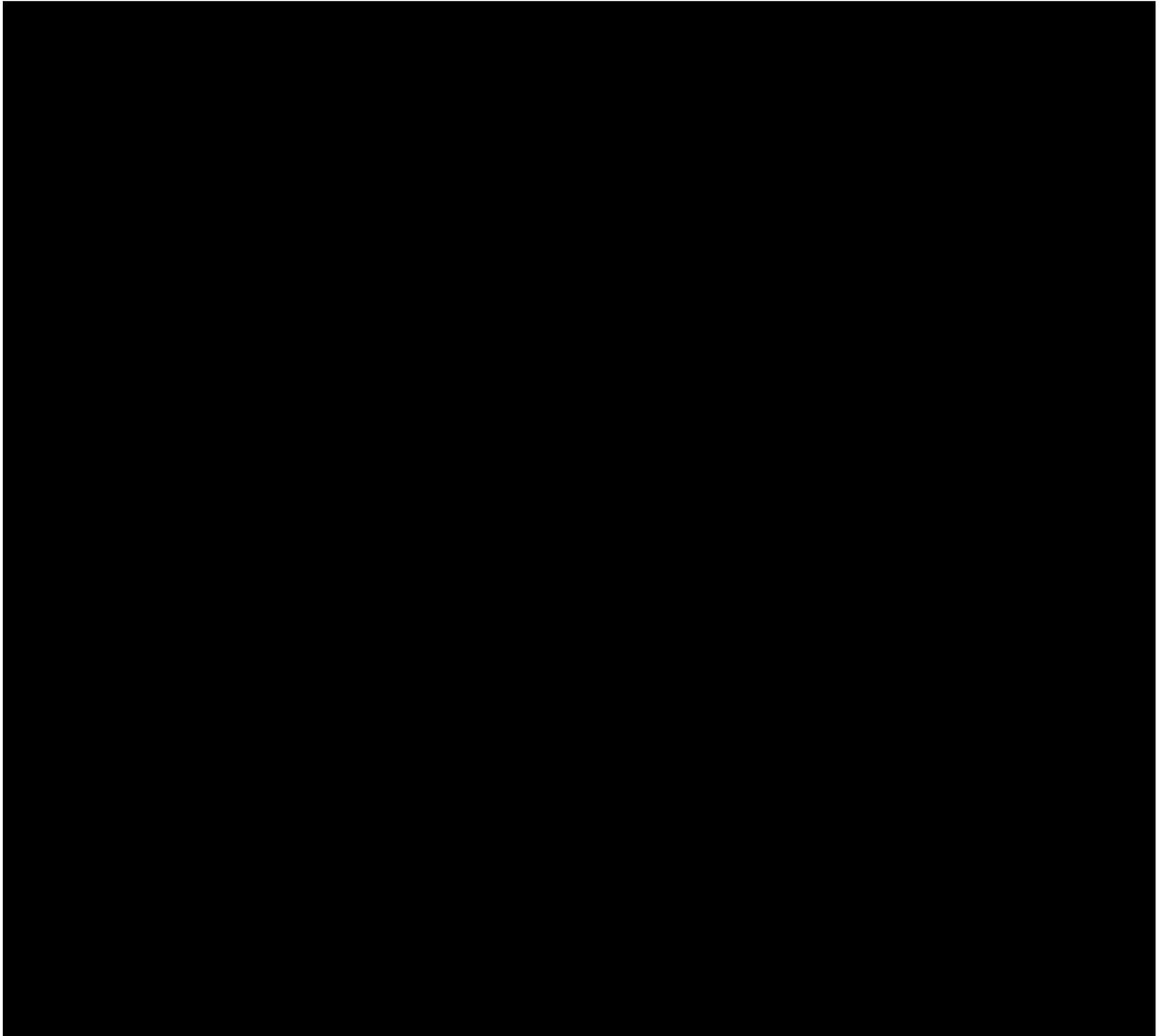


Roboterprojekt 2018

„Der Gerät“



Inhaltsverzeichnis

1. Einführung	4
2. Beschreibung des Projekts	5
2.2. Teammitglieder.....	7
3. Aufbau	9
3.1. Bauteile.....	9
3.2. IR-Tower.....	10
3.2.1.Schaltplan.....	10
3.2.2. Bestückungsliste.....	11
3.2.3. Quellcode	12
3.2.4. Timer 0	13
3.3. Wii Cam.....	14
3.3.1. Schaltplan.....	14
3.3.2. Bestückungsliste.....	15
3.3.3. Quellcode	16
3.4. Hauptplatine	17
3.4.1. Schaltplan.....	18
3.4.2. Bestückungsliste.....	19
3.4.3. Quellcode	21
3.5. Motorsteuereinheit	22
3.5.2. Bestückungsliste.....	24
3.5.3. Quellcode	25
3.5.4. Timer 2	26
3.6. Liniensensor	27
3.6.1. Schaltplan.....	27
3.6.2. Bestückungsliste.....	28
3.6.3. Quellcode	29
3.7. Schallwellensensor	30
3.7.1. Auswertung der Messung	31
3.7.2. Timer 1	32
3.7.3. Timer 3	32
3.8. Radencoder.....	33
3.9. Bumper	33
3.10. Master Mega128	34

.....	34
3.10.1. Pin Belegungen	36
3.11. ESP8266	38
3.12. LCD Display	40
4. Steuerung des Roboters	41
4.1. Funktion	42
4.2 Tastenbelegung Fernbedienung	42
5. Arduino Verbindung	43
6. System Struktur	46
6.1. Sequenzdiagramm	47
6.2. Aktivitätsdiagramm	48
.....	48
