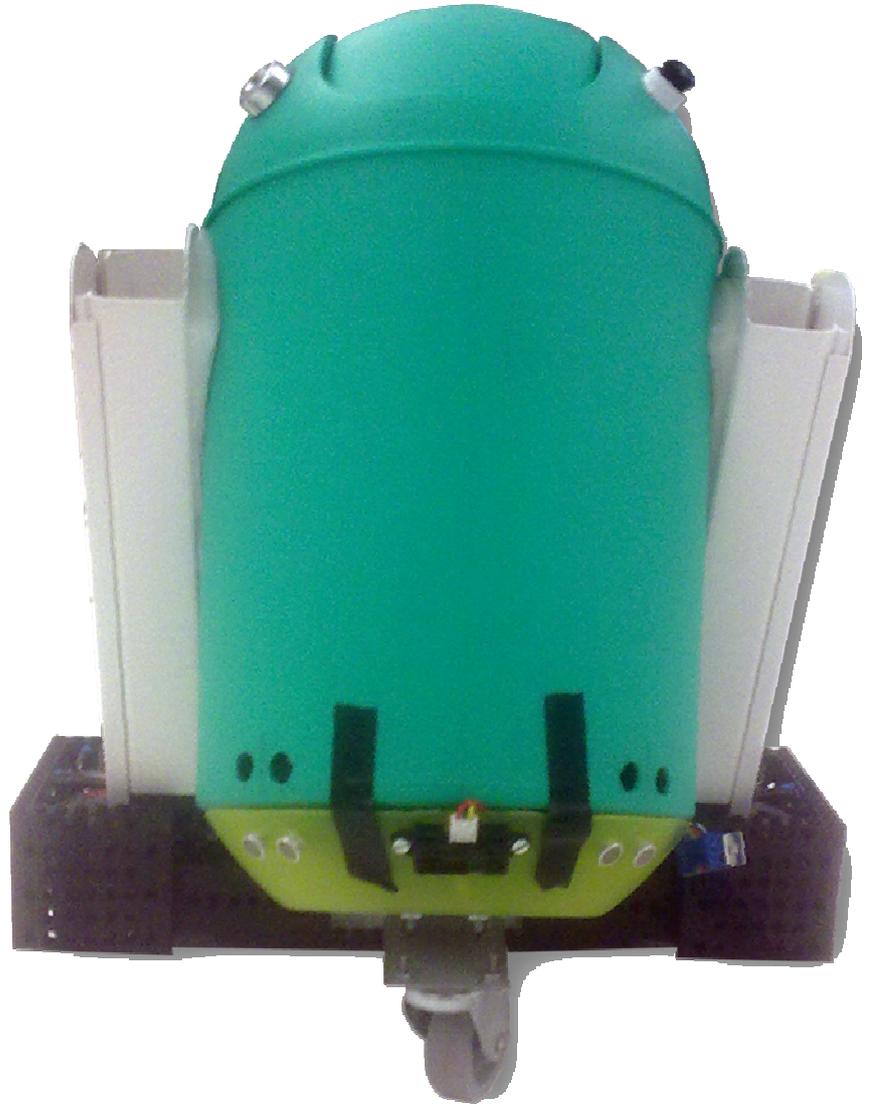


# 2010

FH Brandenburg

Peter Hirschfeld, Sebastian Schmidt, Thomas Trölmich



## [DOKUMENTATION ZU TBR1]

Dokumentation über die verwendete, selbst erstellte Hardware, sowie deren Anschlüsse auf dem Axen-Board. Desweiteren eine Anleitung zum Auseinandernehmen und in Betrieb nehmen des Roboters TBR1 (Trash Basket Robot).

## Inhalt

Erläuterung TBR1.....	3
In Betrieb nahme.....	5
AKSEN-Board flashen.....	6
CMU 3 CAM Flashen.....	7
Die Sensorik.....	8
Die Aktorik.....	10
Weitere Technik .....	11
Anleitung zum Öffnen des Roboters .....	14
Genutzte Anschlüsse auf dem AKSEN-Board .....	21
Schaltpläne der angefertigten Controller.....	22
Headlight - Controller.....	22
Grundlegende Beschreibung.....	22
Funktionsbeschreibung in TBR1 .....	23
Befehlsliste .....	24
Stückliste .....	24
Platinenlayout .....	25
Schaltplan .....	26
Clapper - Controller.....	27
Grundlegende Funktionsbeschreibung .....	27
Stückliste .....	27
Platinenlayout .....	27
Schaltplan .....	28
Nützliche Information .....	29

## Erläuterung TBR1

TBR1, was ausgesprochen für Trash Basket/Box Robot steht, gibt wieder, was das Grundelement, die Grundidee des Roboters ist. EIN MÜLLEIMER!

TBR1 wurde im Rahmen des Projektes Künstliche Intelligenz im fünften Semester an der FH Brandenburg entwickelt. Er ist ein Entertainmentroboter, der mit seinen Menschen im Umfeld über eine Kamera sowie einem Mikrofon interagieren kann und Aufmerksamkeit durch sein ähnliches Aussehen zu STARS WARS R2D2 erregen soll.

TBR1 besitzt ein RN-Speaker-Board, welches es ihm ermöglicht, sich akustisch mitzuteilen. Desweiteren verfügt er über 2 LED-Anzeigen an seinem Kopf, mit denen Gefühlszustände wie Langweile, gute sowie schlechte Laune vermittelt werden können. Mittels einer CMUCAM3 kann TBR1 Bilder aus seiner Umwelt verarbeiten und die ausgewerteten Daten über eine Bluetooth-Verbindung an das AKSEN-Board senden.

In den Radkästen befinden sich jeweils 3 Infrarot-Sensoren, mit Hilfe derer überprüft werden kann, ob TBR1 in irgendeiner Hinsicht blockiert ist. An der Front besitzt er 4 weitere Sensoren: einen akustischen Sensor in Form eines Mikrofons in der Mitte des Rumpfes, 2 Sonar-Sensoren, die ca. 45° versetzt zueinander befestigt sind und einen weiteren Infrarot-Sensor in der Mitte zur Erkennung von Hindernissen. 2 gecrackte Servomotoren<sup>1</sup> dienen TBR1 als Antriebsmotoren, ein weiterer gecrackter Servomotor als Kopfantrieb. Der Kopf des TBR1 lässt sich um 360° frei drehen, dank einer individuell angefertigten Schleifkontaktstange im inneren. Als Gimmick wurde ein weiterer Servomotor im Kopf verbaut, mit dem die Mund-Luke verbunden ist, um diese bewegen zu können.

<sup>1</sup>gecrackte Servomotoren – Servomotoren die sich 360° frei drehen können, aber über keine Positionierung ihrer selbst mehr wissen. Können wie normal Motoren angesteuert werden. Angaben zur Drehbewegung um  $x^\circ$  sind somit sinnlos.

## Inbetriebnahme



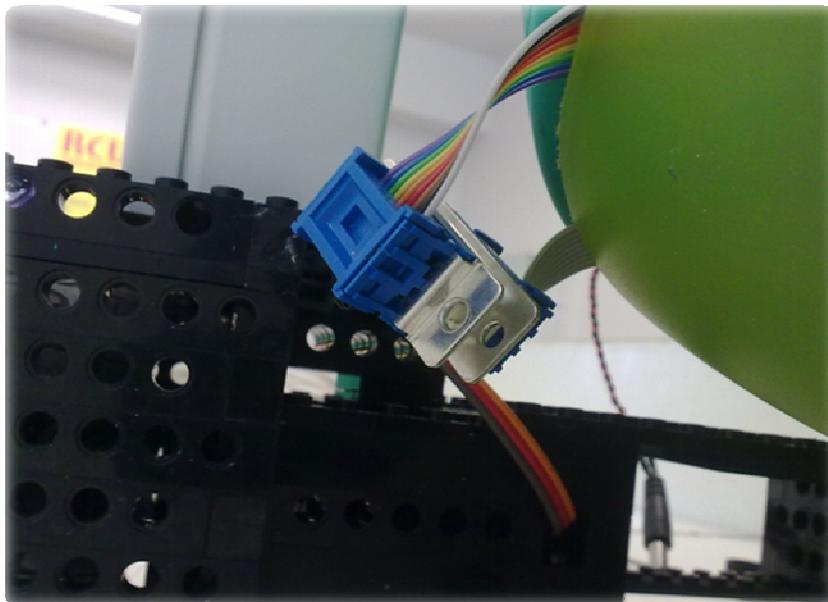
( **BILD 1:** Ansicht von hinten, Bedienelemente zur Inbetriebnahme sowie Umschaltfunktion zwischen dem Flash- (Rote LED) und Betriebsmodus (Grüne LED) des Aksen-Boards mit LED-Anzeige und inkl. des Flash-Tasters )

Um den Entertainmentrobot TBR1 ( Trash Basket Robot 1 ) in Betrieb zu nehmen, muss als erstes der Power-Schalter betätigt werden (links im Bild). Des Weiteren sollte sichergestellt werden, dass der Schalter auf der rechten Seite im Flashmodus steht, sprich die so dass die rote LED leuchtet. Nachdem diese beiden Schritte passiert sind, sollte einen Augenblick, etwa so 10-15 Sekunden, gewartet werden, damit die interne Bluetooth-Verbindung zur Kamera hergestellt werden kann. Letztendlich muss der Modusschalter auf die obere Position geschaltet werden. Sollte TBR1 jetzt nicht losfahren, ist der rote Reset-Taster zu betätigen.

## AKSEN-Board flashen

Zum flashen eines neuen Programms auf das Aksen Board muss der Roboter TBR1 eingeschaltet (linker Arm aus der Sicht der Roboters) und in den Flashmodus versetzt (rechter Arm a.d.S. des Roboters) werden (Siehe [Bild 1](#)).

