HErd

HErd of educational robotic devices

Andreas Bihlmaier, Michael Vollert

8. Juni 2012

Karlsruher Institut für Technologie

Übersicht

1. About us
2. Das HERD Projekt
3. Hardware
4. Software
5. LiveCD

1 About us
2 Das HERD Projekt
3 Hardware
4 Software
5 LiveCD

Who to blame

30 seconds of vanity.

2 Das HERD Projekt

1 About us

3 Hardware

4 Software

5 LiveCD

Das HERD Projekt: Die Anfänge

Motivation

- Bewegende Roboter ein ansprechenderer Zugang zur Informatik.
- Kooperierende (Schwarm-)Roboter umso mehr.

Randbedingungen

- Zielgruppe: Schüler mit Anfängerkenntnissen im Programmieren (Kontrollstrukturen, Funktionen).
- Kostensensitives Umfeld (z.B. Mindstorms zu teuer)
- Keine Zusatzanschaffung für (technische) Schulen, sondern Ersatz für μC-Boards.
- Günstige Erweiterungsfähigkeit mit Sensoren und Anzeigen.

Das HERD Projekt: Die Anfänge

Motivation

- Bewegende Roboter ein ansprechenderer Zugang zur Informatik.
- Kooperierende (Schwarm-)Roboter umso mehr.

Randbedingungen

- Zielgruppe: Schüler mit Anfängerkenntnissen im Programmieren (Kontrollstrukturen, Funktionen).
- Kostensensitives Umfeld (z.B. Mindstorms zu teuer)
- Keine Zusatzanschaffung für (technische) Schulen, sondern Ersatz für μC-Boards.
- Günstige Erweiterungsfähigkeit mit Sensoren und Anzeigen.

Das HERD Projekt

- **H**ERD of **E**ducational **R**obotic **D**evices
  `herd-project.org`
- "open hard- and software of simple, small and cheap **swarm robots** designated for high school education"
- Open-{Source,Hardware} (durch und durch!)
- So billig ($\leq$ 50 Euro) und einfach (mechanisch wie elektronisch) wie möglich
  - kein Getriebe
  - fast nur ICs in alter DIP Bauform
  - einzelne Europlatine, selbstgeätzt, gebohrt und gelötet
- Eigene LiveCD mit allem Benötigten

Hardwarekosten der Entwicklung finanziert durch die *Begabtenstiftung Informatik Karlsruhe*. Vielen Dank.

Das HERD Projekt

- **H**ERD of **E**ducational **R**obotic **D**evices
  `herd-project.org`
- "open hard- and software of simple, small and cheap **swarm robots** designated for high school education"
- Open-{Source,Hardware} (durch und durch!)
- So billig ($\leq$ 50 Euro) und einfach (mechanisch wie elektronisch) wie möglich
  - kein Getriebe
  - fast nur ICs in alter DIP Bauform
  - einzelne Europlatine, selbstgeätzt, gebohrt und gelötet
- Eigene LiveCD mit allem Benötigten

Hardwarekosten der Entwicklung finanziert durch die *Begabtenstiftung Informatik Karlsruhe*. Vielen Dank.

Das HERD Projekt

- **H**ERD of **E**ducational **R**obotic **D**evices
  `herd-project.org`
- "open hard- and software of simple, small and cheap **swarm robots** designated for high school education"
- Open-{Source,Hardware} (durch und durch!)
- So billig ($\leq$ 50 Euro) und einfach (mechanisch wie elektronisch) wie möglich
  - kein Getriebe
  - fast nur ICs in alter DIP Bauform
  - einzelne Europlatine, selbstgeätzt, gebohrt und gelötet
- Eigene LiveCD mit allem Benötigten

TODO:

- Selbe Prinzipien und Schaltplan, jedoch auf billige (Klein)serienfertigung ausgerichtet (SMD, weniger Handarbeit, ordentliche Halterungen).
- Laden und programmieren über USB (wird vermutlich noch auf GPN erledigt)