6.3. Anwendungsfalldiagramm.....	49
6.4. Strukturgramm	50
7. Teamsitzungen	52
7.1. Teamsitzung 1.....	53
7.2. Teamsitzung 2.....	54
7.3. Teamsitzung 3.....	56
7.4. Teamsitzung 4.....	57
7.5. Teamsitzung 5.....	58
8. Zeitplan.....	59
9. Gruppenfazit.....	63
10. Abbildungsverzeichnis.....	65
11. Literaturverzeichnis.....	67
12. Weitere Informationen	68

1. Einführung

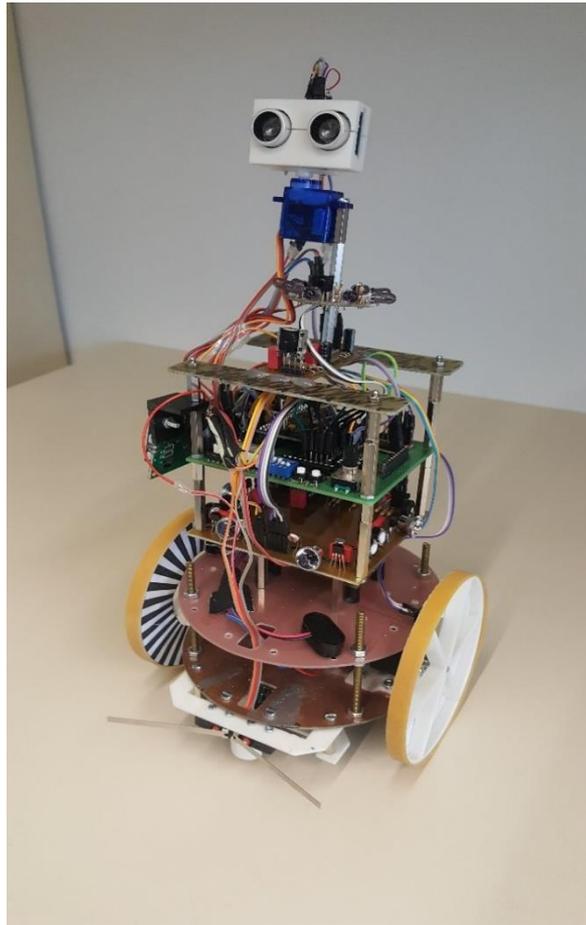


Abbildung 1:Roboter

Wir als Teil der Klasse CI51 wurden damit beauftragt mehrere Untersuchungen über die Erfassung und Verarbeitung von Umgebungsdaten zur kollisions freien Fortbewegung von autonom fahrenden Fahrzeugen vorzunehmen.

Im Hinblick auf die Automobilität der Zukunft und die in der Produktion und Industrie zukünftig zu erwartenden Veränderungen, war es uns mittels verfügbaren Sensorik des Roboters nach mehreren Möglichkeiten zu erforschen. Sowohl Wegstrecken ohne menschliches Eingreifen problemlos zurücklegen, als auch plötzlich auftretende Hindernisse auszuweichen und Daten die bei der Zurücklegung von Wegstrecken oder Objekten an eine Leitstelle zu übermitteln. Deshalb haben wir den einzigwahren „Der Gerät“ entwickelt.