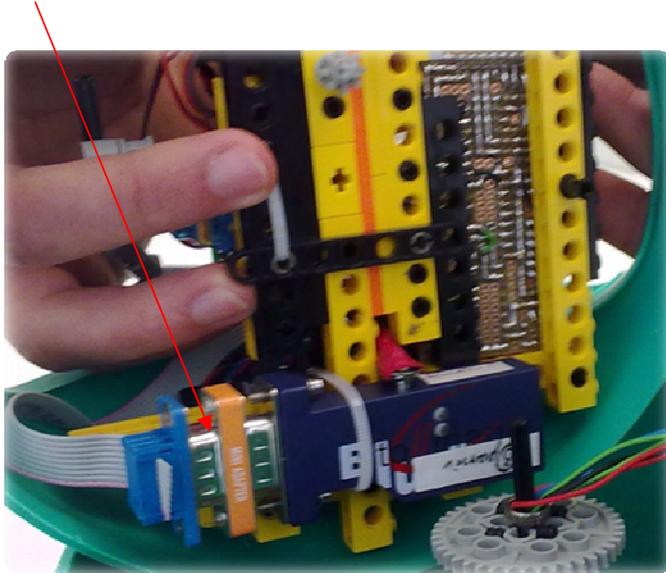
Nachdem dies geschehen ist, muss man die RS232-Verbindung, welche an der Vorderseite außen ersichtlich ist, trennen und an dem farbigen Ende den COM-Anschluss des Quellrechners anschließen. Als Übertragungsgeschwindigkeit gilt die Baudrate von 9600 Baud.



**Bild2:** Com2 Schnittstelle (Kamera- / Flashverbindung)

## CMUCAM3 Flashen

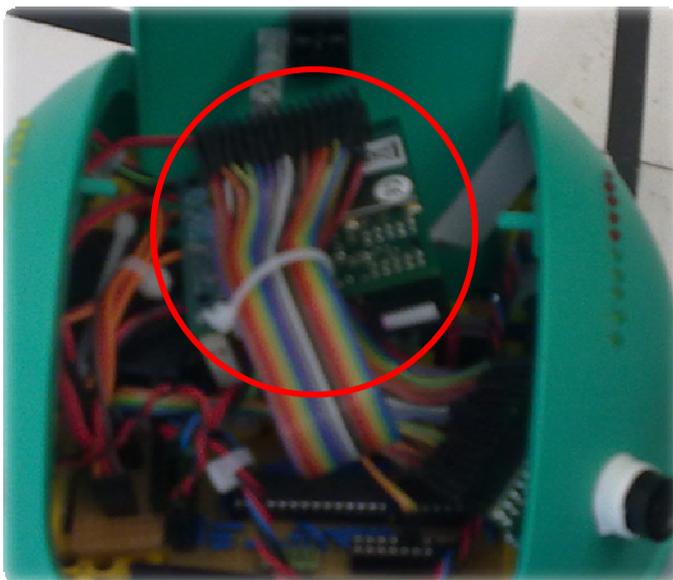
Um die CMUCAM3 flashen zu können, muss zunächst der Kopf vom Rumpf gelöst werden, siehe hierzu [Anleitung zum Öffnen des Roboters](#). Ist dies geschehen, kann die RS232 Verbindungen zwischen dem Bluetooth-Adapter, der am Kopf sitzt, und der Kamera getrennt werden. Der Adapter zwischen Kabel und Bluetooth-Adapter ist ebenfalls zu entfernen.



**Bild 3** Bluetooth-Adapter im Kopf

Dann kann man das Kabel einfach wieder mit dem COM-Anschluss des Quellrechners verbinden und die Kamera in den FLASH-MODUS versetzen.

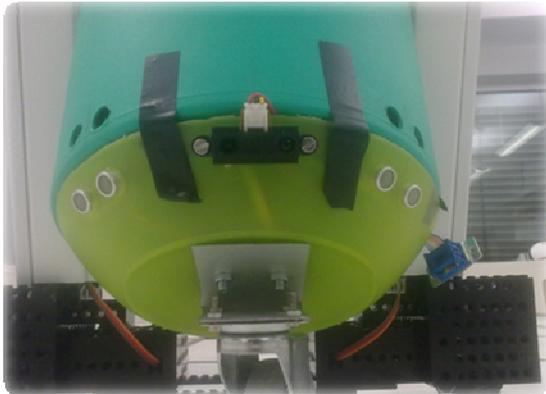
Den FLASH-MODUS der Kamera aktiviert man, indem man die Kamera ausschaltet und mit gedrücktem ISP-Knopf wieder einschaltet.



**Bild 4** Kameramodul mit Power- und ISP-Schalter

## Die Sensorik

TBR1 verfügt über IR-Sensoren an den Radkästen sowie an der Vorderseite zur Blockaden- und Hinderniserkennung. Außerdem befinden sich an der Vorderseite 2 Sonarsensoren, welche ca. in einem 45° Winkel angebracht sind.

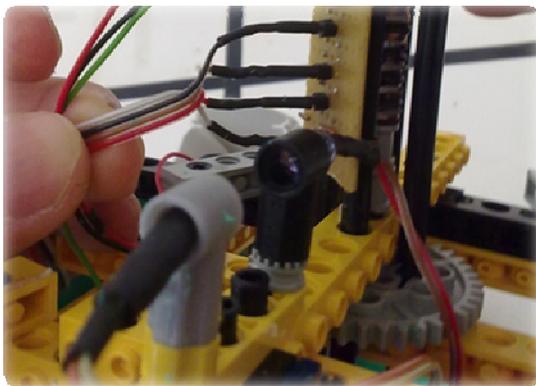


**Bild 5** Frontsensoren

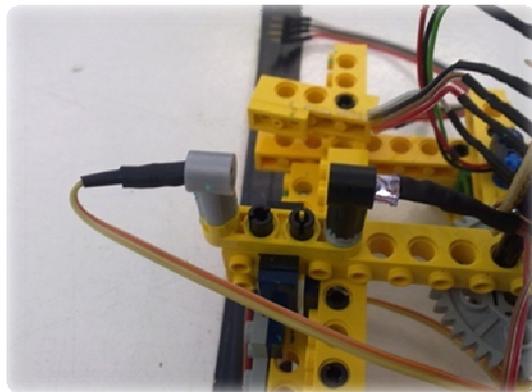


**Bild 6** Radkastensensoren

Im Inneren von TBR1 ist neben der Schleifkontaktstange noch ein weiterer IR-Sensor, welcher zur Positionierung des drehbaren Kopfes genutzt werden kann (Funktionsprinzip einer Lichtschranke).



**Bild 7** Nullstellungssensoren für den Kopf



**Bild 8** Nullstellungssensoren für den Kopf

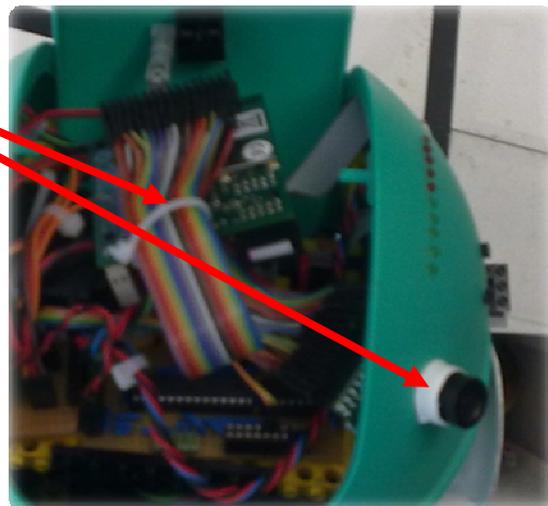
Ein weiterer Sensor befindet sich am Rumpf des Roboters an der Vorderseite. Der sogenannte „Klatsch-Sensor“ ist ein Mikrofon, welches bei einem bestimmten Lautstärkepegel ein Signal liefert. Im Bild unten ist nur die Ausfräsung zu erkennen.



**Bild 9** Ausfräsung für das Mikrofon

Die CMUCAM3 dient im Ursprungsprogramm nur als Rot-Grün-Farb-Sensor.

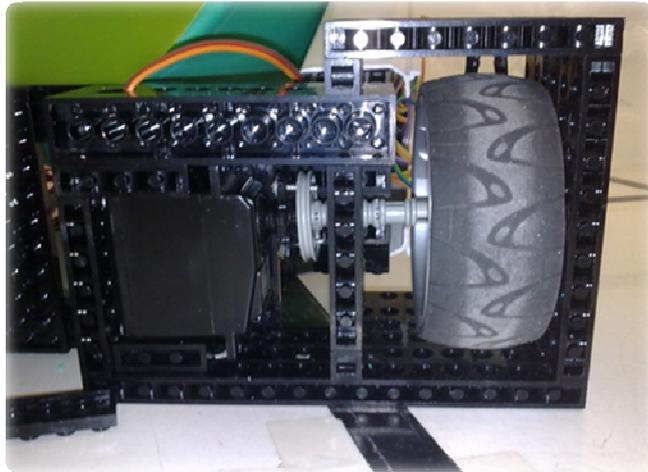
Die Bestandteile der Kamera.