3 Hardware
- top
- middle
- bottom

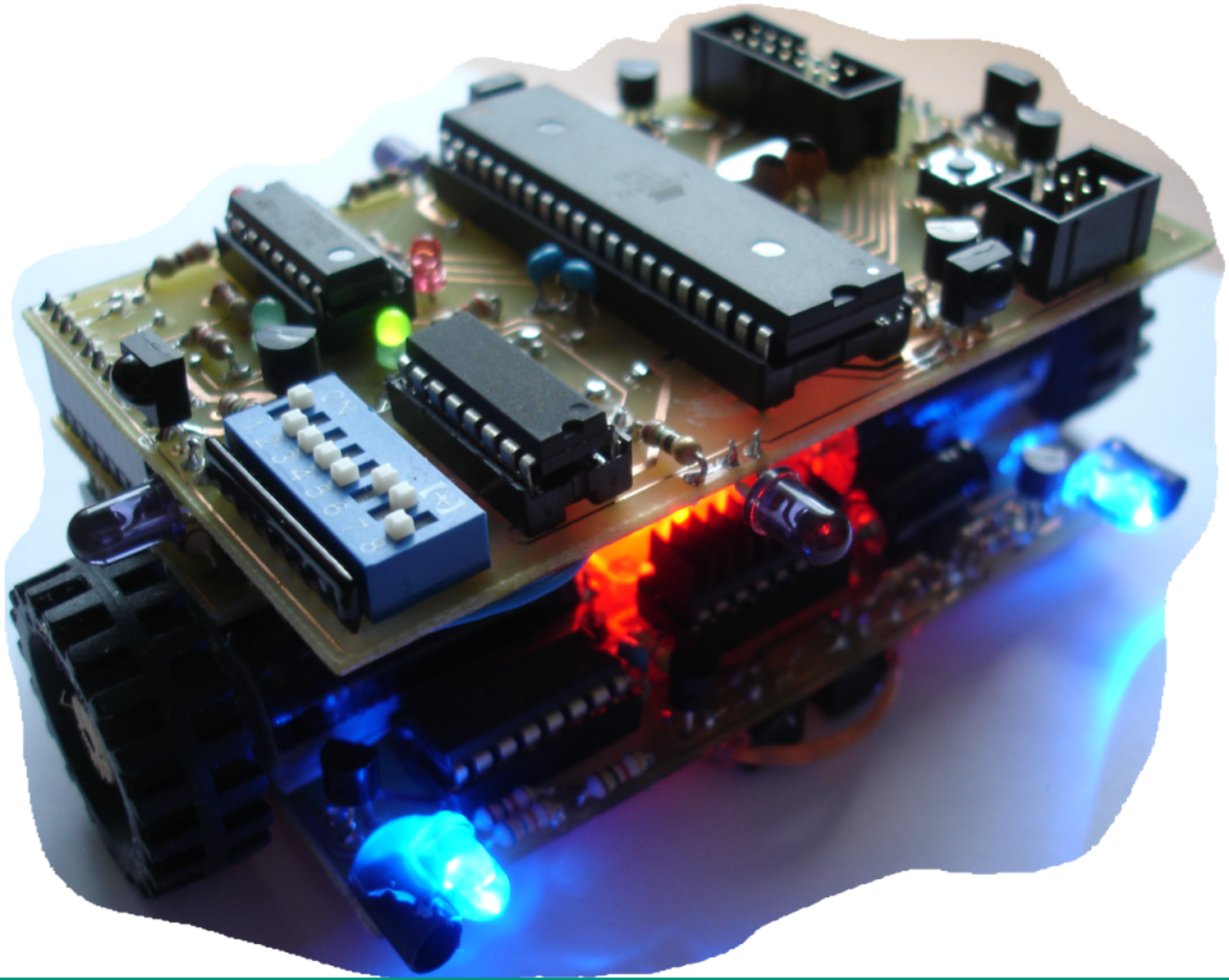
4 Software

5 LiveCD

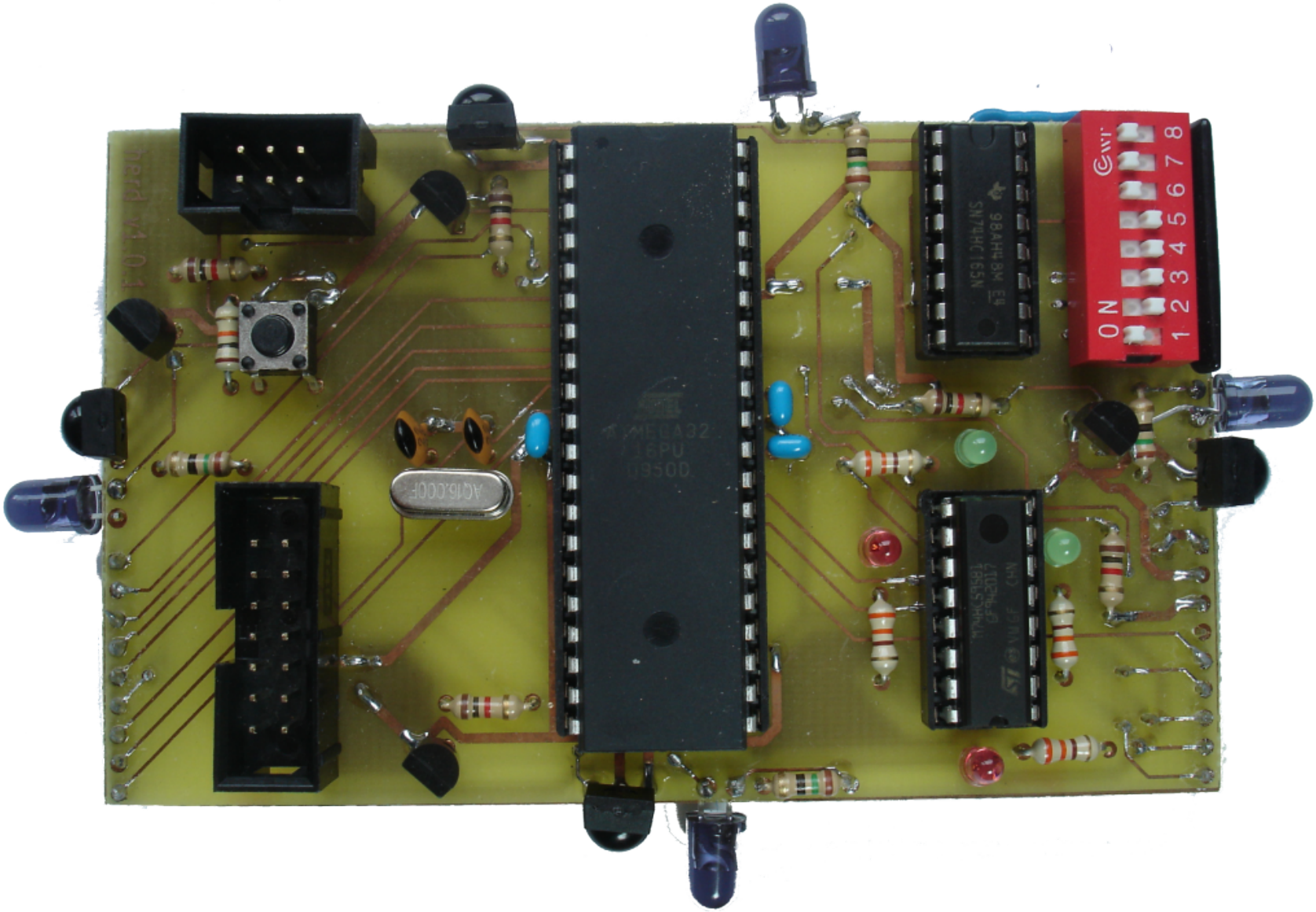
1 About us

2 Das HERD Projekt

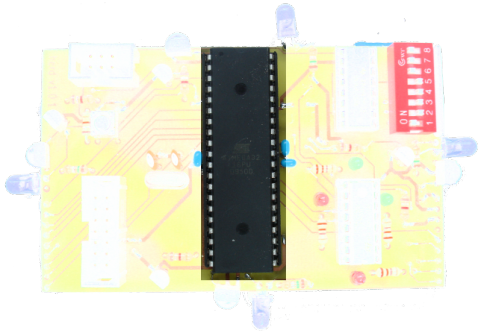
Hardwarekomponenten: Überblick

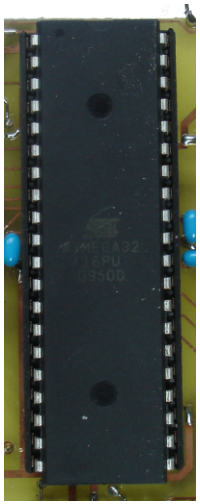


Hardwarekomponenten: top Platine

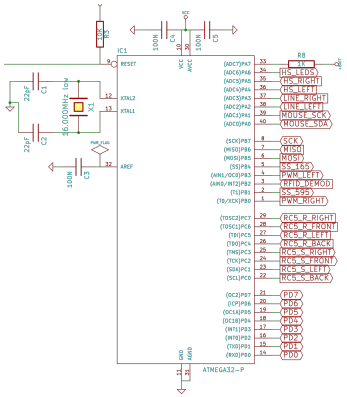


Hardwarekomponenten: Der Mikrocontroller

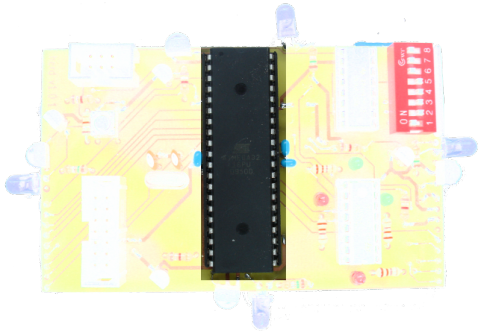


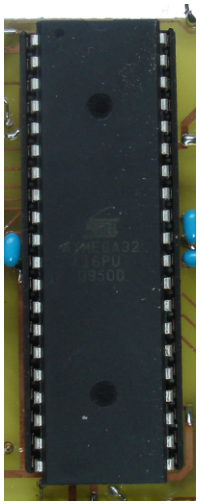


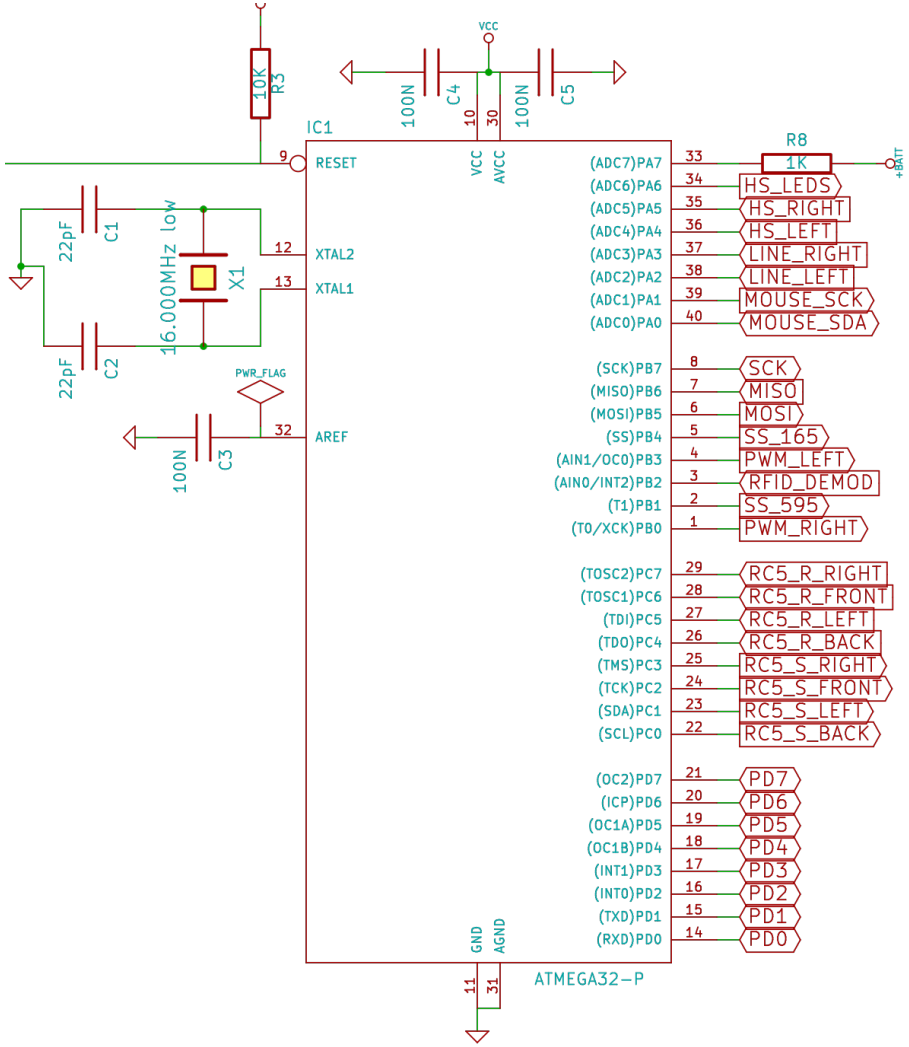
- ATMega32 (32K ROM, 2K RAM, 1K EEPROM, 16 MHz)



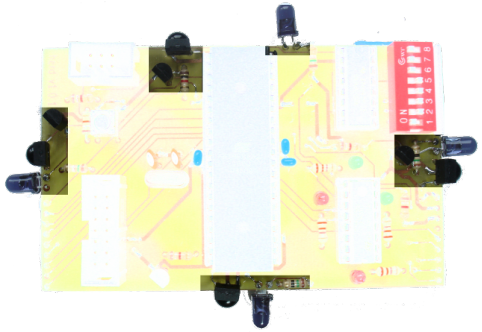
Hardwarekomponenten: Der Mikrocontroller







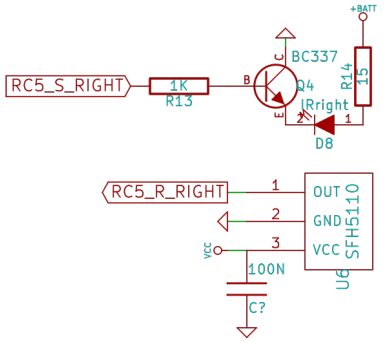
Hardwarekomponenten: Die IR Kommunikation

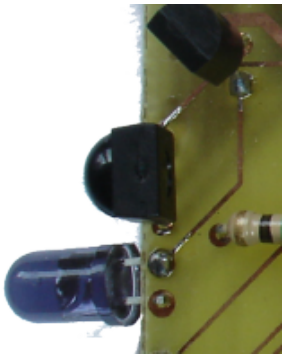


Auf jeder Seite

- IR Empfänger
- IR Sender

auf Basis des RC5 Standard.