2. Beschreibung des Projekts

Die Aufgabe war es, den Roboter hardwaretechnisch zu bauen sowohl softwaretechnisch zu programmieren. Die für die erfolgreiche Funktion zu erreichenden Einzelteile waren:

1. Die beiden Servos des Roboters sollen sich so steuern, dass sie unabhängig voneinander die Drehrichtung und Geschwindigkeit mittels PWM (Pulsweitenmodulation) beeinflussen können.
2. Der Roboter soll in der Lage sein, sich von seinem Ausgangspunkt in allen Richtungen zu bewegen bzw. entfernen. Daraufhin soll er sich auf der Stelle drehen und auf dem gleichen Weg zum Startpunkt zurückkehren.

Hinweis

- a. Der Radencoder lässt sich mit den Pulsen eine Wegmessung implementieren.
 - b. Die Daten bzw. die gemessenen Pulse sollen zusätzlich per WLAN zum Smartphone übertragen werden und dort als gefahrene Strecke/Winkel dargestellt werden.**
3. Auftauchende Hindernisse sollen ausgewichen werden und danach seinen vorherigen eingeschlagenen Weg wiederfinden.
 4. Der Roboter soll das Wanzen/Mottenverhalten imitieren können.
 - a. Hierzu dienen die LDR's (Fotowiderstand)
 5. Der Roboter soll von seiner Ausgangsposition aus eine Leitlinie (Schwarz markiert) folgen können. Wenn ein Hindernis sich auf der Leitlinie befindet, soll der Roboter ihn ausweichen und wieder versuchen die Leitlinie zu finden um ihr weiter zu folgen.

Hinweis

- a. Die Sensordaten des Reflexkopplers sollen zusätzlich per WLAN zum Smartphone übertragen werden**
6. Anhand der Wii Cam soll der Roboter die Position eines zweiten Roboters finden und sich ihm nähern. Im zweiten Schritt soll er den gefundenen Roboter folgen können.

Hinweis

- a. Die Daten der Wii Cam sollen zusätzlich per WLAN zum Smartphone übertragen werden**
7. Der Roboter soll mit Hilfe des Ultraschallsensors, welcher auf dem kleinen Servo auf dem Roboter befestigt ist, Informationen über Objekte in der Umgebung sammeln und kartieren (Eine Karte erstellen).

Hinweis

- a. Die Daten des Ultraschallsensors sollen zusätzlich per WLAN zum Smartphone übertragen werden**
8. Alle Funktionen sollen mit Hilfe der RC5 Fernsteuerung ausgewählt und die Fahrfunktionen mit Tastendruck gesteuert werden können.

Wichtig

- a. Ebenfalls auch über die vom ESP8266 gehostete Website in einem Menü ausgewählt und gesteuert werden können.**

