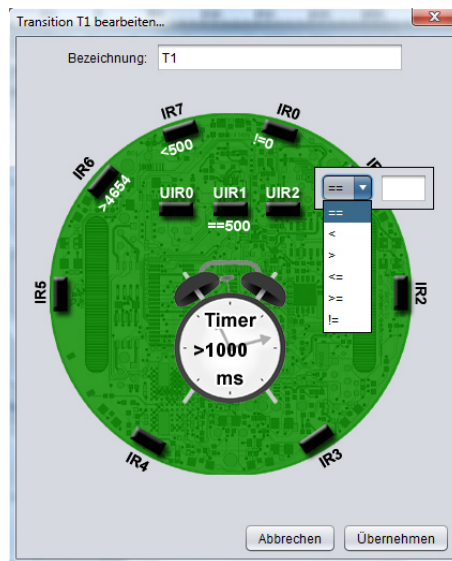


Abbildung 4: Fenster zum Zuweisen von Bedingungen

3.3 Programmsteuerung

Unter den oben genannten Werkzeugen befinden sich Schaltflächen mit denen die Programmausführung gesteuert werden kann⁷:

⁷ dieselben Funktionen sind auch über die Menüleiste in dem “Programm”-Menü erreichbar.

- durch klicken auf den “connect”-Button kann das Programm mit einem Roboter verbunden werden.
- das grüne Dreieck ist der Startbutton, durch den die Programmausführung auf dem Roboter gestartet wird.
- ein Klick auf das rote Quadrat (Stop) stoppt die Programmausführung und hält den Roboter an.

3.3.1 Verbinden zu einem Roboter

Soll das erstellte Programm auf dem Roboter ausgeführt werden, muss zuvor eine Verbindung zu diesem aufgebaut werden. Dazu verwenden Sie den “connect”-Button, Sie werden dann dazu aufgefordert einen COM-Port anzugeben. Zuvor muss also eine virtuelle COM-Schnittstelle mit Hilfe ihrer Bluetooth-Software eingerichtet werden. Merken Sie sich die Nummer des virtuellen Ports und geben Sie diese dann in der Form “COMx”, wobei x für die Nummer steht, ein⁸. Je nach Rechner und Bluetooth-Software kann der Verbindungsaufbau unterschiedlich lange dauern, die Verbindung wurde erfolgreich aufgebaut sobald die orange Kontroll-LED am e-puck leuchtet.

3.3.2 Verbindung trennen

Um eine hergestellte Verbindung wieder abzubauen, klicken Sie auf denselben Button, den Sie auch zum Herstellen genutzt haben. Dieser ist nachdem eine Verbindung hergestellt wurde mit “disconnect” beschriftet. Oder wählen sie “Verbindung trennen” aus dem Programmmenü. Wollen Sie die Arbeit mit einem anderen Roboter fortsetzen, können Sie aus diesem Menü auch direkt erneut “Verbinden. . .” auswählen, die zuvor hergestellte Verbindung wird dann automatisch getrennt, bevor eine neue aufgebaut wird.

3.3.3 Programmausführung beginnen

Nachdem die Verbindung aufgebaut wurde, kann das Programm durch klicken auf den Startbutton gestartet werden. Es erscheint zeitgleich ein Fenster in dem kontinuierlich die Werte der Sensoren ausgegeben werden. Zudem wird der aktive Zustand während der Programmausführung durch einen farbigen Rahmen markiert. Während der Ausführung eines Programms kann dieses nicht verändert werden!

3.3.4 Programmausführung beenden

Um die Programmausführung zu beenden, klicken Sie den Stopbutton. Der Roboter hält an und das Programm kann wieder editiert werden. Die Programmausführung kann jederzeit wieder durch klicken auf den Startbutton begonnen werden. Die zuvor aufgebaute Verbindung bleibt bestehen und muss nicht erneut aufgebaut werden.