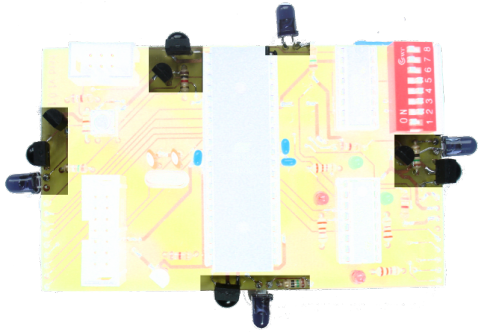


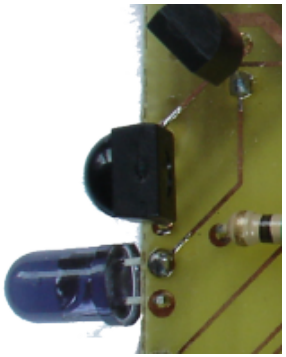
TODO?*:

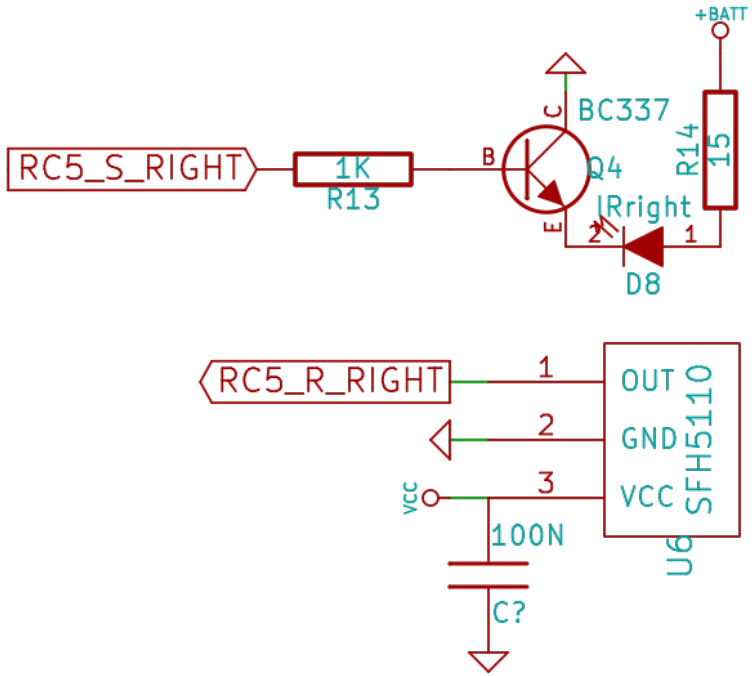
- Separater ATtiny für low-level processing
- 6 statt 4 Transceiver

* Kosten-Nutzen?

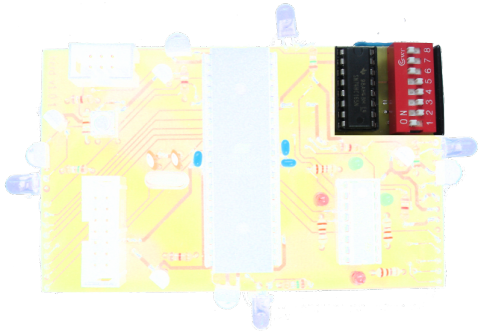
Hardwarekomponenten: Die IR Kommunikation

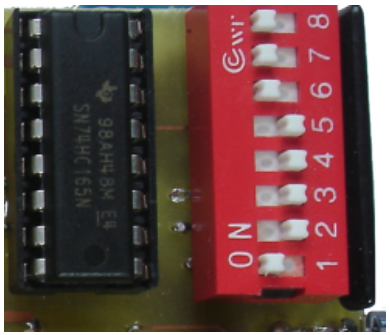






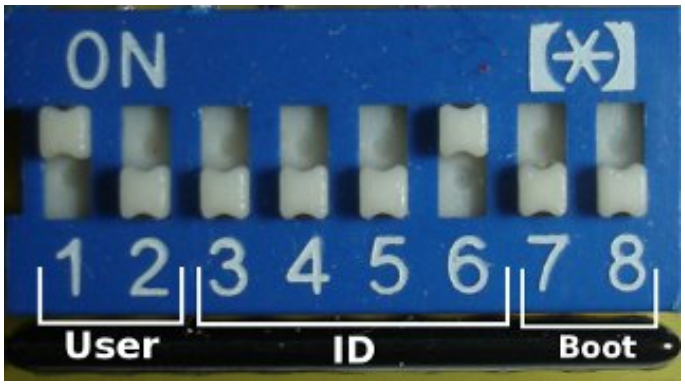
Hardwarekomponenten: Die Switches

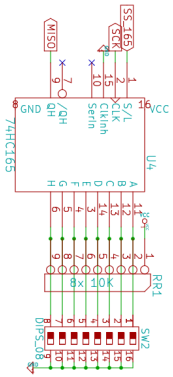




Zur Kodierung von

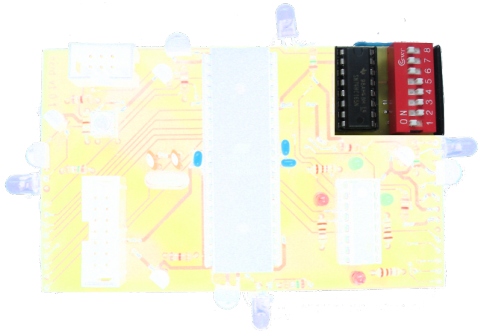
- Bootloader
- `id()`
- benutzerspezifischen Dingen

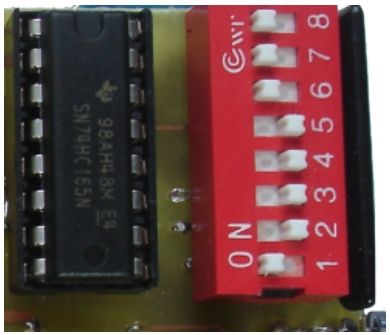


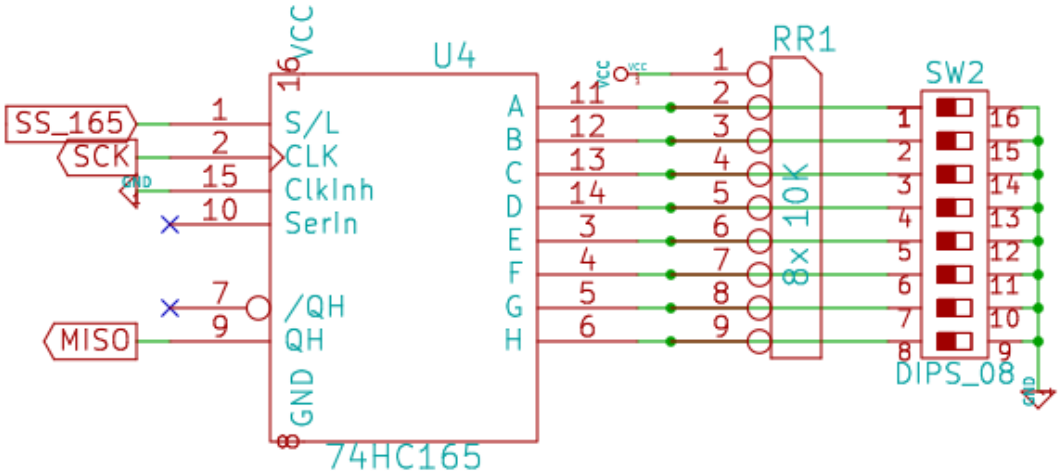


Eingelesen durch Schieberegister (um Eingänge am μC zu sparen).

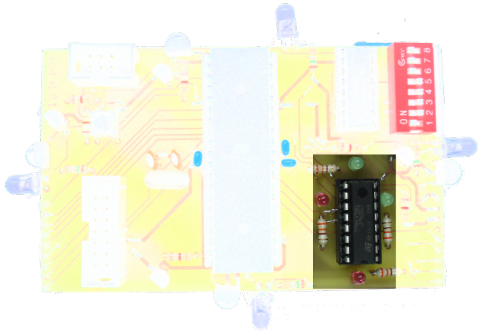
Hardwarekomponenten: Die Switches

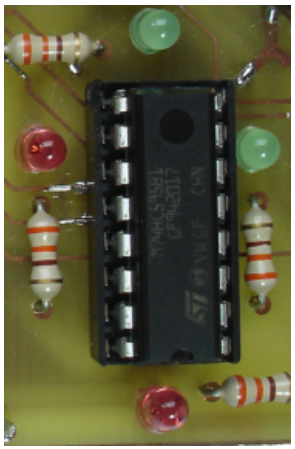






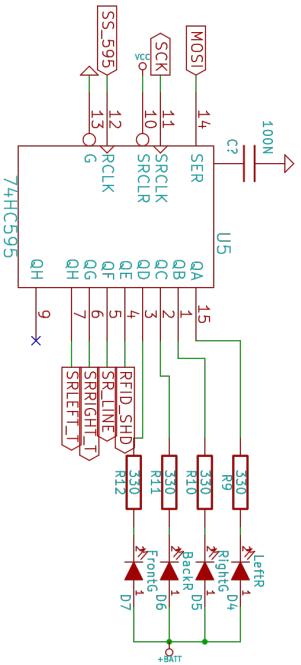
Hardwarekomponenten: Visualisierung LEDs





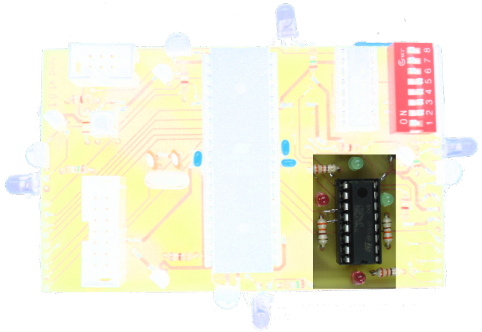
Zur benutzerspezifischen Verwendung, z.B.