**Bild 10** Kamera Bestandteile

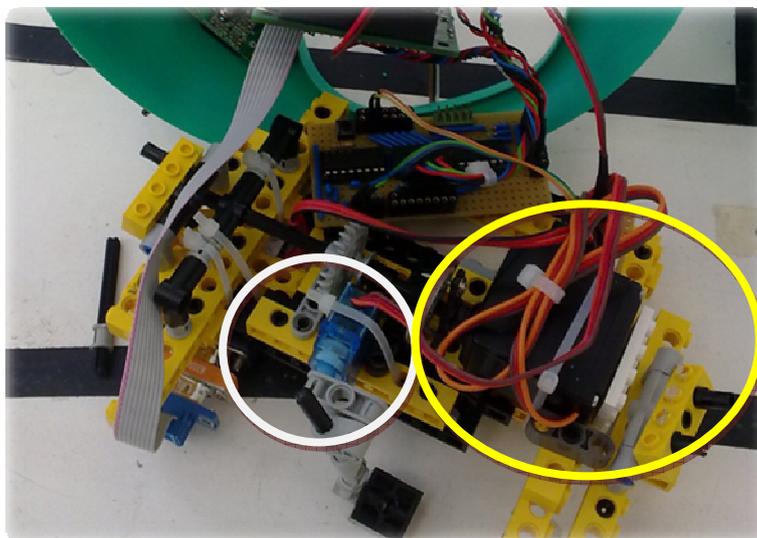
## Die Aktorik

Der Roboter besitzt 3 „gecrackte“ (360° drehbare) Servomotoren und einen normalen Servomotor. Zwei der „gecrackten“ Servomotoren dienen TBR1 als Antriebsmotor mit einer 1:1 Übersetzung. Sie kommen jeweils unterhalb der Arme, in den Radkästen, zum Einsatz.



**Bild 11** Radkasten von unten mit gecracktem Servomotor

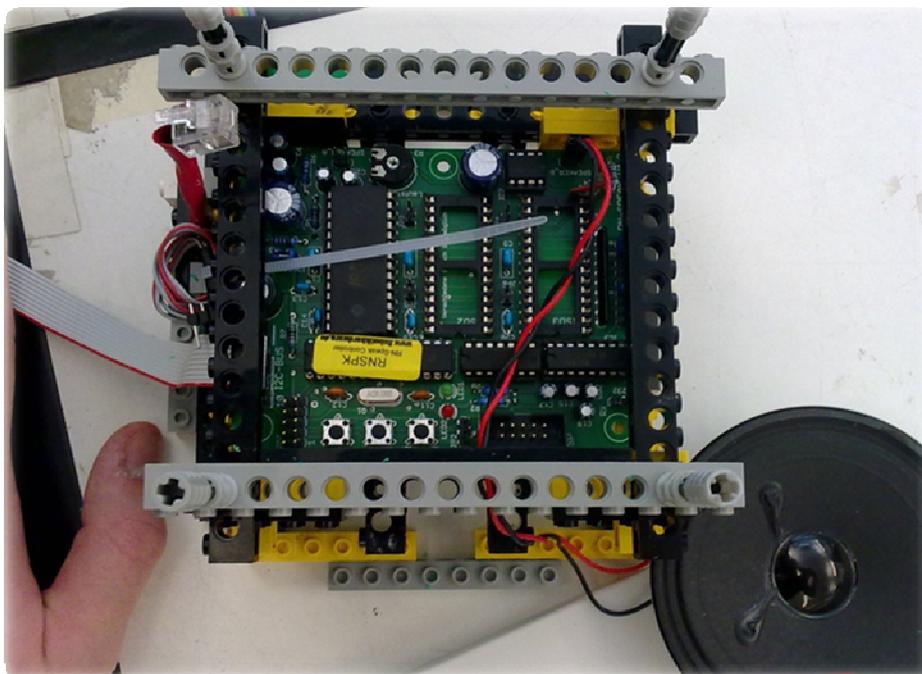
Der dritte „gecrackte“ Servomotor wird intern zum bewegen des Kopfes verwendet (siehe gelber Kreis im Bild unten). Er ist im Kopfteil des Roboters integriert, welches mit der Haube verbunden ist. Auf dem Bild ist ebenfalls der letzte Servomotor erkennbar (siehe weißer Kreis), dieser dient als Öffner / Schließer für den „Mund“ und ist mit diesem über einen Klettverschluss verbunden.



**Bild 12** Servomotoren im Kopf, links für die Mundluke und rechts für die Kopfdrehung

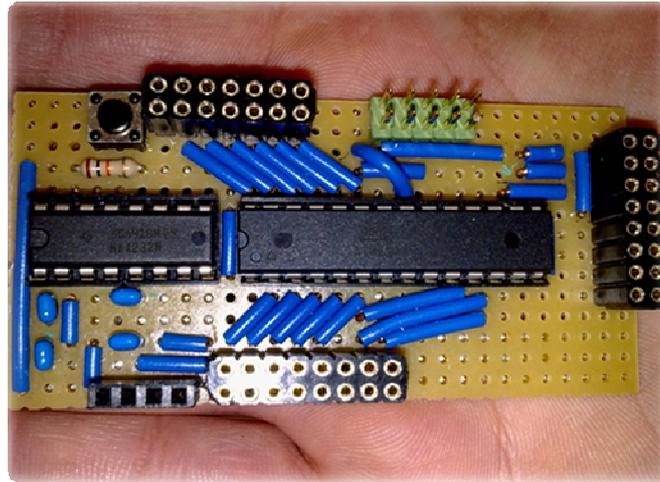
## Weitere Technik

Die akustische Interaktion findet mit Hilfe eines 8Ohm 1Watt Lautsprecher und einem RN-Speaker-Board's statt. Der interne Speicher kann bis zu 90 Sekunden Soundaufnahmen speichern. Über I<sup>2</sup>C kann man dem RN-Speaker-Board Befehle übermitteln (siehe Programmcode `playSound()` ).



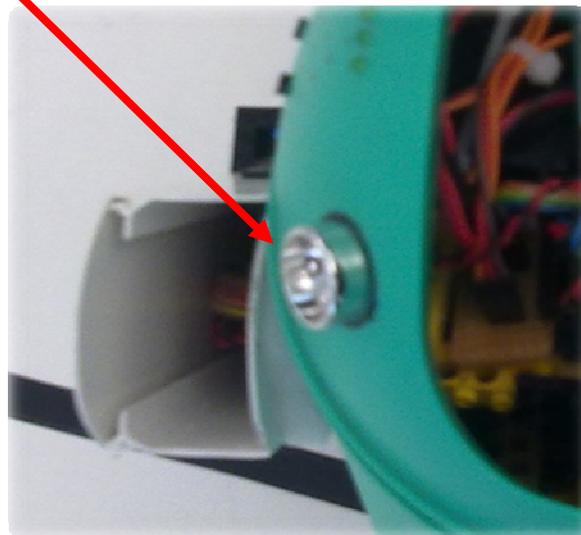
**Bild 13** RN-Speakerboard inklusive Lautsprecher

Der selbst gebastelte Headlight-Controller, zur Steuerung der LED's am Kopf, ist ebenfalls über I<sup>2</sup>C anzusprechen.



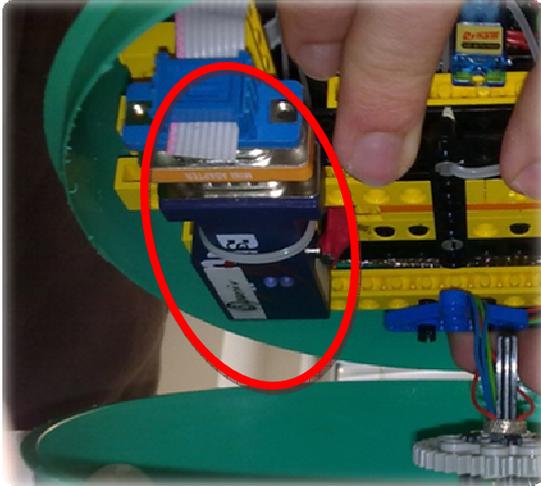
**Bild 14** Headlight-Controller

Eine dreifarbige Rainbow-LED, die am Kopf befestigt wurde, kann über den Headlight-Controller angesprochen werden.

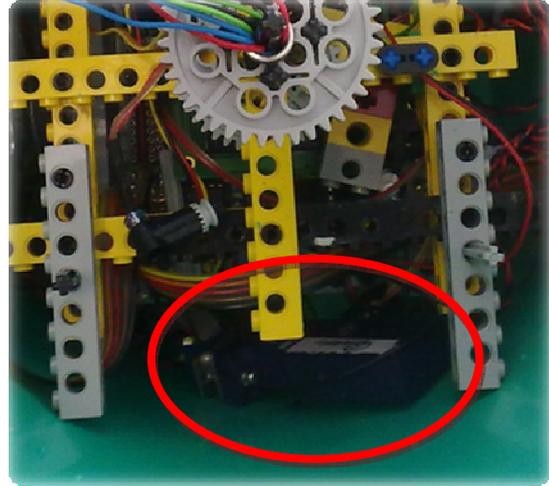


**Bild 15** Rainbow-LED

Über eine Bluetooth-Verbindung ist die Kamera aus dem Kopf mit dem Axen-Board (falls die RS232 Verbindung angeschlossen ist) verbunden. Ein Bluetooth-Adapter ist am Kopfteil des Roboters befestigt, der andere Adapter befindet sich im Rumpf des Roboters.