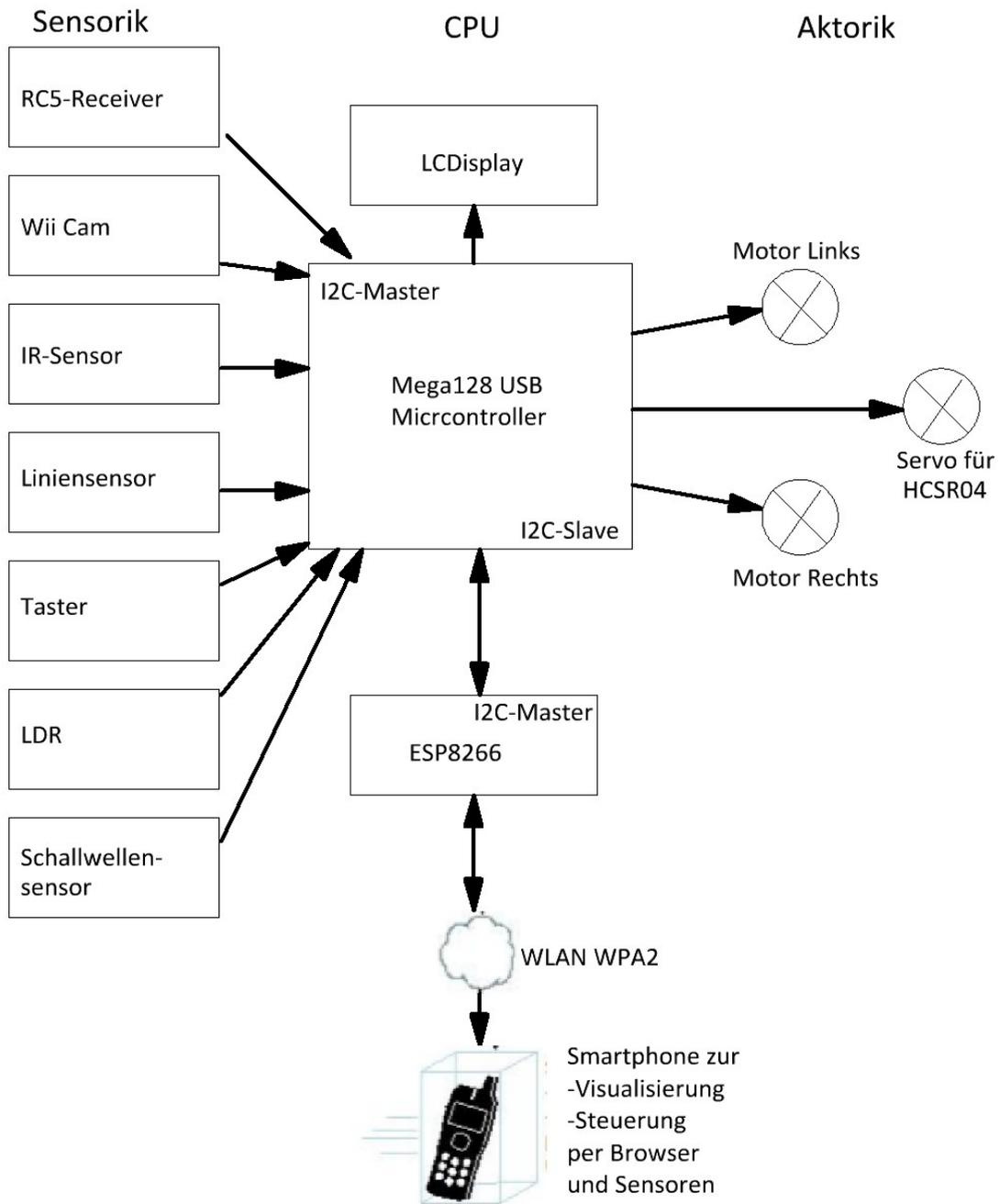


Abbildung 2: Die Aufgabendarstellung als Schaubild (Quelle: projekt roboter_3d 2017 Funkmodul_IOT.pdf)

2.2. Teammitglieder

Name: Mehmet [REDACTED]

Alter: 19

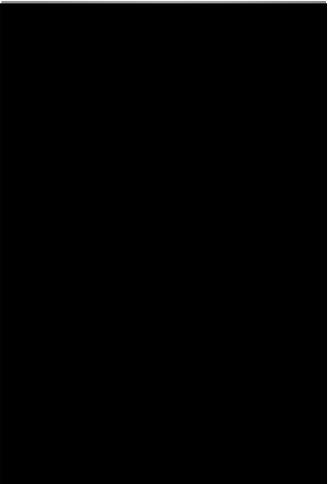
Rolle/Aufgabe: Dokumentator

- Das Dokument erstellen
- 

Name: Serdar [REDACTED]

Alter: 19

Rolle/Aufgabe: Projektleiter

- Zeitplan erstellen
 - Präsentation Erstellen
 - Intro/Outro erstellen
- 

Name: Florian Wößner

Alter: 18

Rolle/Aufgabe: Programmierer

- ESP programmieren

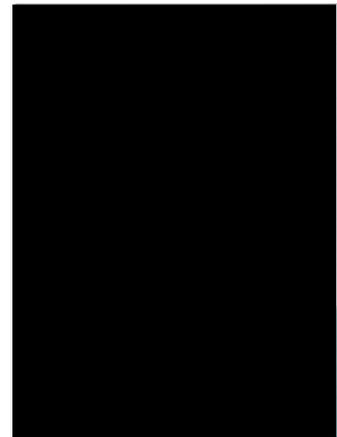


Name: Benjamin [REDACTED]