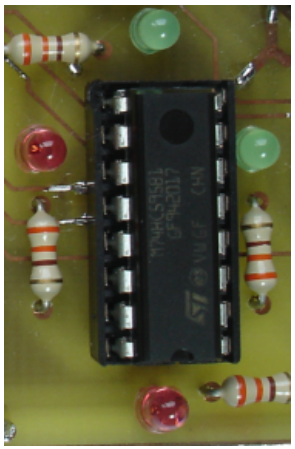
- Status- und Richtungsanzeige
- Fehlerfälle

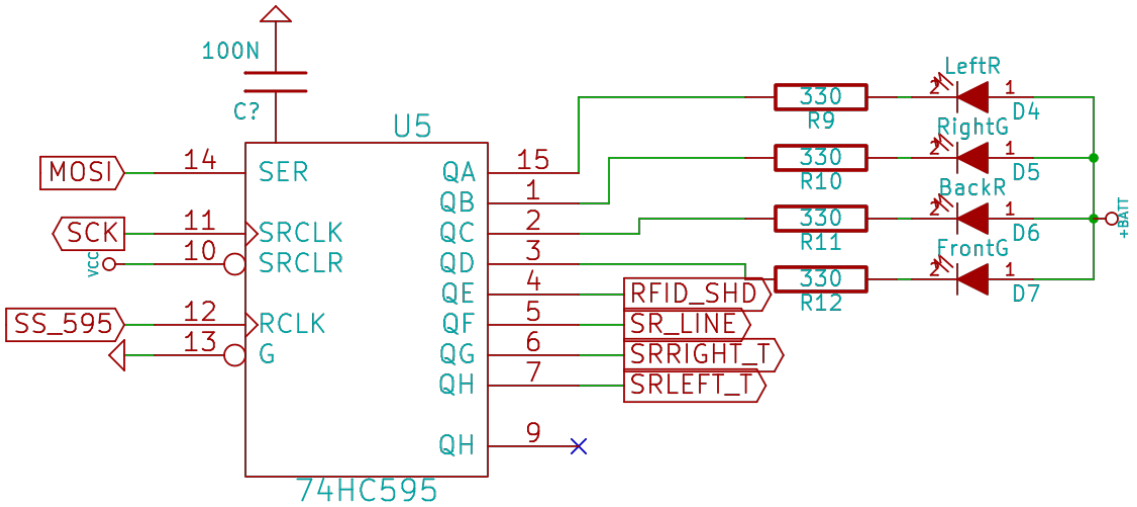


Angesteuert durch Schieberegister (um Ausgänge am μC zu sparen).

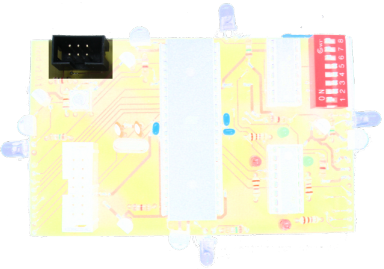
Hardwarekomponenten: Visualisierung LEDs



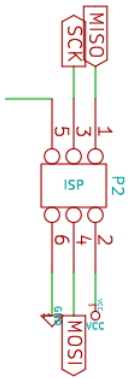




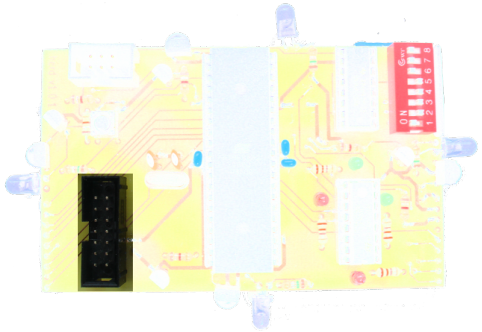
Hardwarekomponenten: Programmierport



Zur Übertragung der am PC kompilierten Programme auf den Roboter.

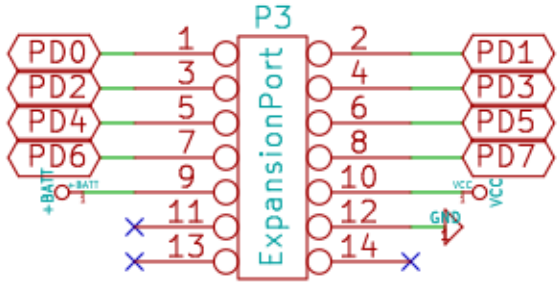


Hardwarekomponenten: Erweiterungsport



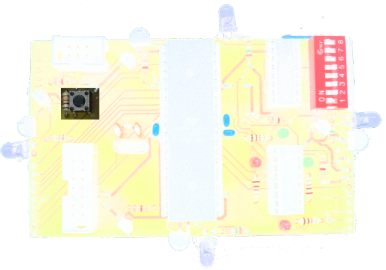
Für (TODO: teilweise noch zu entwerfende) Erweiterungsmodule, wie

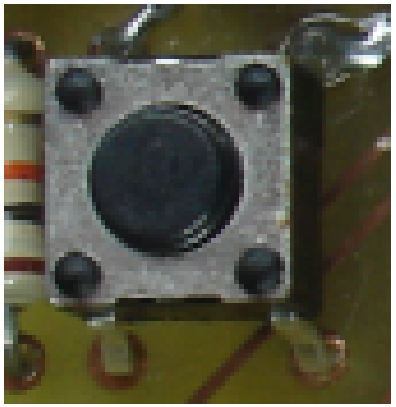
- Sensoren (Licht, Schall, Distanz)
- serielle Schnittstelle
- LCD
- mehr LEDs und 7-Segmentanzeigen



Einfach zu verwenden, da ein *vollständiger* Port des μC und Spannungsversorgung *direkt* verfügbar ist.
Roboter kann so auch für TI-Unterricht verwendet werden (als μC-Platine mit hübschen Unterbau).

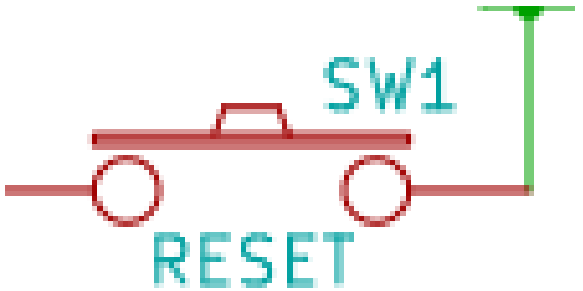
Hardwarekomponenten: Resetknopf



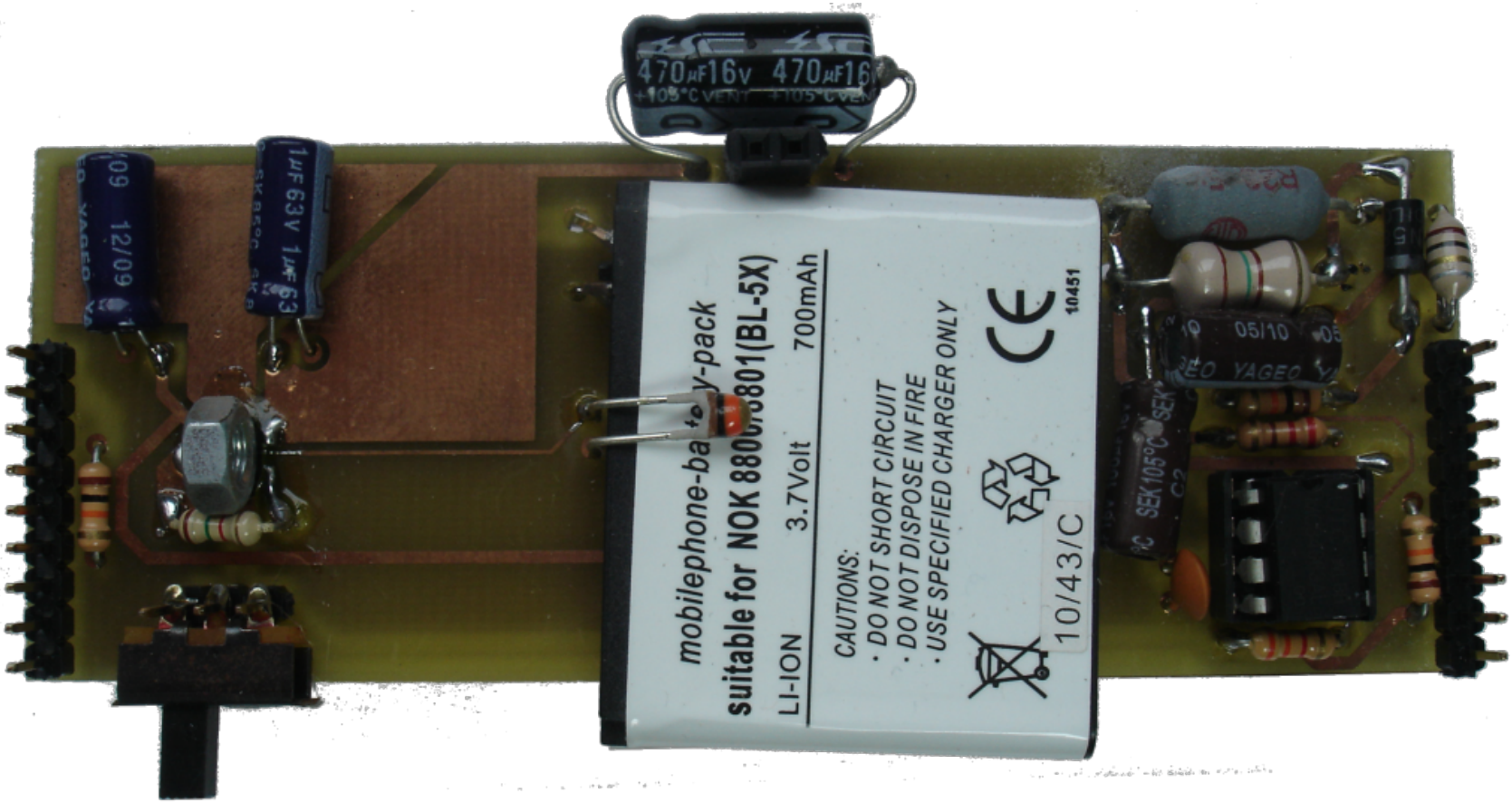


Neustart

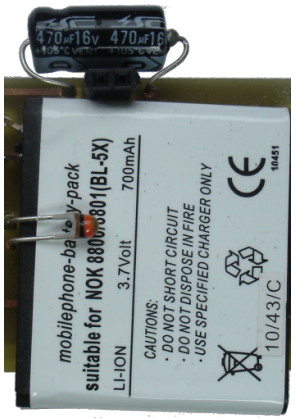
"Did you try turning it off and on again?"