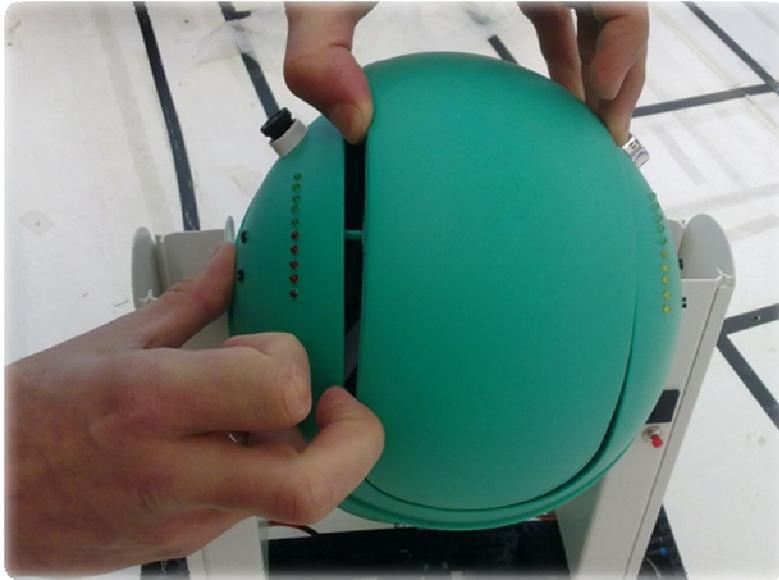
**Bild 16** Bluetooth-Adapter im Kopf



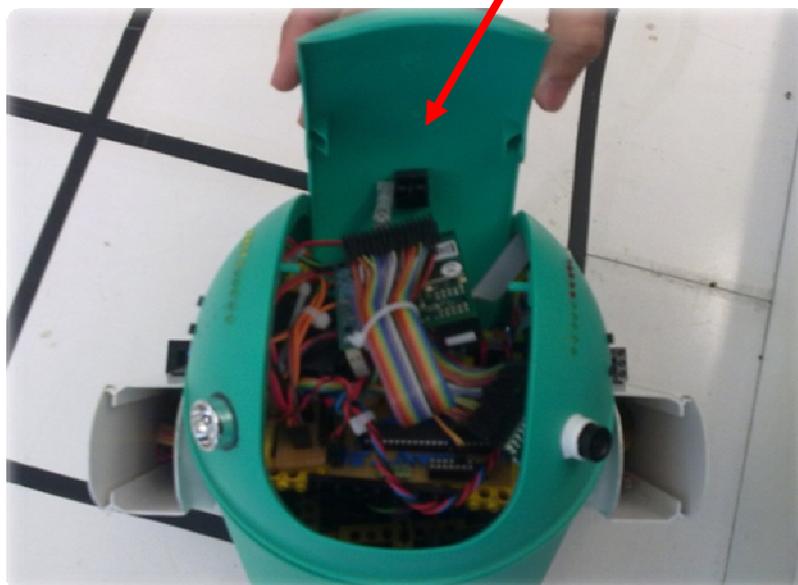
**Bild 17** Bluetooth-Adapter im Rumpf

## Anleitung zum Öffnen des Roboters

Das Öffnen des Roboters gestaltet sich ein wenig schwierig. Als erstes gilt es die Mund-Luke zu öffnen, wobei **VORSICHT** angesagt ist. Denn diese ist mit einem kleinen Servomotor über einen starken Klettverschluss verbunden. Die Mund-Luke kann man an den Seiten aushängen wie im nachfolgenden Bild gezeigt.

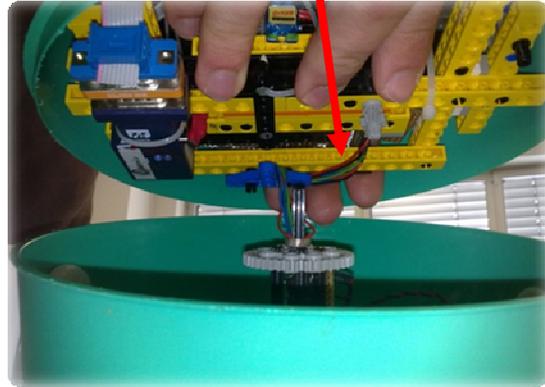
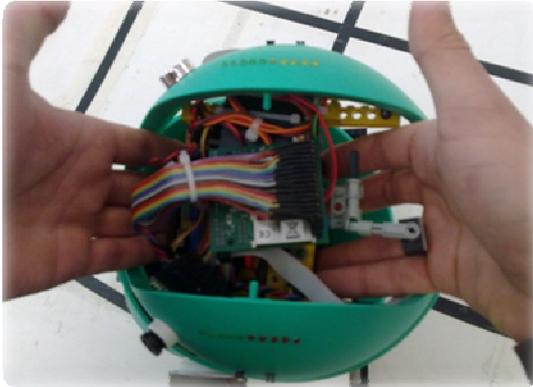


Nachdem beide Seiten der Mund-Lucke ausgehängen sind, kann man diese nach hinten kippen (siehe nachfolgendes Bild). Man erkennt die Befestigung des Servomotors an der Mund-Luke. Diese gilt es zu lösen.

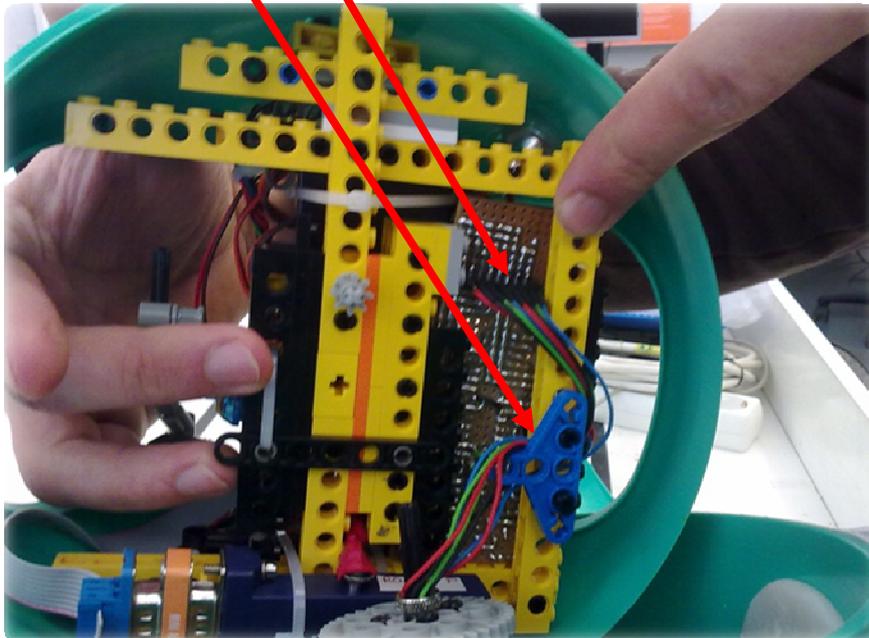




Mit einem geübten Griff lässt sich der Kopf vom weiteren Rumpf nun leicht trennen. Auf den Bildern sieht man, wie dieser Griff aussieht. Es gilt beim Trennen des Kopfes vom Rest des Rumpfes **VORSICHT** -> Diesen nicht gleich weg nehmen, da Kabelverbindungen von der Schleifkontaktstange zum Kopf bestehen.

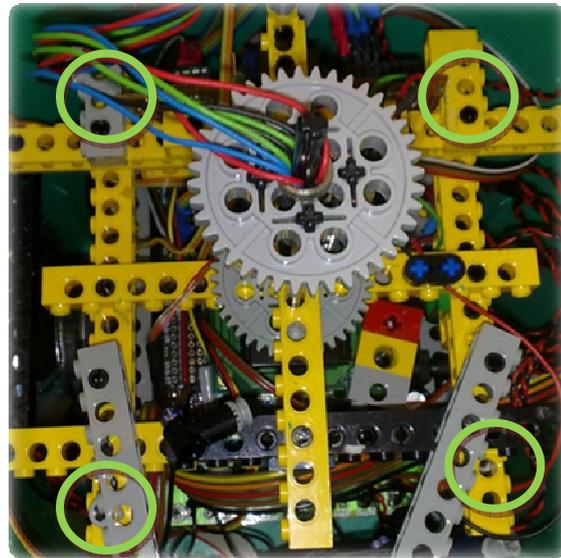
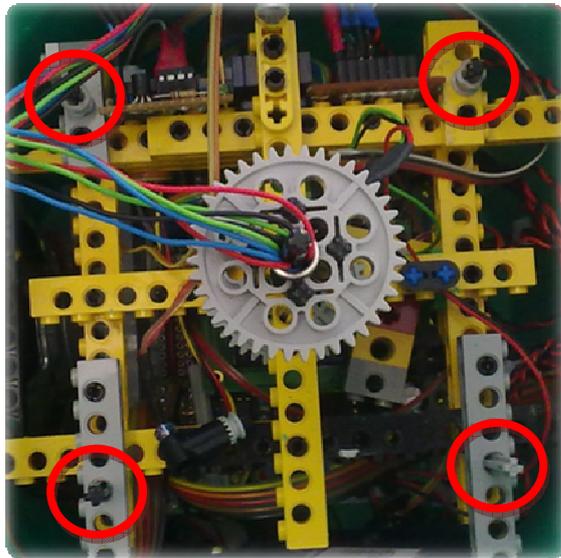


Die Kabelhalterung / -verbindung muss getrennt werden, um den Kopf komplett von dem Rumpf zu entfernen.