⁸Die Bezeichnung des virtuellen COM-Ports kann unter Linux und anderen Betriebssystemen anders lauten

4 Schnellstart und Beispielprogramm

Im Folgenden wird Schritt für Schritt die Erstellung und Ausführung eines kleinen Programms mit Hilfe von ZUSMORO erläutert, das die meisten Elemente des Programms verwendet. Es simuliert ein einfaches Fluchtverhalten, der Roboter fährt gerade aus und sobald die vorderen Sensoren ein Hindernis detektieren, weicht er zurück, dreht sich in eine andere Richtung und fährt dann wieder vorwärts usw. . .

4.1 Zustände erstellen

Das Programm besteht aus drei Zuständen:

- Wählen Sie das “create state”-Werkzeug aus und erstellen Sie drei Zustände durch klicken auf freie Stellen der weißen Editorfläche.
- Doppelklicken Sie auf den Zustand mit der Bezeichnung “State 1”, dieser sollte doppelt umrandet sein (er ist der Startzustand).
- Ändern Sie in dem sich öffnenden Fenster die Bezeichnung in “vorwärts”, schieben Sie die Schieberegler der beiden Motoren nach oben auf einen Wert von 514 und aktivieren sie LED1; klicken sie anschließend auf “Übernehmen & Schließen”.
- Ändern Sie für “State 2” die Bezeichnung auf “rückwärts”, die Regler der Motoren stellen Sie beide auf -1000, aktivieren Sie LED5 (LED5a wird automatisch mit aktiviert, da diese nur gemeinsam angesteuert werden können) und bestätigen Sie wieder mit Klick auf “Übernehmen & Schließen”.
- Nun zu “State 3”: Bezeichnung: “drehen”, Motoren: -514 und +514, LED3 und LED7 aktivieren; anschließend wieder mit “Übernehmen & Schließen” bestätigen.

4.2 Transitionen erstellen

Nun werden die Zustände durch Transitionen miteinander verbunden:

- Wählen das “Transitions”-Werkzeug aus, bewegen Sie die Maus über den Zustand “vorwärts” halten Sie die linke Maustaste gedrückt, bewegen Sie die Maus über den Zustand “rückwärts” und lassen Sie die Taste dann los.
- Verfahren Sie in gleicher Weise von “rückwärts” zu “drehen” und von “drehen” wieder zu “vorwärts”.
- Verschieben Sie durch Benutzung des “move”-Werkzeugs die erstellten Zustände und Transitionen um die Übersichtlichkeit zu verbessern.

Ihr Programm sollte nun wie in Abbildung 5 aussehen.