Hardwarekomponenten: Spannungsversorgung (= middle Platine)

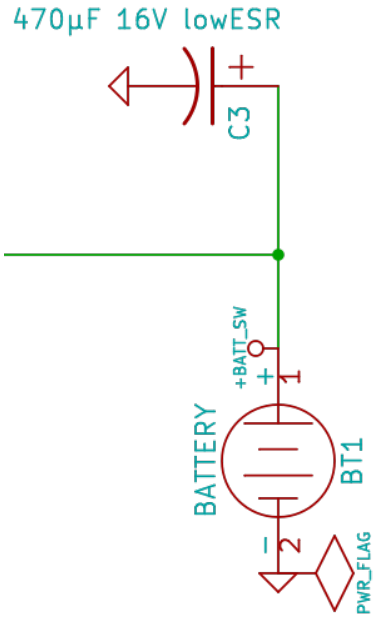


Hardwarekomponenten: Akku



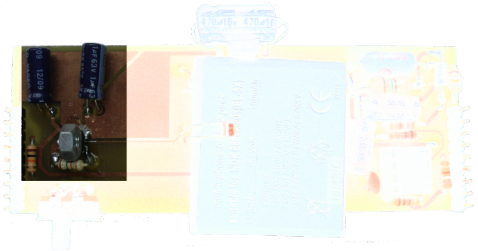
- Li-Ion 3.7V 700mAh ausreichend für Schuldoppelstunde (ffa Gebrauchsweisen)[1]
- Handyersatzakku enthält bereits Schutzschaltung!

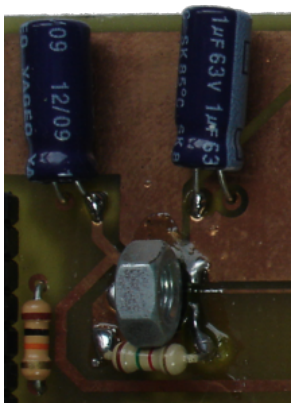
TODO: Mehr Kapazität zum gleichen Preis



[1] Sobald Programmierung und damit gleichzeitig Laden über USB erfolgt sowieso

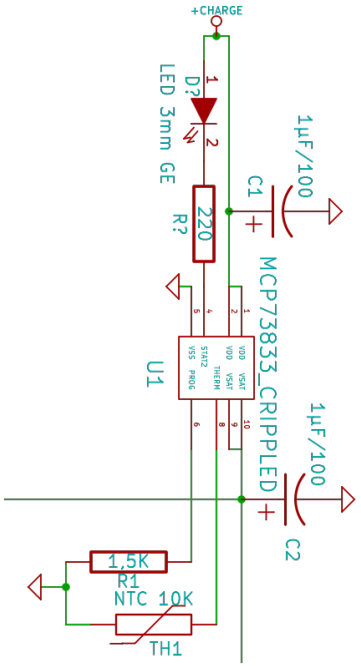
Hardwarekomponenten: Laderegler Li-Ion



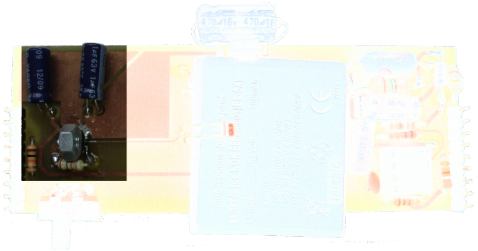


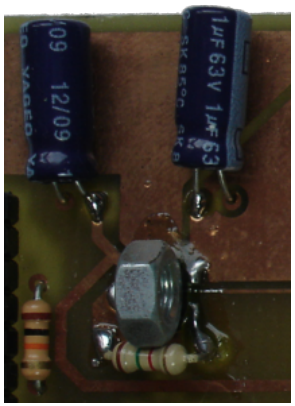
- MCP73833

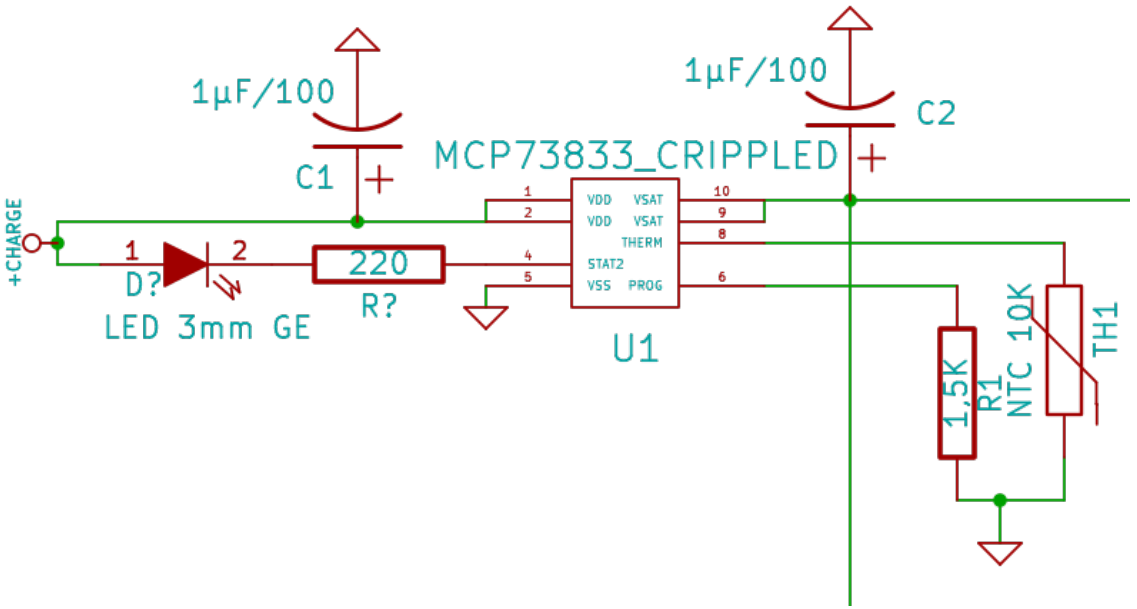
  leider MSOP package . . . kein Lötvergnügen
- Vollautomatisch
- Spannungs-, Strom- und Temperaturüberwachung



Hardwarekomponenten: Laderegler Li-Ion



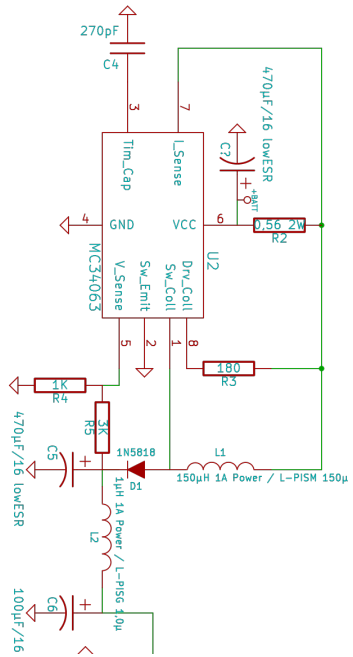




Hardwarekomponenten: Spannungswandler

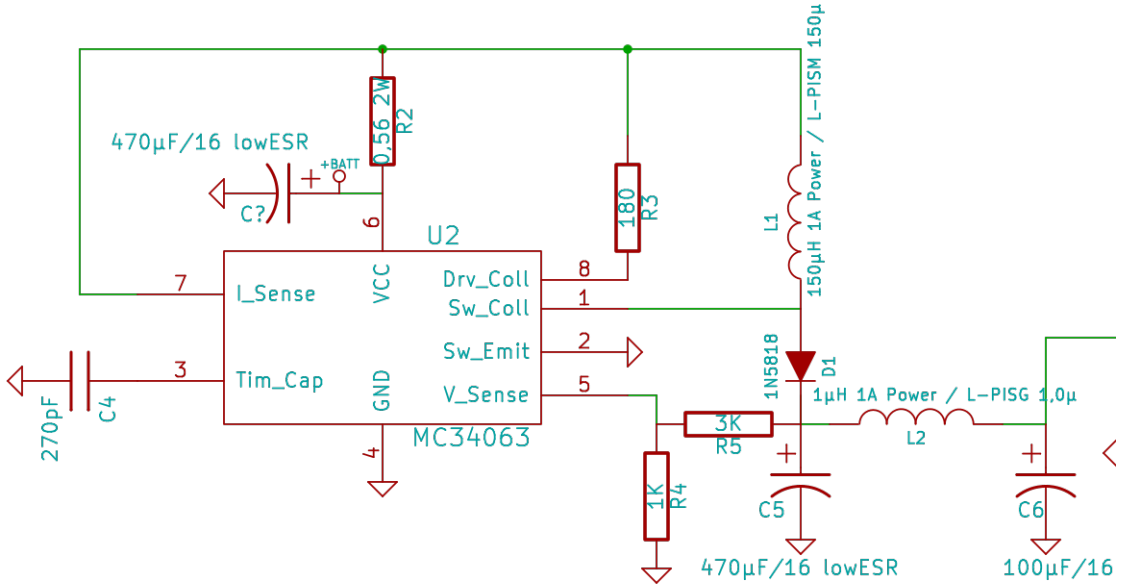
- 3.6-4.2V (Li-Ion) $\rightarrow$ 5V
- MC34063A
- “Power-Induktivitäten”[1] verwenden!