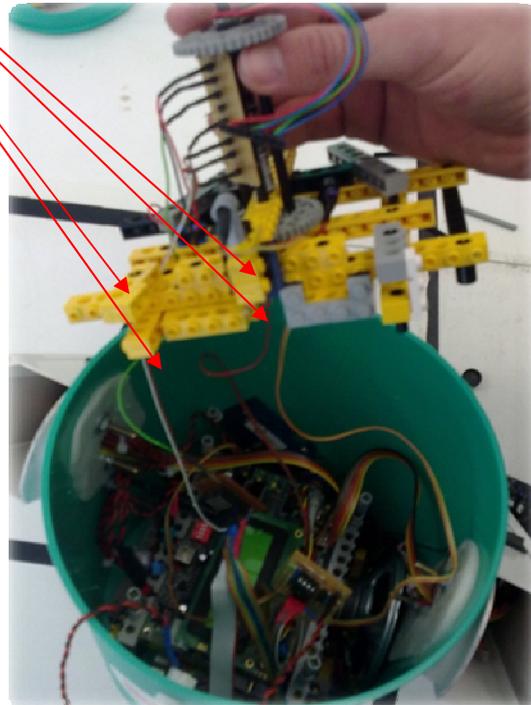
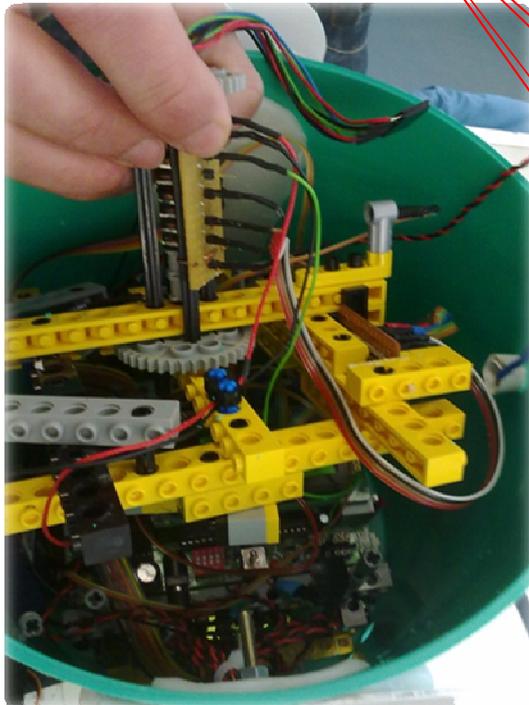


Nun kann man den Kopf beiseite packen, die CMUCAM3 flashen oder den Kopf weiter auseinander nehmen, in dem man die Einrasthalterungen an den Seiten des Kopfes löst.

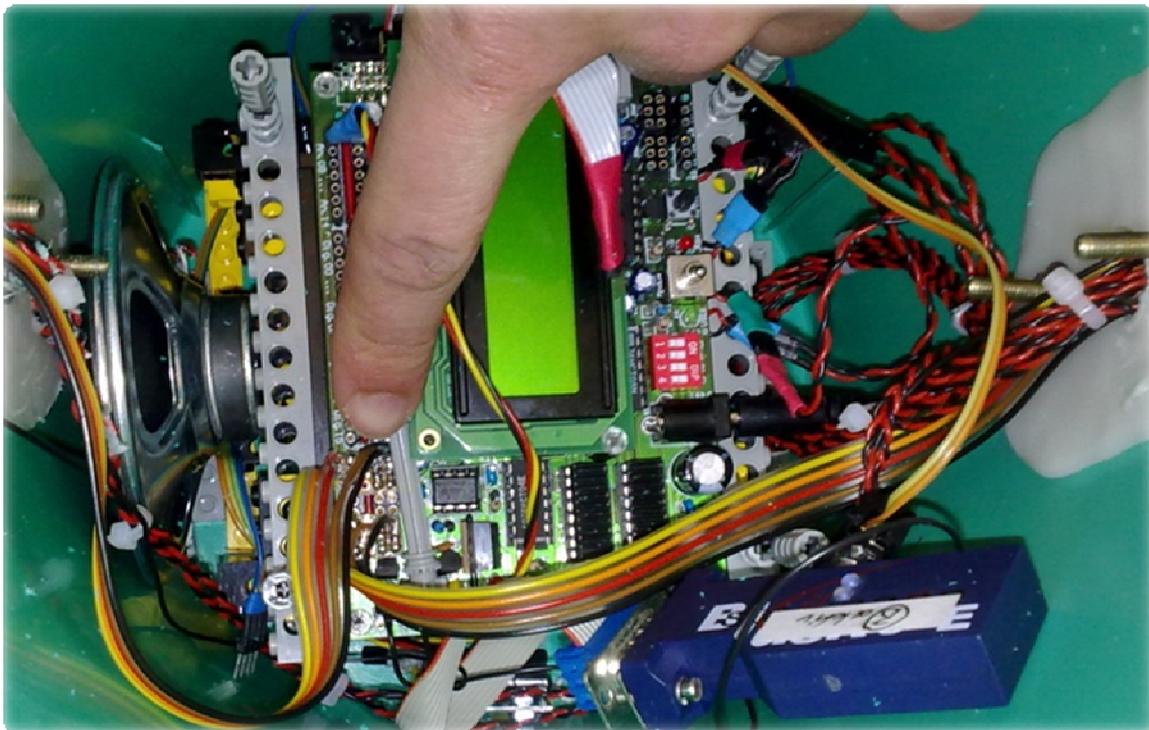
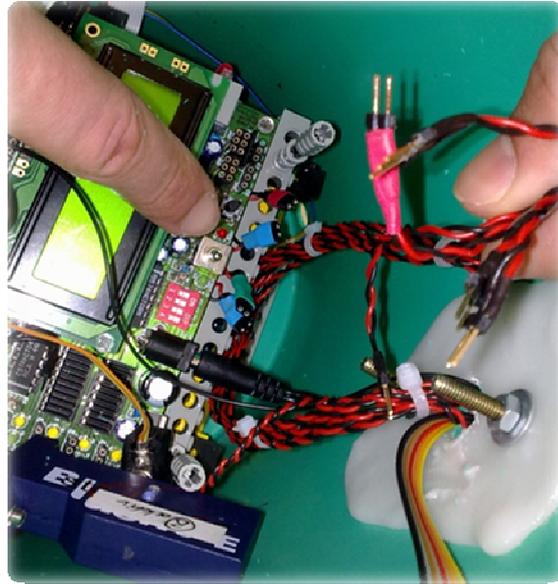
Um den Rumpf des Roboters zu leeren, müssen als erstes 4 Fixierungen entfernt werden.



Diese vier Stangen kann man einfach mit den Fingern heraus ziehen. Als nächstes kann man den oberen Teil des Roboters anheben. Hier ist auch wieder **VORSICHT** angesagt, denn es bestehen wieder Kabelverbindungen, die es vor der kompletten Entnahme zu lösen gilt.

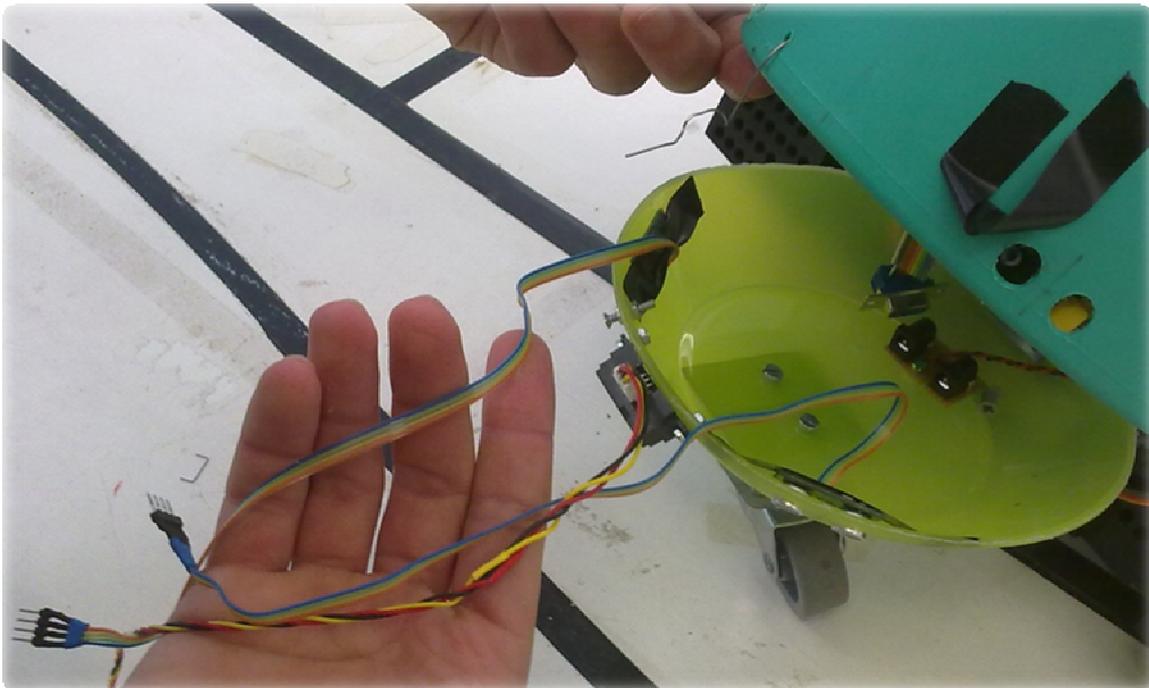
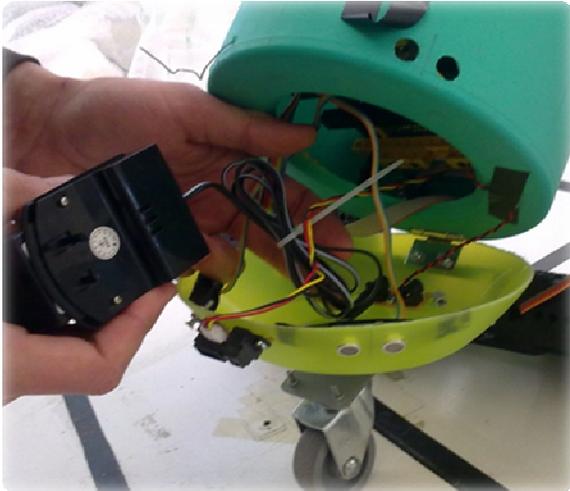


Danach löst man die weitere Verkabelung, den I<sup>2</sup>C-BUS, die Kabelleitung für die Bedienelemente an den Armen, sowie die Kabel für die Motoren und die IR-Sensoren in den Radkästen.



Dann kann der Bluetooth-Adapter vom Aksen-Board sowie von der RS232-Schnittstelle entfernt werden.

