

Autonome Systeme  
Hochschule für Technik und Wirtschaft  
SS2011

---

# BoidBot $A_2R$

— Dokumentation —

---

29. Juli 2011

**Andreas Bilke, B. Sc.**  
(andreas.bilke@student.htw-berlin.de)

**Andreas Günther, B. Sc.**  
(andreas.guenther@student.htw-berlin.de)

# Inhaltsverzeichnis

<b>1</b>	<b>Aufgabenstellung</b>	<b>1</b>
<b>2</b>	<b>Hardware</b>	<b>2</b>
2.1	Antrieb . . . . .	2
2.2	Sensoren . . . . .	3
2.3	Hardware-Controller . . . . .	4
<b>3</b>	<b>Software-Controller</b>	<b>4</b>
3.1	Verwendete Softwarekomponenten . . . . .	4
3.2	Grundverhalten . . . . .	6
3.3	Verhaltensbasierter Roboter . . . . .	6
3.4	Algorithmus . . . . .	7
3.5	Herausforderungen . . . . .	7
<b>4</b>	<b>Zusammenfassung</b>	<b>8</b>

# 1 Aufgabenstellung

Ziel der Übung Autonome Systeme war es, einen Roboter zu entwickeln, der eine frei gewählte Aufgabe lösen kann. In dieser Dokumentation wird beschrieben, wie zwei autonome Systeme, basierend auf der Arduino Plattform, entwickelt wurden.

Beide Roboter sollen sich nach dem Prinzip der Boids<sup>1</sup> verhalten. Jeder Boid (Roboter) besitzt drei Grundregeln. Mit diesen Regeln ist es möglich komplexe Systeme wie einen Schwarm zu simulieren. Die Regeln sind

- Separation
- Angleichung
- Zusammenhalt

Durch diese Regeln wird erreicht, dass in diesem Fall zwei Roboter gemeinsam durch einen Raum fahren können. Dabei können sie Hindernissen ausweichen und bleiben dennoch im Kollektiv. In Abbildung 1<sup>2</sup> sind die für das System notwendigen Regeln graphisch abgebildet.

Es wird die technische Lösung beschrieben wie beide Roboter Hindernissen ausweichen und sich gegenseitig erkennen können.

---

<sup>1</sup><http://www.red3d.com/cwr/boids/>

<sup>2</sup>Bilder entnommen von <http://www.red3d.com/cwr/boids/>

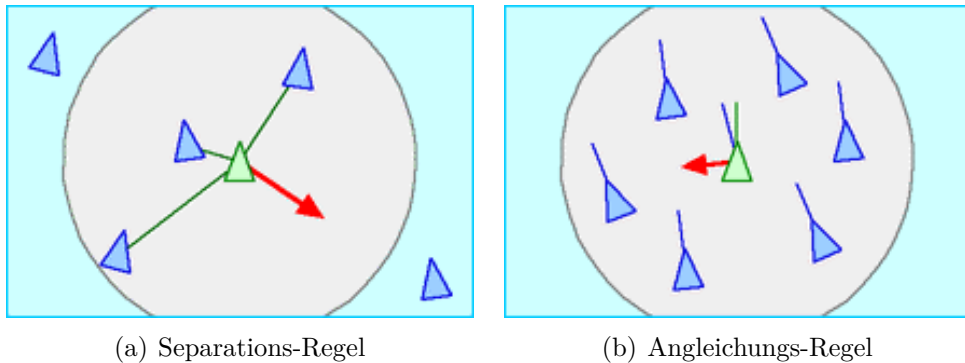


Abbildung 1: Relevante Regeln der Boids

## 2 Hardware

In diesem Abschnitt wird der technische Aufbau des BoidBot  $A_2R$  (siehe Abbildung 2) beschrieben. Grundgerüst des BoidBot  $A_2R$  bildet eine  $100\text{ cm}^2$  große Acrylglasplatte, an der alle weiteren Komponenten angebracht sind.