(. . . Bild ein *negativ* Beispiel)



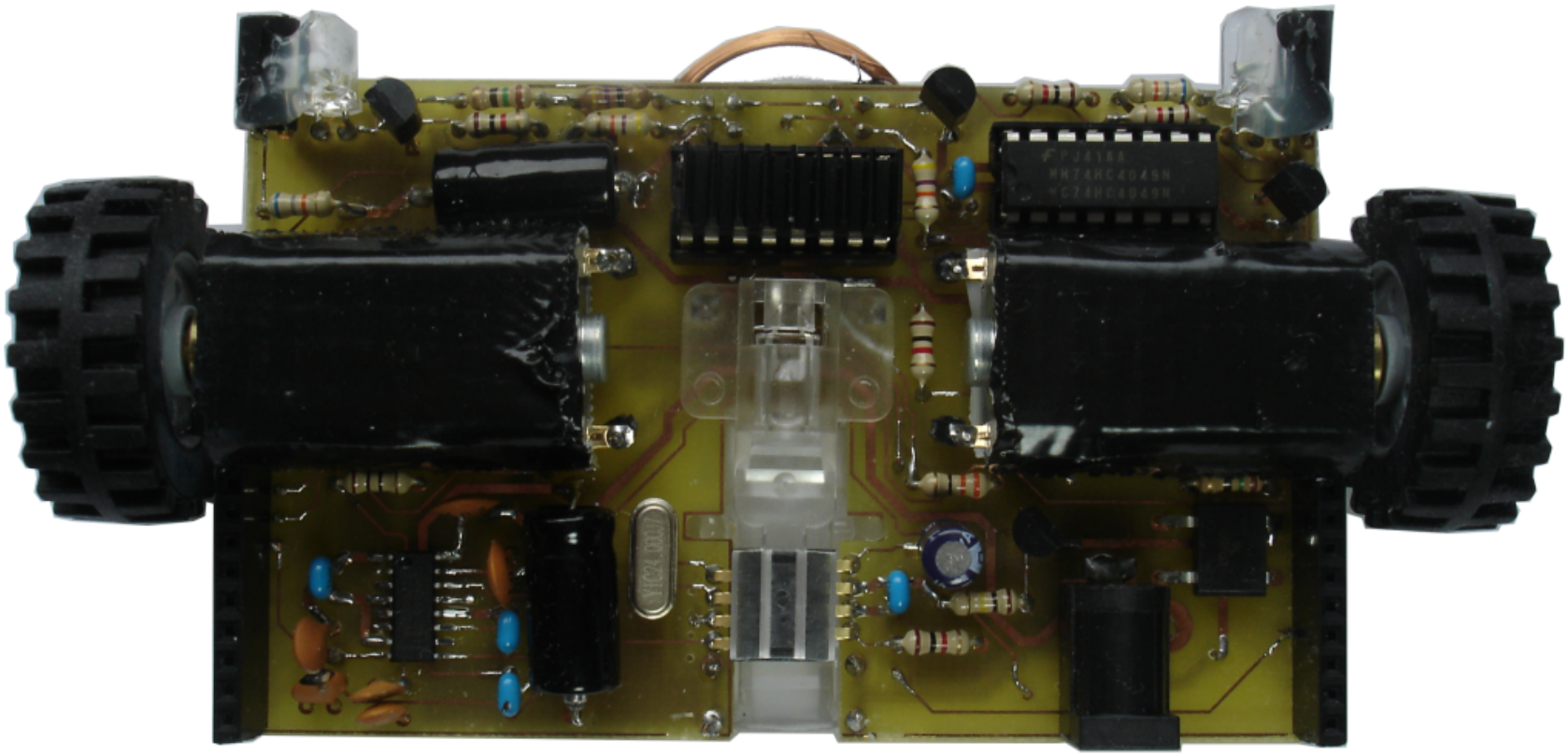
[1] Reichelt Namensgebung

Hardwarekomponenten: Spannungswandler

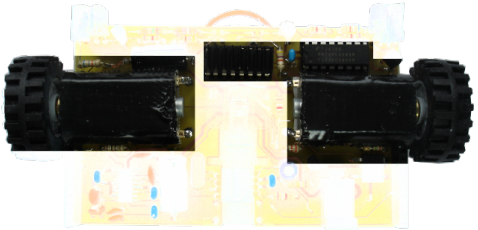


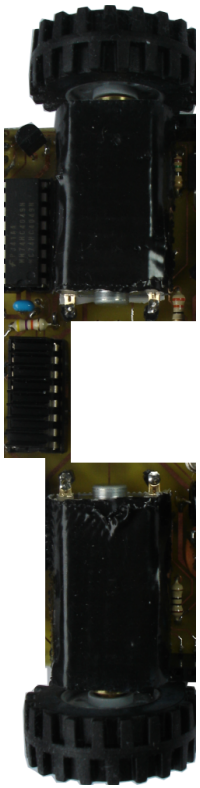
TODO: Ripple weiter reduzieren (ohne größeres L bzw. C)

Hardwarekomponenten: bottom Platine



Hardwarekomponenten: Antriebstechnik

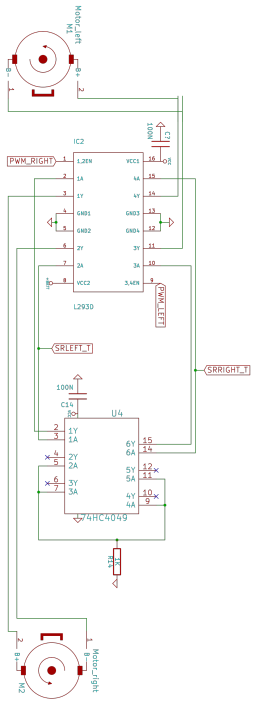




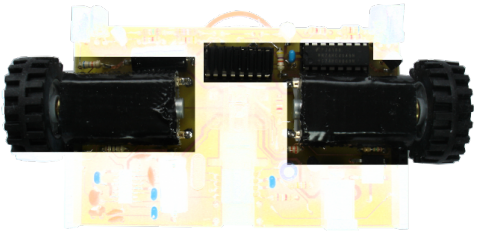
Zentraler Aspekt der Robotik: Bewegung

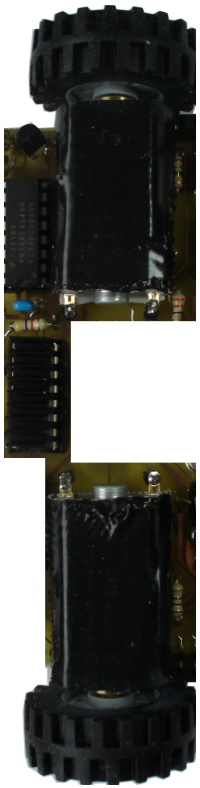
- Billigste Motoren ohne Untersetzung
- Keinerlei gezieltes Fahren ohne implementiere (PID-)Regelung

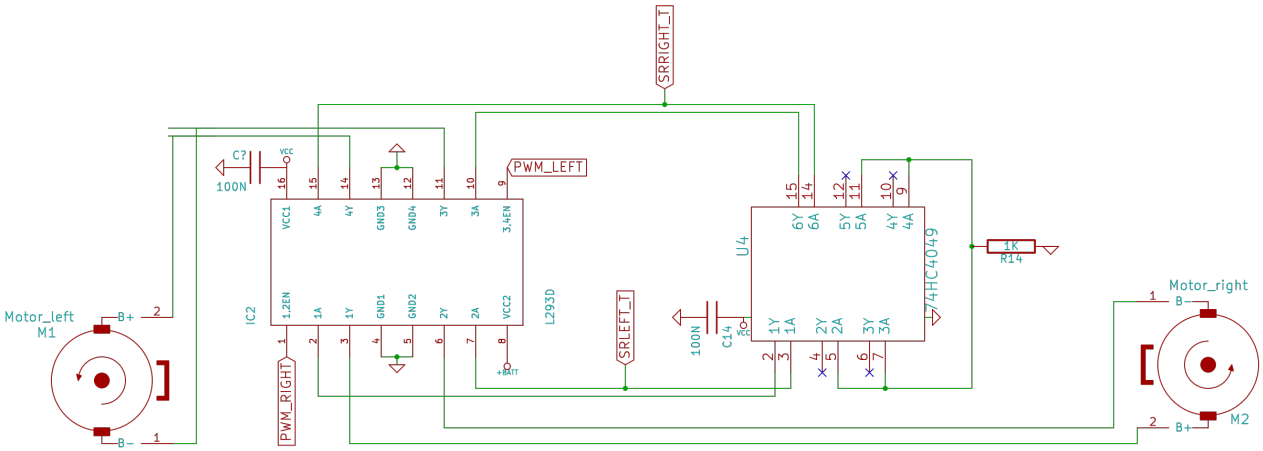
TODO: Besseres Fahrverhalten (durch bessere Regelungstechnik oder Mechanik)