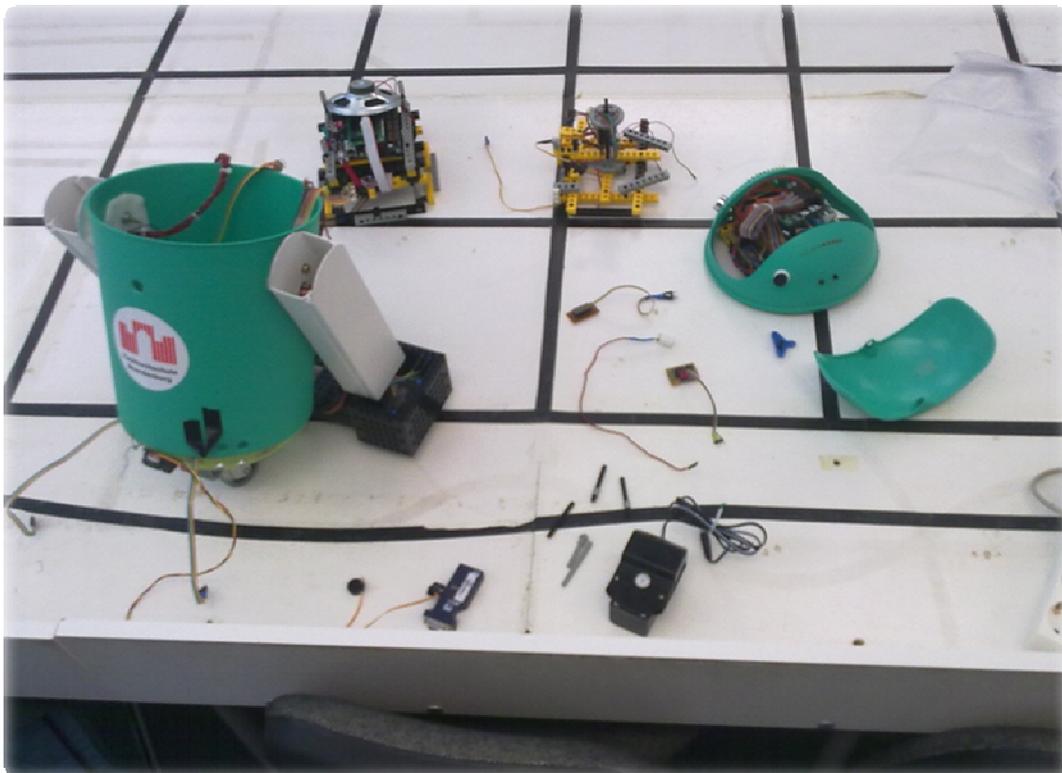
Nachdem alles vom Aksen-Board abgesteckt ist, auch das RJ25-Kabel an der Seite, muss die außenstehende RS232-Verbindung noch getrennt werden. Anschließend kann der Roboter an der Vorderseite geöffnet werden um den Akku zu entnehmen und um die Sensor-Kabel **VORSICHTIG** heraus zu ziehen.



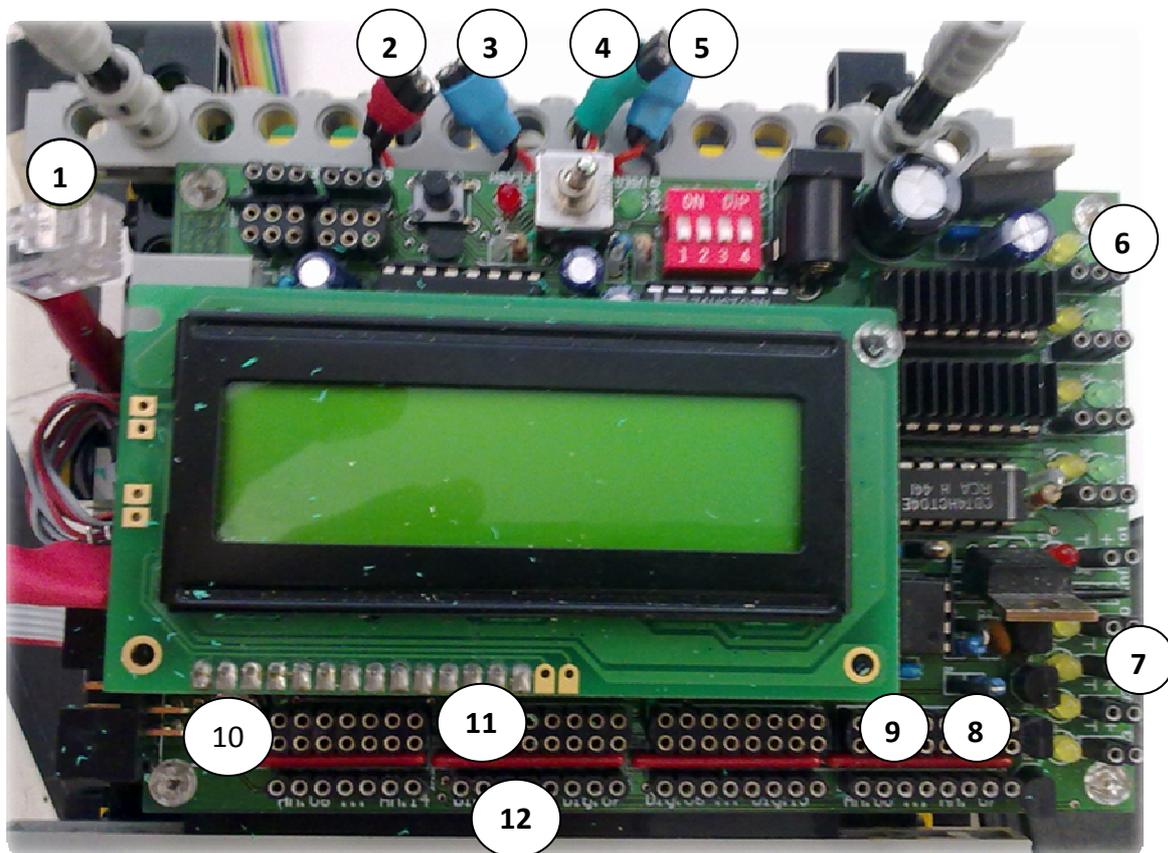
Nun kann man den Rest des Innenlebens des Roboters aus dem Rumpf entnehmen, allerdings ist bei den Verschraubungen der Arme eine Stelle erreicht, welche ein wenig Feingefühl erfordert.



**Nun hat man den Roboter „komplett“ auseinander genommen.**



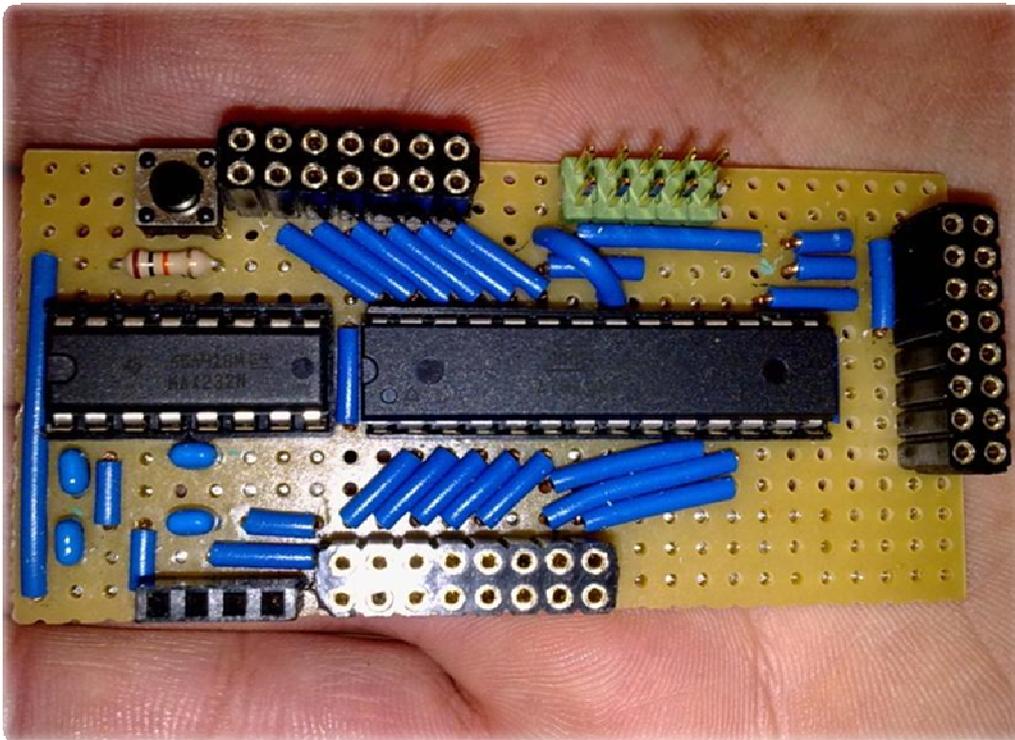
## Genutzte Anschlüsse auf dem AKSEN-Board



1. RS232 (Programmier- / Kamera-Schnittstelle)
2. Resetknopf am Bein
3. Rote LED am Bein
4. Modusschalter am Bein
5. Grüne LED am Bein
6. Motoren (für TCR1 Beispielprogramm rotes Kabel nach innen)
7. Sonarstromversorgung
8. Sonar rechtes Bein
9. Sonar linkes Bein
10. Sonar vorne
11. I<sup>2</sup>C (Digit\_In 0-1)
12. Clapp-Sensor (Digit\_In 2)

## Schaltpläne der angefertigten Controller

### Headlight - Controller



#### Grundlegende Beschreibung

Der Headlight-Controller ist eigens für TBR1 entwickelt worden. Er übernimmt in dessen Zusammenhang die Ansteuerung der einzelnen LED's im Kopf von TBR1. Da aber alle PORTs vollständig nach außen geführt sind und diese individuell belegbar sind, ist der Headlight Controller im Ganzen ein eigenständiges Atmega8-Experimentierboard. Programmiert wird der Controller über die vorhandene ISP (In System Programmable) Schnittstelle. Der MAX232 auf dem Board ist für die RS232 Pegelumwandlung einer eventuellen UART Verbindung mit einem PC zuständig. Falls es passieren sollte, dass der Atmega8 nicht mehr reagiert, weil er in einer Endlosschleife stecken geblieben ist, ist es mittels Taster S1 möglich, den Mikrocontroller neu zu starten, Für weitere Möglichkeiten und Informationen ist der Schaltplan und das Datenblatt des Mikrocontrollers zu Rate zu ziehen. Diese extra Features auf dem Board geben dem Headlight-Controller eine gewisse Flexibilität, so dass er auch schnell für andere Aufgaben umprogrammiert werden kann. Um den Atmega8 auf dem Board neu zu programmieren, ist ein ISP-Programmer von Nöten. Bau- und Schaltpläne dieses Programmers gibt es auf etlichen Internetseiten.