Alter: 19

Rolle/Aufgabe: Programmierer

- ATmega programmieren



3. Aufbau

Der Roboter besteht aus drei Ebenen. Die Antriebsebene, die Hauptebene (Mikrocontroller & Hauptplatine) und der Kopf. An der Vorderseite der Antriebsebene sind zwei Taster und unter der Trägerplatte ist der Liniensensor. An den Seiten befinden sich Räder für die Fortbewegung. Diese werden durch Motoren angetrieben, die an der Innenseite montiert sind. Die Antriebsebene endet mit der Trägerplatte auf den Motoren. Die Hauptplatine fängt mit dem Motortreiber an. Über dem Motortreiber befindet sich die Hauptplatine und darüber der Mikrocontroller der für die Steuerung des Roboters zuständig ist. Auf der oberen Seite der Hauptebene ist die Wii Cam befestigt und darüber befindet sich der IR-Tower. Über dem IR-Tower ist der Kopf (Schallwellensensor) an einem Servo befestigt.

3.1. Bauteile

Der Roboter besteht aus vielen Bauteilen. Es sind sowohl elektronische als auch mechanische Komponenten vorhanden:

- zwei Basisplatten rund (130mm)
- 3D-Druck Halter für Taster und Liniensensor
- zwei Räder 3D-Druck (100x8mm)
- zwei Kugellager (10mm Stahlkugel)
- zwei Motorhalter 3D-Druck
- zwei Gleichstrommotoren
- acht Distanzhülsen (Kunststoff, 8mm Länge)
- zwei Zylinderkopfschrauben (M2, 5x10 für Räder)
- 16 Zylinderkopfschrauben (M3x6)
- 4 Zylinderkopfschrauben (M3x25)
- 16 Muttern (M3)
- vier Gewindestangen (M4)
- 16 Muttern (M4)
- vier Zylinderkopfschrauben (M2x10 + vier Muttern M2)
- vier Zylinderkopfschrauben (M3x16 + vier Muttern M3)
- zwei Zylinderkopfschrauben (M3x30 + vier Muttern M3)
- zwei Distanzhülsen (Kunststoff, 10mm)

Der Roboter hat fünf Platinen die für die Steuerung dienen. Alle fünf Platinen sind mit dem Megaboard verbunden. Im folgendem werden sie die Funktionalitäten und Eigenschaften der Platinen sehen.

3.2. IR-Tower

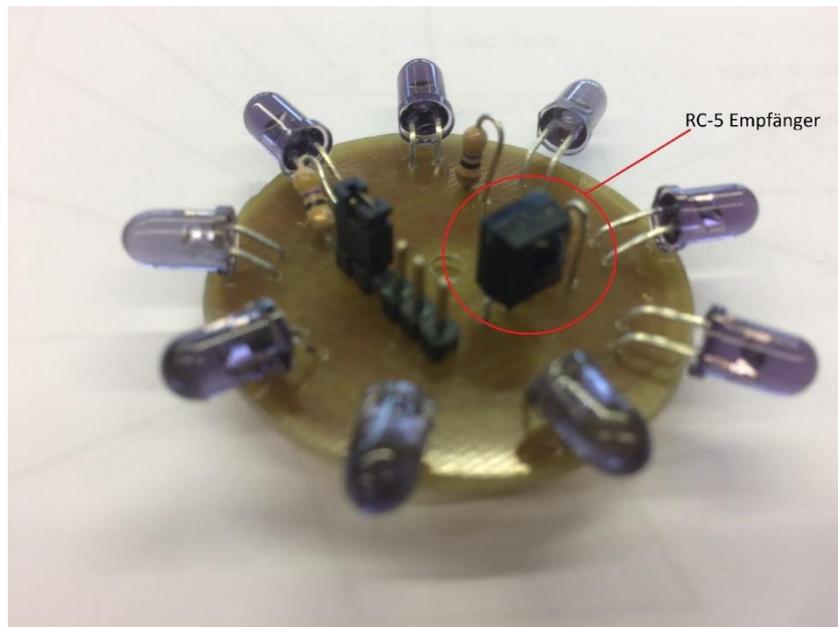


Abbildung 3: IR Tower mit RC5-Empfänger

Der IR-Tower ist hauptsächlich der Empfänger für die Befehle. Die Befehle bekommt er durch einen Tastendruck aus der Fernbedienung. Per Infrarot erkennt der RC5-Empfänger auf der Platine die Signale. Die Signale werden durch die Kabelverbindung zum Mikrocontroller geführt und dort verarbeitet. Daraufhin wird der Befehl ausgeführt.