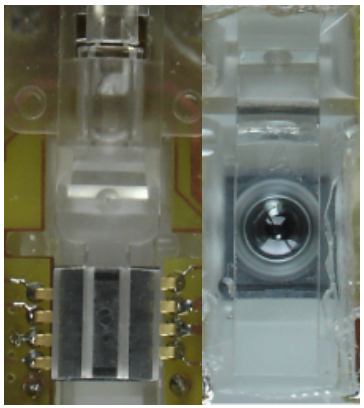
Hardwarekomponenten: Antriebstechnik





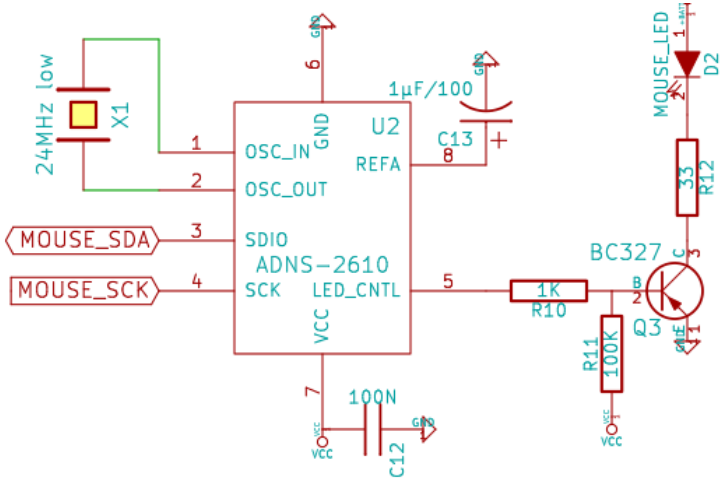


Hardwarekomponenten: Maussensor



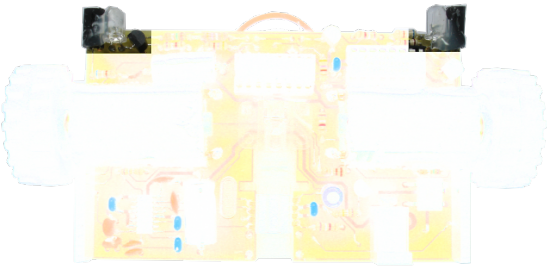
Erfassung der Bewegung durch optischen Maussensor. Zur

- Motorenregelung
- Wegmessung



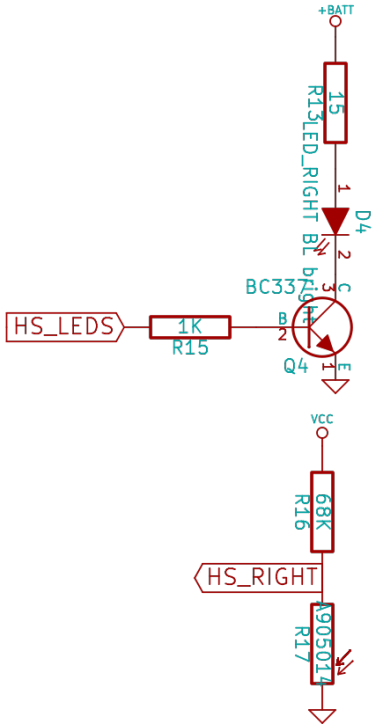
TODO: Fehlermodell, Autokalibrierung (Umrechnung in Distanz unabhängig von Untergrund)

Hardwarekomponenten: Hindernisssensoren

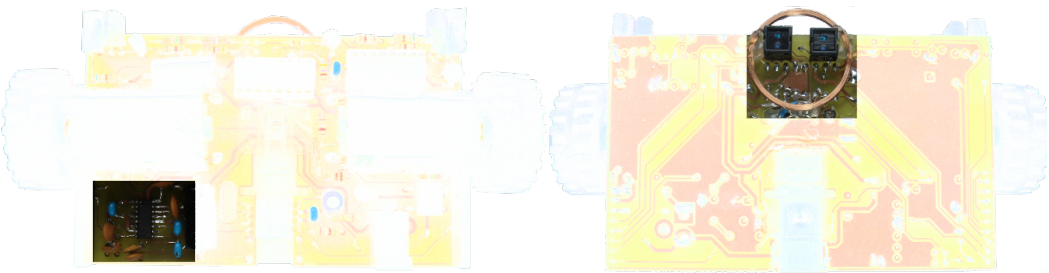


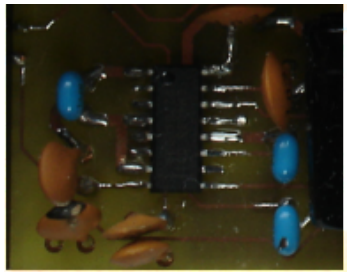
Kollisionsvermeidung durch

- gepulstes Licht (Blinkenlights!)
- LDRs (Lichtempfindliche Widerstände)

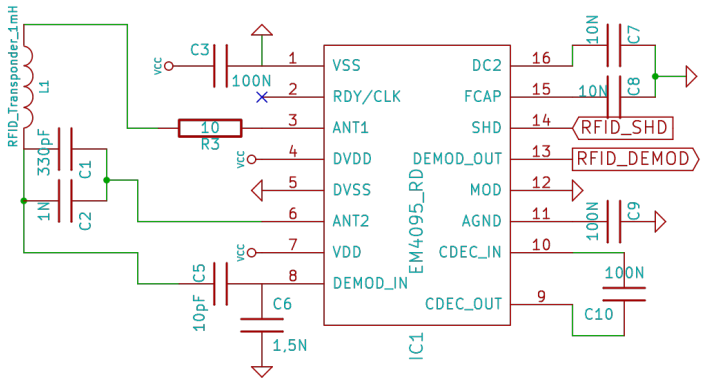


Hardwarekomponenten: RFID



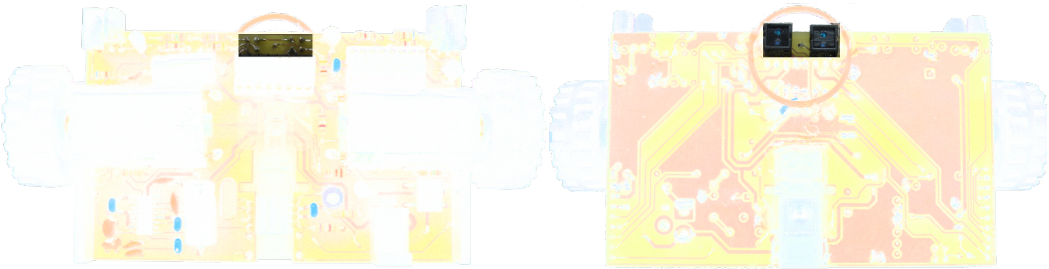


- RFID Tagleser zur
- absoluten Positionierung
- Gegenstands“erkennung”



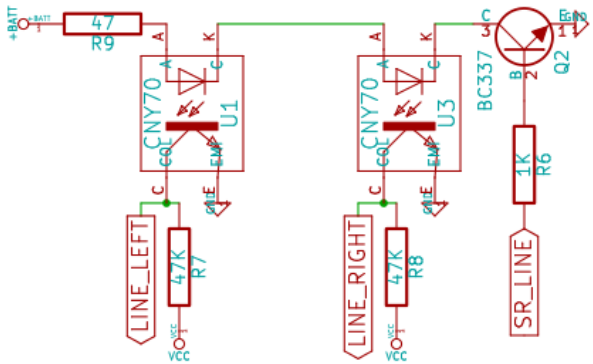
TODO: Weniger Rauschen bei laufenden Motoren.

Hardwarekomponenten: Linien- und Bodensensoren



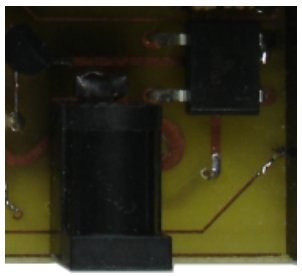
Reflexlichtsensoren zur

- Erkennung von Linien
- Detektion von Abgründen



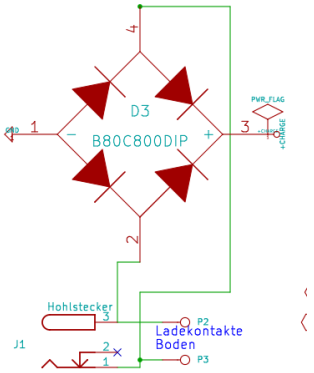
TODO: SMDifizieren

Hardwarekomponenten: Ladebuchse



Aufladen durch ungeregelte externe Spannungsquelle (7,5V).

Spannungsabfall beachten!