## Funktionsbeschreibung in TBR1

### *Verkabelung der LED's*

Wie schon erwähnt, übernimmt der Headlight Controller die Ansteuerung der Kopf-LED's im Roboter TBR1. Hierzu muss vorerst sicher gestellt sein, dass die 7-poligen Stecker der LED-Leisten richtig im Buchsenstecker des Controllers stecken. Der Stecker der Rot-Grünen Leiste muss in den Port B und die Gelb-Grüne Leiste in den Port D des Controllers. Hierbei ist noch zu beachten, dass die beiden schwarzen Kabel des jeweiligen Steckers in die inneren Buchsen von PB7 und PB6 bzw. PD7 und PD6 zu stecken sind. Die Rainbow-LED des Roboters muss in die Buchse PB0. Da diese eine LED ist, muss auf die Polung geachtet werden. Das rote Kabel der Rainbow LED muss in die äußere und das schwarze Kabel in die Innere Buchse von PB0.

### *Verkabelung des I<sup>2</sup>C Busses*

Der I<sup>2</sup>C Bus besteht aus einer CLOCK- und einer DATEN-Leitung. Diese sind in die dafür vorgesehenen Buchsen des Controllers zu stecken. Die CLOCK-Leitung in PC5 (SCL) und die DATA-Leitung in PC4 (SDA).

### *Inbetriebnahme des Headlight Controllers*

Wenn die Verkabelung der LED's und des I<sup>2</sup>C Busses vorgenommen wurde, kann der Headlight-Controller in Betrieb genommen werden. Hierzu ist es nur nötig, die erforderliche Betriebsspannung, von minimal 4,5V – maximal 6V, an die vorgesehenen Buchsen GND und +5V anzuschließen. Diese Buchsen liegen auf dem Controller an Buchse 1 und 2 von JP5. Da der Headlight Controller keinen Verpolungsschutz besitzt, sollte hierbei mit Bedacht gearbeitet werden. Sobald dies getan ist, arbeitet der Controller. Sollte es einmal passieren, dass der Controller nicht richtig arbeitet, ist es möglich, ihn mittels S1 neu zu starten ohne ihn von der Betriebsspannung zu lösen. Nun erwartet der Atmega8, mit der I<sup>2</sup>C Slave Adresse 100, Befehle. Gültige Befehle sind der angefügten Befehlsliste zu entnehmen.

## Befehlsliste

Der Headlight Controller fungiert als I<sup>2</sup>C Slave und erwartet darüber auch seine Befehle. Seine Adresse ist die 100. Diese ist aber auch individuell durch neuprogrammierung Einstellbar.

Hier nun die komplette Befehlsliste:

61h	
62h	
63h	
64h	
65h	
66h	
67h	
68h	
69h	
70h	
71h	
72h	
73h	
74h	
75h	
76h	
77h	
78h	
79h	Rainbow LED ON
80h	Rainbow LED OFF

## Stückliste

C1 = 0,1 µF

C2 = 0,1 µF

C3 = 0,1 µF

C4 = 0,1 µF

R1 = 10 kΩ

IC1 = Atmega8 16PU

IC2 = MAX232N

S1 = Taster

JP1 = 8x2 Buchsenleiste ( 2,54" )

JP2 = 8x2 Buchsenleiste ( 2,54" )

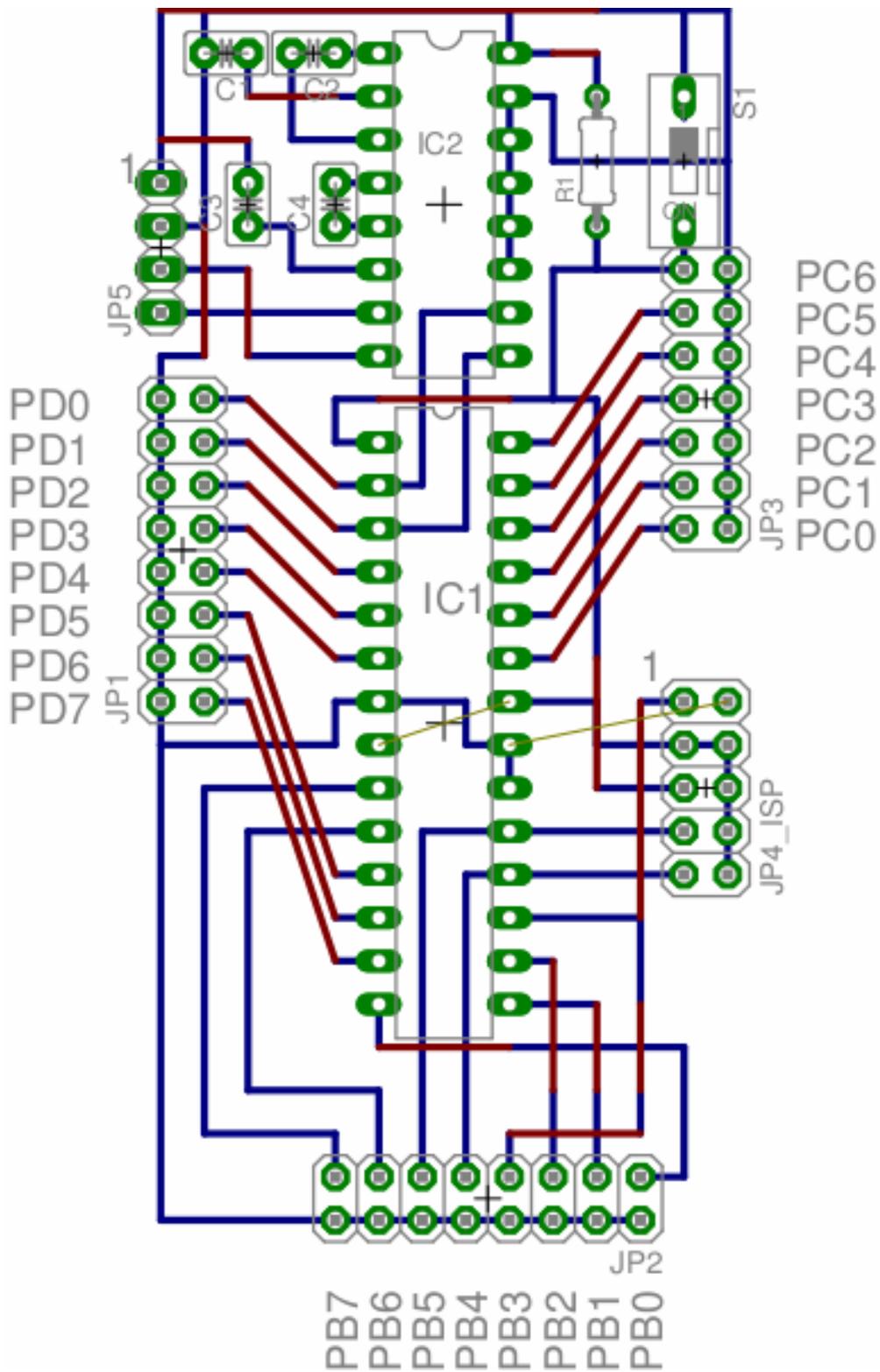
JP3 = 7x2 Buchsenleiste ( 2,54" )

JP4 = 5x2 Stiftleiste ( 2,54" )