Als Controller dient ein *Arduino Uno*<sup>3</sup>, dass mit einem selbstentworfenem Controller-Shield zur Steuerung des Antriebs und der Sensorik ausgestattet wurde. Das gesamte System wird durch eine Batterie mit einer Spannung von zirka 9,6 V versorgt. Insgesamt sind innerhalb dieses Projekts zwei Roboter entstanden, BoidBot  $A_2R$  I und BoidBot  $A_2R$  II, die in Abbildung 2 abgebildet sind.

### 2.1 Antrieb

Auf der Unterseite der Grundplatte sind zwei Servomotoren angebracht, die den BoidBot  $A_2R$  antreiben sowie ein Stützrad. Die hier verbauten Servomotoren haben in ihrer ursprünglichen Bauform nur einen Einstellbereich von  $270^\circ$  und somit als Antriebsmotoren nicht nutzbar. Daher wurden diese Motoren entsprechend modifiziert<sup>4</sup>, damit sie als Antriebsmotoren einsetzbar sind.

Zum Antrieb des Sensor-Auslegers, der sich um den Roboter bewegt, ist auf der Grundplatte ein bipolarer Schrittmotor angebracht. Dieser bewegt

<sup>3</sup>[www.arduino.cc](http://www.arduino.cc)

<sup>4</sup><http://www.fpoppa.de/brecker/robotik/tutorials/servo-mod>

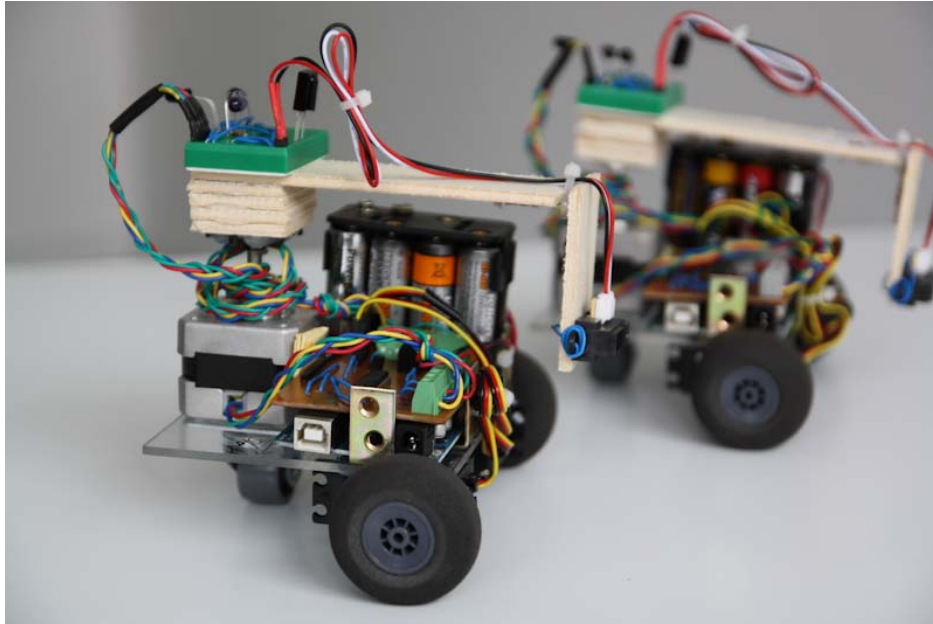


Abbildung 2: BoidBot  $A_2R$

den Ausleger Kreis um den BoidBot  $A_2R$  herum. So ist gewährleistet, dass möglichst wenig Sensoren benötigt werden, um den Bereich um den Roboter sensorisch abzudecken.