4 Software
- Überblick
- API Dokumentation
- Beispiele
- Bootloader

5 LiveCD

1 About us

2 Das HERD Projekt

3 Hardware

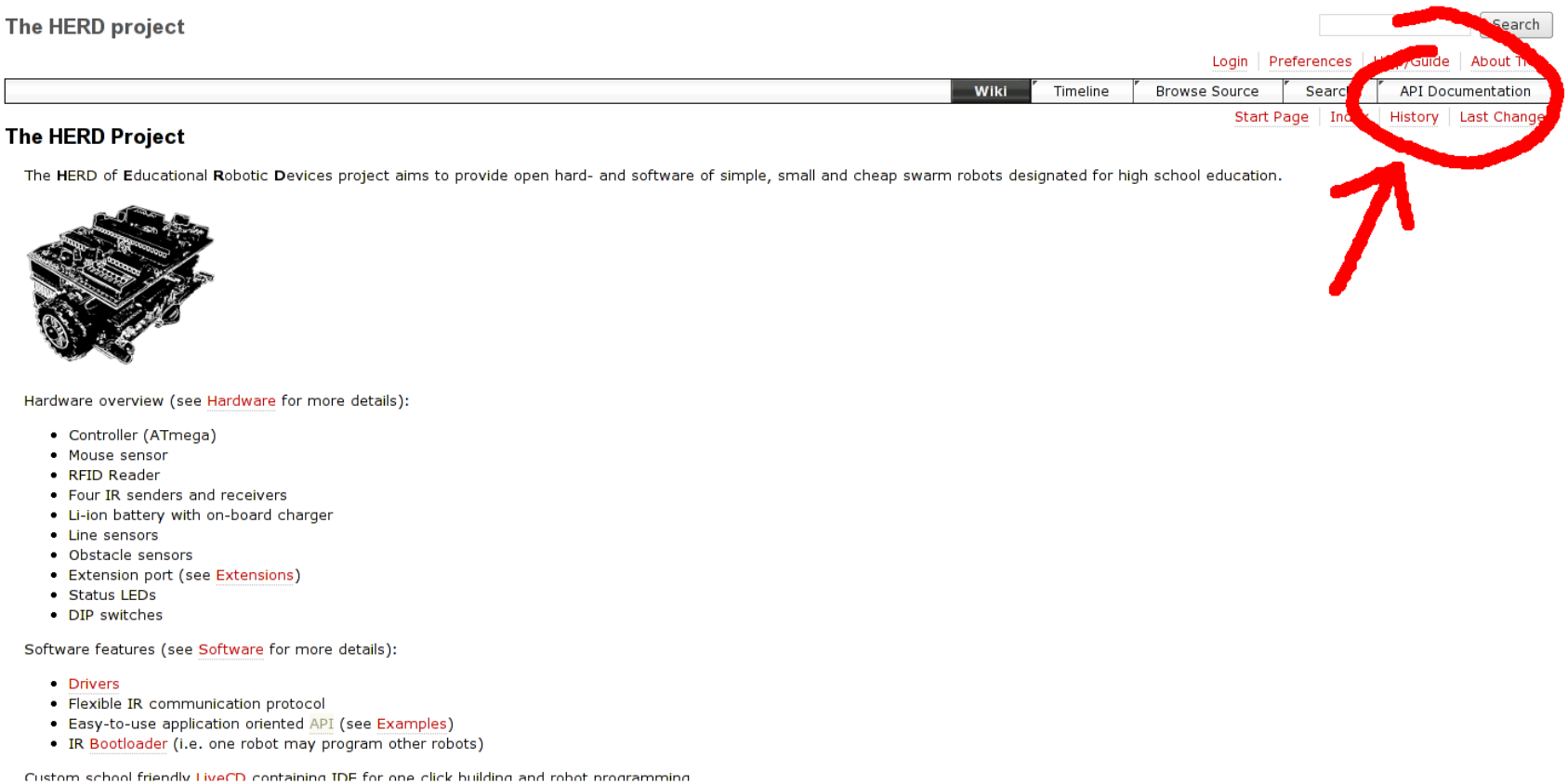
Software: Überblick

- Vollständig in C geschrieben (low- wie high-level)
- High level API (Doxygen dokumentiert) für Sensoren und Aktoren
- Kein Betriebssystem

TODO: Pointer vollständig vermeiden, (C++ highest-level API?)

API Dokumentation

http://herd-project.org/doxygen



Beispiele: example1

```
1  #include <all.h>
2
3  int
4  main(void)
5  {
6      /* initialize devices */
7      init_all();
8
9      while(1) {
10         /* turn left led on */
11         led_set(LEFT);
12         /* sleep for one second */
13         delay_ms(1000);
14         /* turn right led on */
15         led_set(RIGHT);
16         /* sleep for one second */
17         delay_ms(1000);
18     }
19
20     return 0;
21 }
```

Weitere Beispiele

http://herd-project.org/wiki/Examples

The HERD project

Examples

Example1	initialize devices and blink with LEDs
Example2.1	wait until dip switch changed then drive forward
Example2.2	like Example2.1 with examination of return values
Example3	read rfid tag (if any) and blink according to which tag was there
Example4	makes HERD robot a simple remote controlled robot by a standard TV remote

TODO: Mehr (und auch komplexere) Beispiele

Der Bootloader

Sinn und Zweck

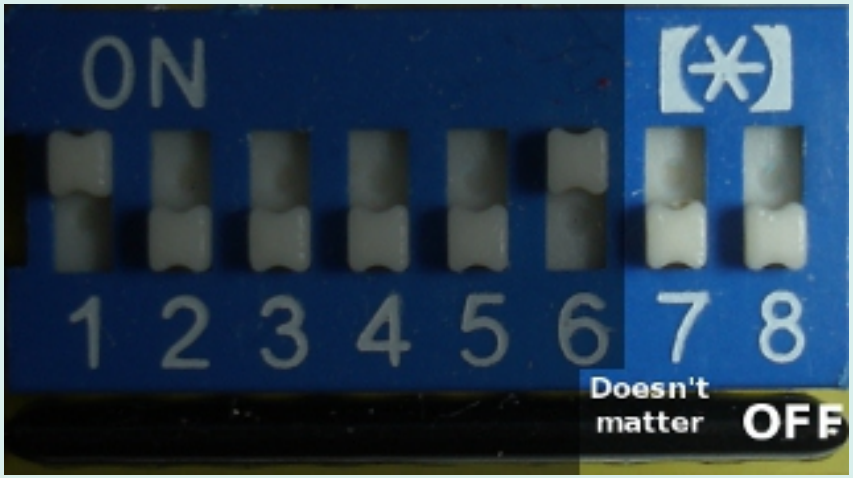
Aktuellen Code auf diesem Roboter auf beliebig viele andere Roboter duplizieren (Schwarmkonzept).

Der Bootloader

Sinn und Zweck

Aktuellen Code auf diesem Roboter auf beliebig viele andere Roboter duplizieren (Schwarmkonzept).

Bootloader **nicht** ausführen (Normalen Code ausführen)



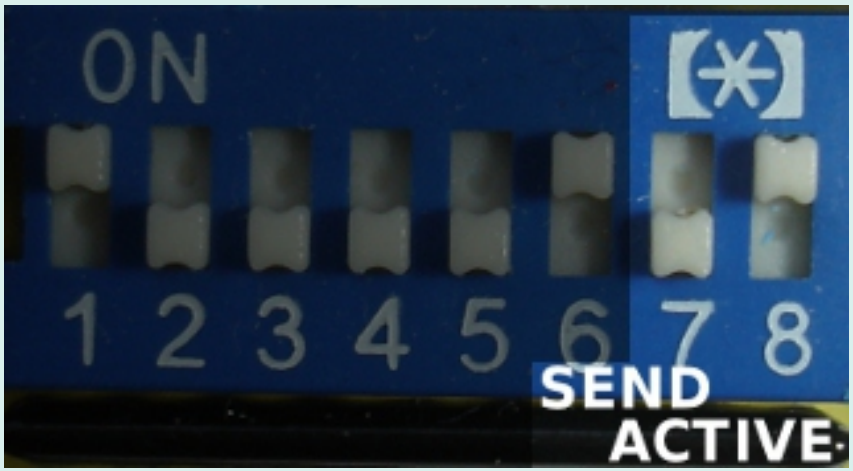
Der Bootloader

Sinn und Zweck

Aktuellen Code auf diesem Roboter auf beliebig viele andere Roboter duplizieren (Schwarmkonzept).

Bootloader ausführen:

Von diesem Roboter zu anderen (d.h. Senden)



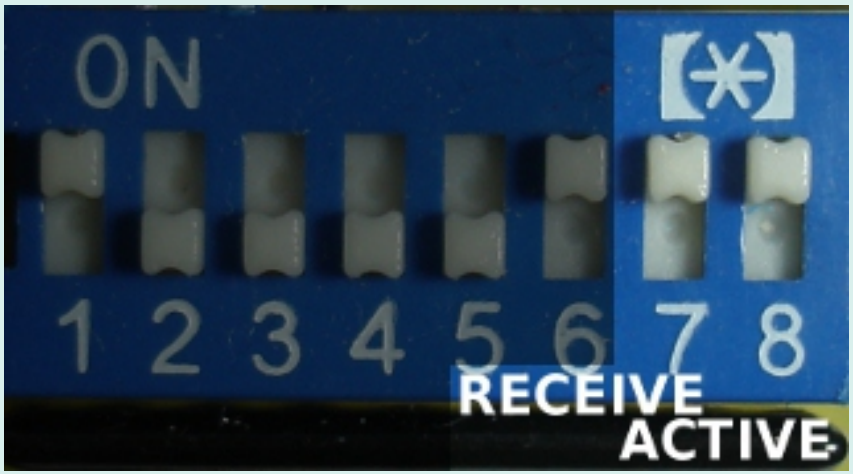
Der Bootloader

Sinn und Zweck

Aktuellen Code auf diesem Roboter auf beliebig viele andere Roboter duplizieren (Schwarmkonzept).

Bootloader ausführen:

Zu diesem Roboter von einem Anderen (d.h. Empfangen)



Empfang auf *Rückseite*

5 LiveCD

1 About us
2 Das HERD Projekt
3 Hardware
4 Software

LiveCD

enthält

- angepasste Eclipse IDE
  (inkl. kompilieren und Roboter programmieren)
- vollständigen Source Code, Dokumentation (inkl. dieser Präsentation) und Schaltpläne
- Unterstützung für diverse Programmieradapter

erleichert

- Einstieg
- Administration für Lehrkörper :)

TODO: Eclipse Integration verbessern (Doxygen, anklickbare Funktionstabelle, . . .)

6 Demo

1 About us
2 Das HERD Projekt
3 Hardware
4 Software
5 LiveCD

Einzel- und Multiroboter

Demonstration

Zur Praxis

TODO

Viel Spaß am Gerät