3.2.1. Schaltplan

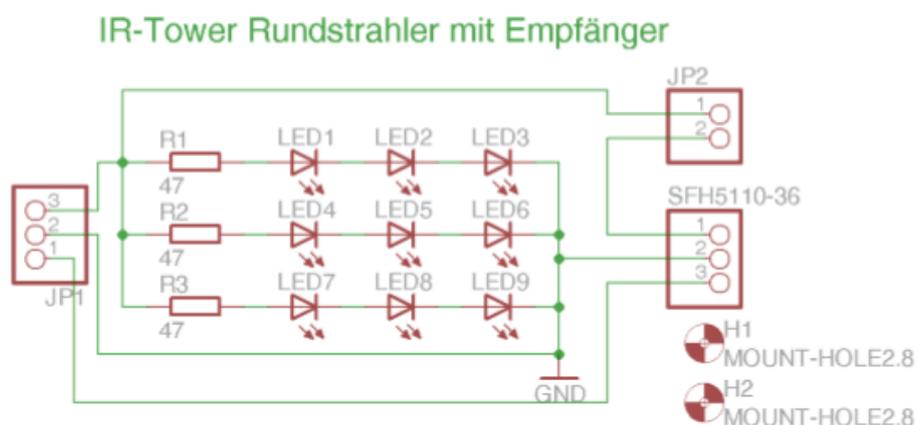


Abbildung 4: Schaltplan des IR-Towers

Auf jeder der drei Leitungen befindet sich ein Widerstand und drei LEDs. Sie sind einzeln in Reihe geschaltet und die wiederum in parallel geschaltet. Die Platine hat zwei Stecker, jeweils 2 und 3 polig. JP1 ist der 3 polige Stecker und JP2 der 2 polige Stecker. Der RC-5 Encoder ist im Schaltplan als SFH5110-36 (Modellnummer RC5) gekennzeichnet.

3.2.2. Bestückungsliste

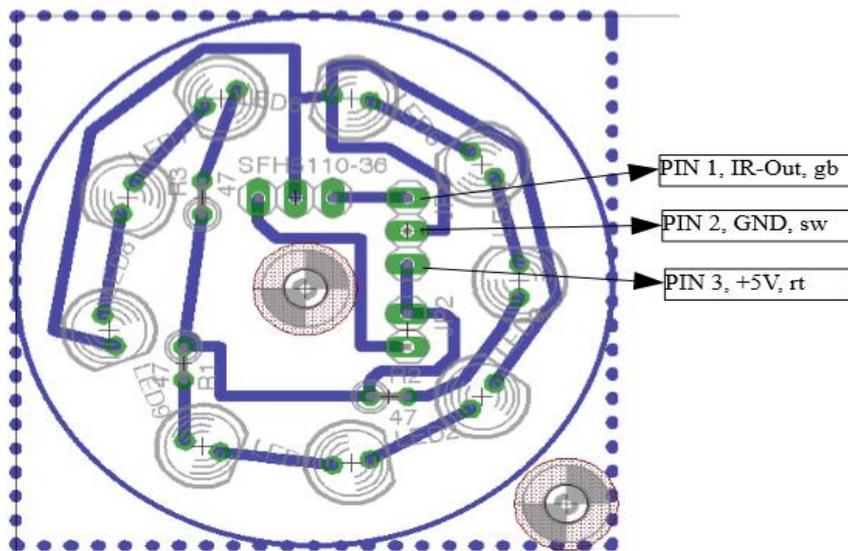


Abbildung 5: Bestückungsliste des IR-Towers (Quelleprojekt roboter_3d 2017 Funkmodul_IOT.pdf:)

Auf dem 3-poligen Stecker gilt der Pin-1 für die Übertragung der Signale vom RC-5 Encoder. Pin-2 und Pin-3 umfassen die Elektronik. Pin-2 ist an Masse verbunden und Pin 3 an die Versorgungsspannung (5 Volt). An der äußeren Seite der Platine sind neun Infrarot-LED's.