## 2.2 Sensoren

Am bereits erwähnten Sensorturm sind verschiedene Sensoren zur Abstandsmessung sowie zum senden und empfangen von Infrarotlicht angebracht. Die Abstandsmessung wird mit einem Infrarot-Proximity-Sensor vorgenommen, der am Ende des Sensor-Auslegers angebracht ist. Zum Senden von Infrarotlicht, sind drei Infrarot-LEDs über der Schrittmotorachse angebracht. Sie dienen dazu, den Roboter einem anderen BoidBot  $A_2R$  gegenüber identifizierbar zu machen. Um selber einen BoidBot  $A_2R$  zu erkennen, besitzt der Roboter einen Infrarotempfänger, der ebenfalls über der Schrittmotorachse in Richtung des Auslegerarms angebracht ist.

## 2.3 Hardware-Controller

Wie bereits erwähnt ist ein *Arduino Uno* Basis der Hardwaresteuerung. Zusätzlich ist daran ein selbstentworfenes Controller-Shield angeschlossen, welches eine Schnittstelle zu den restlichen Komponenten des Systems bildet. Daran sind Stromversorgung, Schrittmotor, Sensoren und die Antriebs-Servomotoren angeschlossen. Für die Ansteuerung des Schrittmotors befindet sich auf dem Controllerboard ein *L293D H-Driver*<sup>5</sup>. Die Servomotoren werden durch das Arduino gesteuert, allerdings mussten wir feststellen, dass durch den Umbau es keinen Ruhepunkt gibt, an denen die Motoren still stehen. Dies ist aber notwendig um den Roboter anzuhalten. Durch einen vorgeschalteten Transistor, kann für das Anhalten die Versorgungsspannung für die Servomotoren komplett abgeschaltet werden. In Abbildung 3 ist die Verschaltung der einzelnen Komponenten dargestellt.

## 3 Software-Controller

In diesem Abschnitt wird die entwickelte Software für die Roboter beschrieben. Die Grundplattform ist Arduino, daher wurde zur Erstellung der Software die Arduino Entwicklungsumgebung genutzt. Als Programmiersprache wird ein C-Dialekt verwendet.

### 3.1 Verwendete Softwarekomponenten

Die Arduino Plattform besitzt eine große Gemeinschaft, welche für viele Bauteile spezielle Bibliotheken zur Verfügung stellt. Diese erleichtern das Entwickeln von eigenen Projekten.

Es wurden Bibliotheken für die Ansteuerung von Servos<sup>6</sup>, einem Schrittmotor<sup>7</sup> und zur Steuerung der Infrarot LEDs<sup>8</sup> genutzt.

---

<sup>5</sup><http://www.datasheetcatalog.org/datasheet/texasinstruments/l293.pdf>