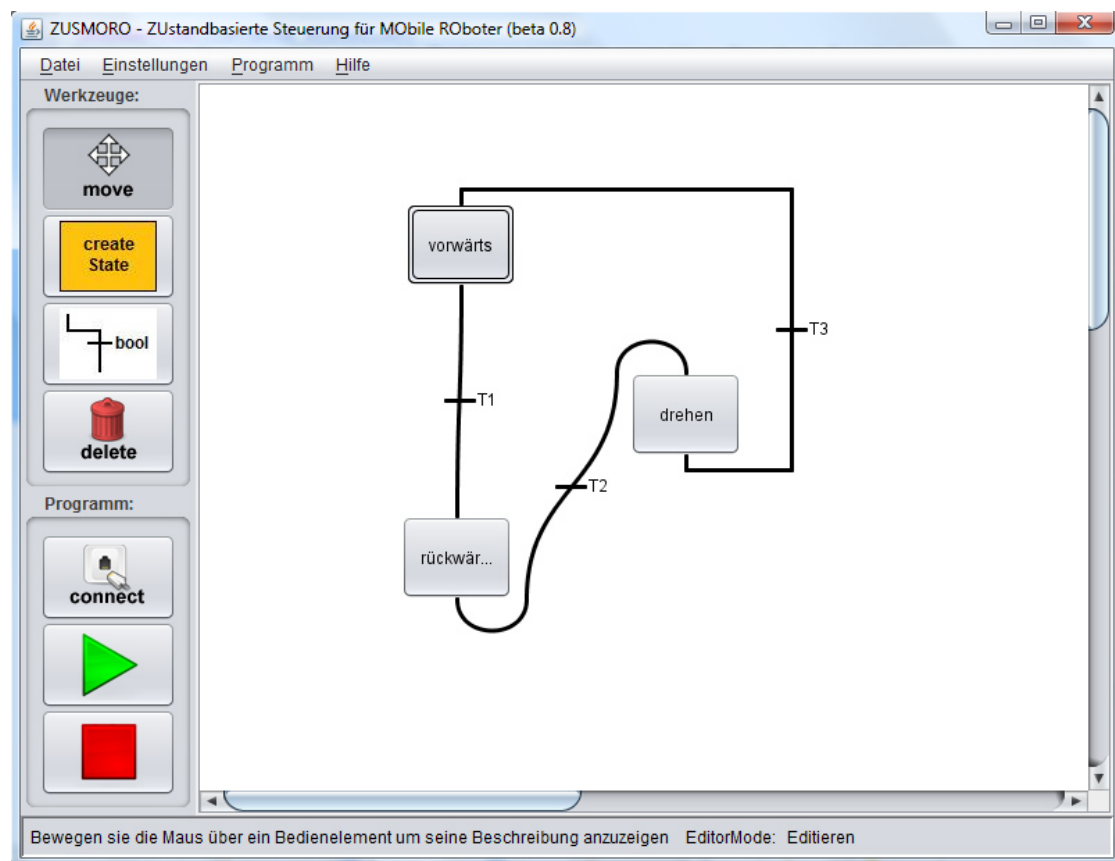


Abbildung 5: Zwischenstand des Beispielprogramms

4.3 Bedingungen einstellen

Abschließend müssen noch die Bedingungen für die Transitionen eingestellt werden:

- Doppelklicken Sie auf die Transition zwischen den Zuständen “vorwärts” und “rückwärts” diese sollte die Bezeichnung T1 tragen.
- Stellen Sie für IR7 und IR0 jeweils “>500”⁹ ein und übernehmen sie die Änderungen.
- Für T2 wählen Sie lediglich den Timer aus und stellen für ihn “>800” ein, dies bewirkt dass der Zustand “rückwärts” nach frühestens 800 ms verlassen wird.
- T3 erhält ebenfalls als einzige Bedingung Timer “>500”, bestätigen Sie jeweils mit “Übernehmen”.

4.4 Programm sichern

Das Programm ist nun soweit fertig, zur Sicherheit sollten Sie es jetzt abspeichern:

- Wählen Sie dazu den Eintrag “Speichern unter...” aus dem Dateimenü aus.

⁹die Infrarotsensoren liefern umso größere Werte, je näher sich ein Hinderniss vor ihnen befindet.

- Wählen Sie Speicherort und Dateiname aus und klicken Sie dann auf “Speichern”.

4.5 Verbindung herstellen

Damit das Programm ausgeführt werden kann müssen Sie eine Verbindung zu einem e-puck herstellen.

- Klicken Sie auf die “connect”-Schaltfläche im linken unten Bereich des Hauptfensters.
- Geben Sie den Namen des virtuellen COM-Ports ein, den Sie zuvor mit Ihrer Bluetooth-Software für den e-puck eingerichtet haben.
- Bestätigen Sie mit “Ok” und warten Sie bis die orangene Kontroll-LED am e-puck leuchtet¹⁰.
- Sobald diese aufleuchtet ist die Verbindung aufgebaut.

4.6 Programm ausführen

Klicken Sie auf den grünen Startbutton um die Programmausführung zu starten. Sie können nun jeweils auf der Editorfläche verfolgen welcher Zustand aktiv ist. Zudem öffnet sich ein Fenster, dass kontinuierlich die Sensordaten darstellt.

4.7 Programmausführung abbrechen

Zum Anhalten des Roboters und Beenden der Programmausführung klicken sie den roten Stop-button.

Herzlichen Glückwunsch!
Sie haben ihr erstes Programm erstellt, gespeichert und ausgeführt.

¹⁰diese befindet sich auf der gegenüberliegenden Seite der grünen Stromversorgungs-LED