Stückliste:

- 1 x Fernsteuerungsempfänger (RC-5)
- 9 x Infrarot-LED (LD271)
- 3 x Widerstand (47 Ohm, gb-vt-sw)
- 1 x Jumper (für JP2)
- 2 polige Stecker
- 3 polige Stecker

3.2.3. Quellcode

```
//-----RC5-----

#define RC5TIME      1.778e-3      // 1.778msec
#define XTAL         16.0E6
#define PULSE_MIN   (unsigned char)(XTAL / 512 * RC5TIME * 0.2 + 0.5)
#define PULSE_1_2   (unsigned char)(XTAL / 512 * RC5TIME * 0.8 + 0.5)
#define PULSE_MAX   (unsigned char)(XTAL / 512 * RC5TIME * 1.2 + 0.5)
#define rc5          PINA.0
```

Abbildung 6: Makros RC5

Am Programm Abschnitt sehen wir, dass der RC5-Encoder auf dem PINA.0 definiert ist. Die Signale kommen von PIN 1(Platine) und gelangen weiter zu PINA.0. Das Programm entnimmt die Signale aus PINA.0 und verarbeitet sie.

```
int rc5_recieve(unsigned char *ucToggle, unsigned char *ucAddress, unsigned char *ucData){
    unsigned int i;

    #asm("cli")
    i = uiRc5Data;           // read two bytes from interrupt !
    uiRc5Data = 0;
    #asm("sei")
    if( i ){
        *ucToggle = i >> 11 & 1;
        *ucAddress = i >> 6 & 0x1F;
        *ucData = (i & 0x3F) | (~i >> 7 & 0x40);
        return i;
    }else
        return 0;
}
}

```

Abbildung 7: RC5 Code

```
if(rc5_recieve(&ucToggle, &ucAddress, &ucData)){
    /*#ifndef DEBUG
    rc5_display();
    #endif*/
    itoa(ucData,cState); //RC5 Befehl in Variable
    iState = atoi(cState);

    switch (ucData){
        case 1:
            PORTE.0 = !PORTE.0;
            delay_ms(100);
            break;
    }
}

```

Abbildung 8: RC5 Code

In Abbildung 1 wird der Befehl (Command) vom RC5 in eine Variable gespeichert. Die Variable wird weiter an die RC5 Funktion in Abbildung 2 weitergegeben. Hier wird der ausgewählte State in die Variable „iState“ eingespeichert. Im nächsten Verfahren wird das gewünschte State aufgerufen.

3.2.4. Timer 0

Hier haben sie einen Einblick über den Timer 0. Dies ist der Timer des RC-5 Empfängers. Informationen über den Timer selbst und über die Initialisierung sind im folgendem aufgelistet.

Timer 0:

- 8-Bit
- Clock source: System Clock
- Clock value: 62,500 kHz
- Mode: Normal top=0xFF
- Timer Period: 4,096 ms
- $TCCR0 = (0 \ll WGM00) | (0 \ll COM01) | (0 \ll COM00) | (0 \ll WGM01) | (1 \ll CS02) | (1 \ll CS01) | (0 \ll CS00)$
- $TCNT0 = 0x00$

- $TCCR0: (1 \ll CS02) | (1 \ll CS01) | (0 \ll CS00) = 110 = /256 = 16 \text{ Mhz}/256 = \mathbf{62.500ms}$

Wenn der Timer gestartet wird, wird auch der Prescaler neu initialisiert!

- $TCNT0 = 254$
- Prescaler : **0,282 ms**
 - **angepasst an RC-5 Code Übertragung**

3.3. Wii Cam

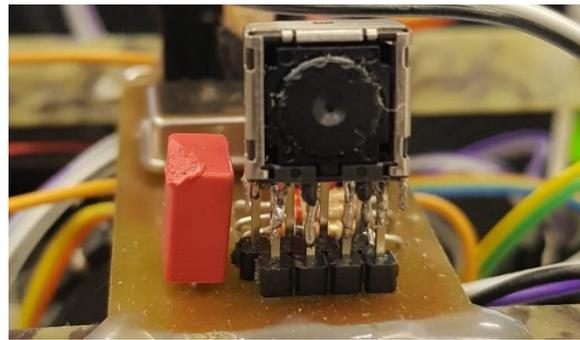


Abbildung 9: Wii Cam Platine

Die Funktionalität der Wii Cam beschränkt sich darauf die Position eines zweiten Roboters zu finden. Jeder Roboter hat eine Art Leuchtturm mit neun Infrarot LEDs. Durch die Berechnung der Entfernung mithilfe der Wii Cam soll der Roboter den zweiten Roboter bei der Fortbewegung folgen können.