<sup>6</sup><http://www.arduino.cc/playground/ComponentLib/Servo>

<sup>7</sup><http://arduino.cc/en/Reference/Stepper>

<sup>8</sup><https://github.com/shirriff/Arduino-IRremote>

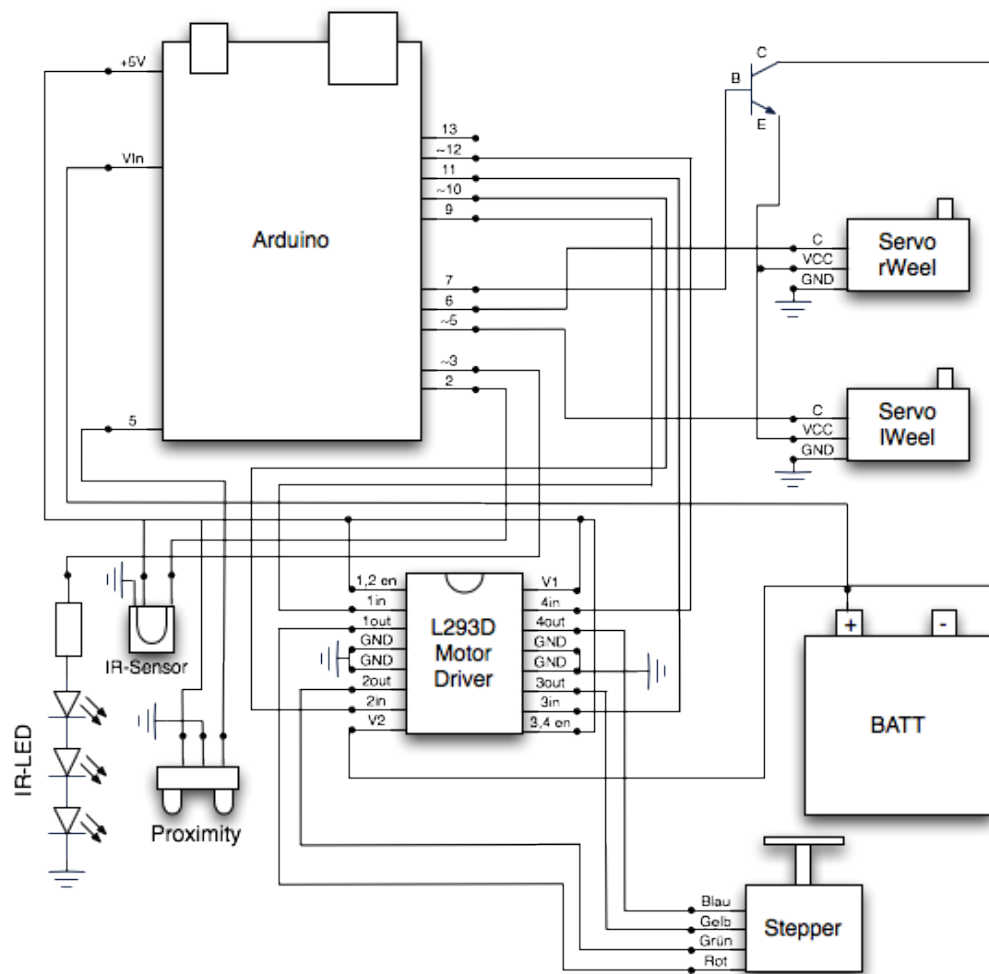


Abbildung 3: Schaltbild des Hardware-Controllers

## 3.2 Grundverhalten

Um den Sensorkopf, an dem der Distanzsensor und der Infrarotempfänger installiert sind, zu steuern wurde ein Schrittmotor genutzt. Dieser kann eine vollständige Umdrehung in 200 Schritten absolvieren. Das bedeutet, dass ein Schritt dieses Motors den Schwenkkopf um  $1,8^\circ$  dreht.

Im Grundverhalten wird dieser Kopf im Halbkreis um den Roboter bewegt, damit er den kompletten Bereich vor ihm abtasten kann. In jedem Schritt wird der aktuell ermittelte Wert des Distanzmessers gelesen und verarbeitet.

Der Distanzsensor hat eine Reichweite von 10 bis 80 cm. Versorgt wird der Sensor über einen 5 V Anschluss. Je nach ermittelter Entfernung kann am Sensorausgang eine entsprechende Spannung abgelesen werden. Das Ablesen erfolgt mittels analogem Eingang am Arduino, welches die Eingangsspannung auf einen Wert zwischen 0 und 1023 abbildet.

Damit der jeweilige Roboter vom anderen erkannt werden kann, wird ein Infrarot Signal permanent ausgesendet. Dazu wird die IRremote Bibliothek genutzt, die es erlaubt beliebige Daten über Infrarot zu verschicken. Erkannt werden diese Lichtsignale über einen Infrarotempfänger. Dieser besitzt neben der Stromversorgung auch einen Signalausgang. Wird kein IR Signal empfangen, liegt an diesem eine Spannung von 5 V an. Beim Erkennen eines IR Signals, fällt die Spannung kurzfristig auf 0 V. Um diese Veränderung zu erkennen, wurde der Empfänger an einen Interrupt-fähigen Pin des Arduinos angeschlossen. Kommt es zum Spannungsabfall an diesem Pin wird eine Funktion der Mikrocontrollers aufgerufen. Dieser speichert zur weiteren Verarbeitung den aktuellen Winkel des Schwenkkopfes.