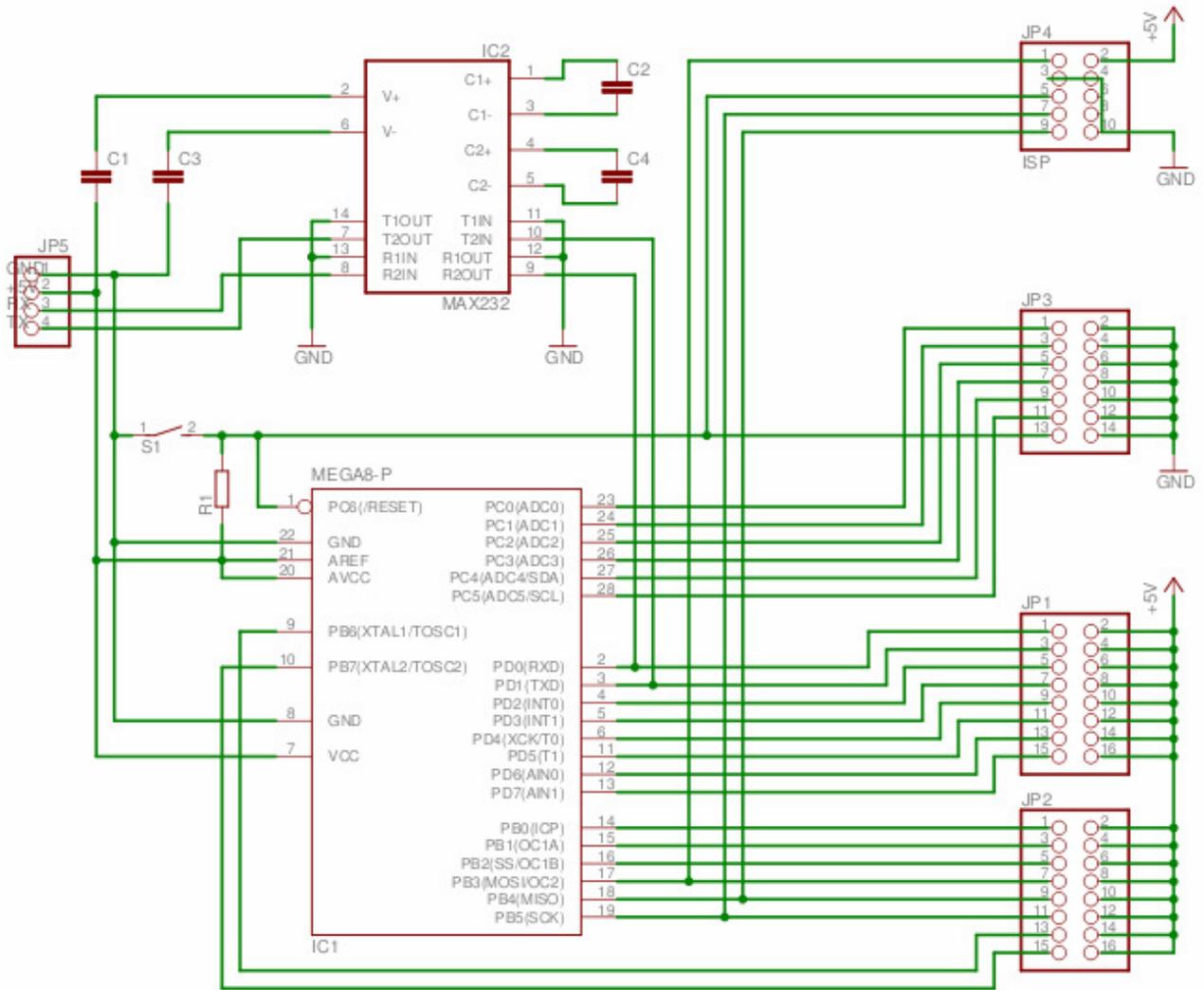
JP5 = 4x1 Buchsenleiste ( 2,54" )

## Platinenlayout

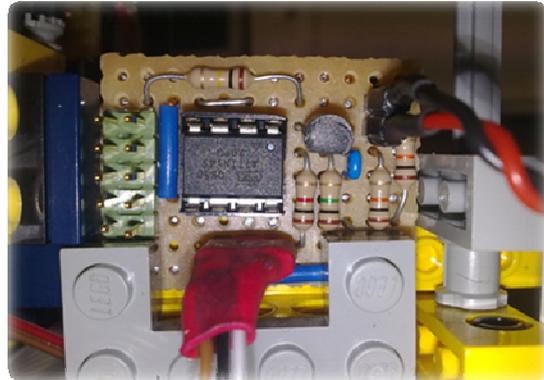
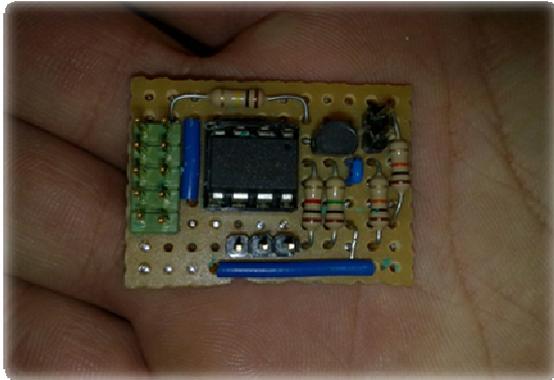
Ansicht von Oben!



## Schaltplan



## Clapper - Controller



### Grundlegende Funktionsbeschreibung

Der Clapper Controller ist ebenfalls eigens für TBR1 entwickelt worden. Er dient zum Detektieren lauter Geräusche in der Umgebung, wie zum Beispiel einem Klatschen. Zusätzlich ist ein kleines Kondensatormikrofon von Nöten, welches auf JP1 gesteckt werden muss. Sobald die erforderliche Betriebsspannung von 5V an JP3 angeschlossen ist, fängt der Attiny45 auf dem Board an zu arbeiten. Nun wartet er bis ein lautes Geräusch vom Mikrofon detektiert wird. Falls dies geschieht, legt der Microcontroller den PIN 2 von JP3 für einen kurzen Moment auf High-Pegel. Sollte man eine andere Funktion implementieren wollen, ist hierfür eine ISP (In System Programmable) auf dem Board vorhanden (JP2). Um den Attiny45 auf dem Board neu zu programmieren ist ein ISP-Programmer von Nöten. Bau- und Schaltpläne dieses Programmers gibt es auf etlichen Internetseiten.

### Stückliste

C1 = 0,1 µF

R1 = 1 MΩ

R2 = 1 kΩ

R3 = 100 kΩ

R4 = 10 kΩ

R5 = 10 kΩ

T1 = BC548

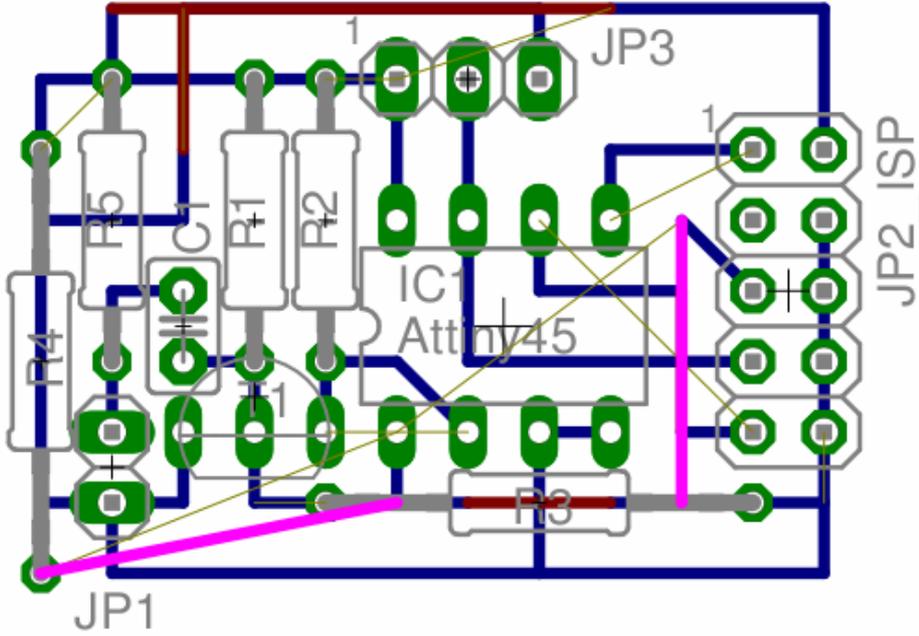
IC1 = Attiny45

JP1 = 1x2 Steckleiste ( 2,54" )

JP2 = 2x5 Steckleiste ( 2,54" )

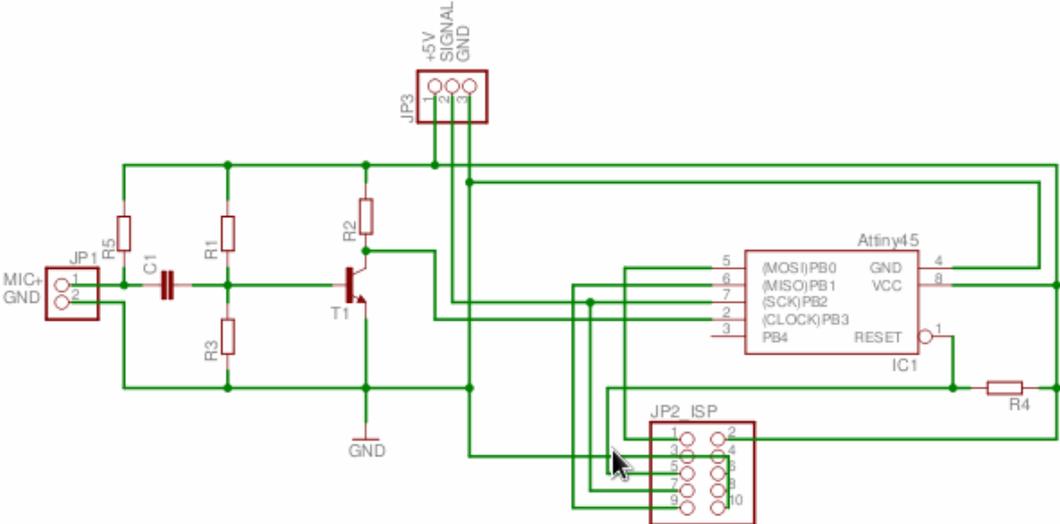
JP3 = 1x3 Steckleiste ( 2,54" )

Platinenlayout



altplan

Sch



## Nützliche Information

Den vorhandenen Quellcode könnte man als API nutzen und darauf aufbauend könnte man intelligentes Verhalten, mit Benutzer-Interaktion über Audio- sowie Kamera-Verarbeitung, implementieren.

### *Kommentar zum Headlight-Controller-Quelltext*

Der Quelltext bietet schon etwas mehr Funktionalität als für TBR1 nötig. Eigentlich war geplant noch Befehle und Funktionen, für das Senden an den I<sup>2</sup>C Master, einzubauen. Aus Zeitgründen konnte diese Funktion aber leider noch nicht vollständig implementiert werden.