## 3.3 Verhaltensbasierter Roboter

Der Algorithmus basiert auf der Verwendung von verschiedenen Verhalten. Diese sind auf dem Roboter mittels einer Statusmaschine implementiert. Das Standardverhalten eines jeden Roboters ist es, gerade aus zu fahren. Falls ein Hindernis mittels des Schwenkarms entdeckt wurde, wird in den Status Objekt ausweichen gewechselt. Konnte mittels Infrarot Empfänger ein anderer Roboter erkannt werden, wird das Verhalten Roboter folgen genutzt.



### 3.4 Algorithmus

Programmiert man ein Arduino müssen mindestens zwei Funktionen implementiert werden. Zum einen die setup Methode um sämtliche Sensoren zu initialisieren, zum anderen eine loop Funktion. Diese wird vom Mikrocontroller permanent aufgerufen. In dieser Funktion muss daher die Logik programmiert werden. Der Grundsätzliche Ablauf dieser Loop-Funktion ist in Algorithmus 1 skizziert.

Der Grundalgorithmus ist, dass die Sensordaten ausgelesen werden und basierend darauf ein neuer Status gewählt wird. Könnte zum Beispiel ein Hindernis auf der linken Seite des Roboters erkannt werden, soll der Roboter nach rechts ausweichen. In den Stati Ausweichen und Folgen wird der Sensorkopf nicht bewegt. Der Grund ist, das so lange das Hindernis beobachtet werden soll bis der Roboter außerhalb des Gefahrenbereichs ist.

---

**Algorithm 1** Programmablauf eines Roboters

---

```
sendIrSignal
if  $s = FORWARD$  then
    moveSensorTurret
    readSensorData
    changeStateIfNecessary
    reactOnStateAndSensorData
else if  $s = AVOID \vee s = FOLLOW$  then
    readSensorData
    changeStateIfNecessary
    reactOnStateAndSensorData
end if
```

---

### 3.5 Herausforderungen

Auf dem Sensorturm sind drei Infrarot LEDs installiert. Zusätzlich wird zum erkennen des anderen Roboters ein Infrarot Sensor benutzt, der sich zusammen mit dem Turm dreht.

In Tests musste festgestellt werden, dass die Erkennung des Infrarot Lichts nur auf relativ geringe Distanzen funktioniert. Ein Grund dafür kann zum einen eine zu schwache Infrarot LED sein, zum anderen hat sich die Ausrichtung der LED als weiteres Hindernis herausgestellt.

Wird der Sensorturm des anderen Roboters nicht bewegt, konnte über Distanzen von ca. 30 - 40 cm eine gute Erkennungsrate beim zweiten Roboter erzielt werden. Jedoch sollen beide Roboter durch einen Raum fahren können, dies erfordert das drehen des Sensorturms. In diesem Szenario war die Erkennungsrate sehr gering, eine akzeptable Leistung konnte nicht erzielt werden.

Zur Verbesserung der Erkennungsrate wird zum einen Vorgeschlagen stärkere Infrarot LEDs zu benutzen und zum anderen mehr von diesen auf dem Turm zu installieren. Damit wird erreicht, das in alle Richtungen das Signal ausgesandt und daher auch aus jeglichen Richtungen erkannt werden kann.

## **4 Zusammenfassung**

Diese Arbeit beschreibt, wie zwei auf Arduino basierende Roboter nach dem Prinzip der Boids Hindernissen ausweichen und sich gegenseitig erkennen. Es wurde dargestellt, wie die Roboter technisch aufgebaut sind und wie die Software diese steuert, damit die an sie gestellten Anforderungen erfüllt werden.

Es wurde gezeigt, dass die Roboter mit Hilfe von Infrarotabstandssensoren ihre Umgebung abtasten und Hindernissen ausweichen können, wenn diese zu nah sind. Auch wurde beschrieben, das sie sich gegenseitig per Infrarot identifizieren können, jedoch auf Grund von baulichen Eigenschaften dies nur in einem eingeschränkten Bereich bzw. nicht während der Fahrt möglich ist, aber dennoch prinzipiell funktioniert.