Die Wii Cam enthält eine Infrarotkamera mit 1024 x 768 Pixeln und eine Hardware die bis zu vier Objekte gleichzeitig erkennen kann. Die Objekte werden von einem Signalprozessor erfasst. Die Cam erfasst sie indem sie nach IR-Objekten im Sichtfeld sucht.

3.3.1. Schaltplan

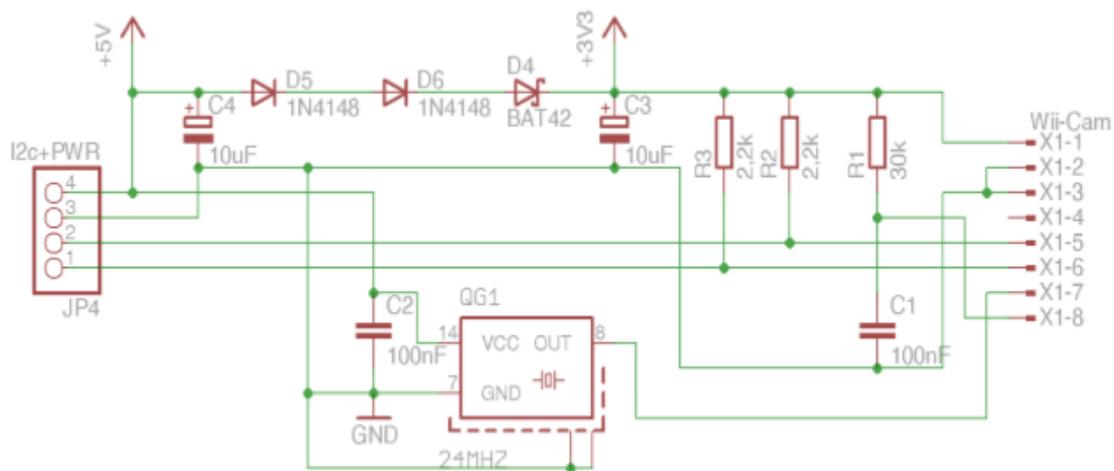


Abbildung 10: Schaltplan der Wii Cam Platine (Quelle: projekt roboter_3d 2017 Funkmodul_IOT.pdf)

Die Wii Cam ist in der Lage vier Infrarot-Signale (Objekte) zu erkennen. Die Position stellt sie durch X/Y Koordinaten fest. Der JP4 ist der I2C-Bus, womit man Kommunikation mit der Wii Cam aufbauen kann. Oberhalb der Schaltung sind drei Dioden in Reihe geschaltet. Die D5 und D6 sind Silizium Dioden und die D4 ist eine Schottkydiode. Die Dioden drosseln die Spannungen von 5 Volt zu ca. 3,3 Volt und die Kondensatoren dazwischen glätten die Spannung. Die QG1 ist ein Generator und generiert ein Takt mit 24 MHz.

3.3.2. Bestückungsliste

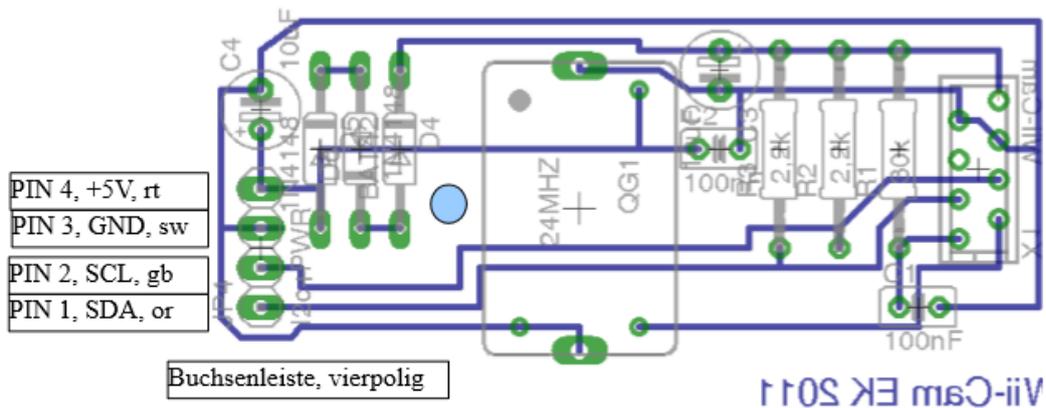


Abbildung 11: Bestückungsliste der Wii Cam (Quelle: projekt roboter_3d 2017 Funkmodul_IOT.pdf)

Die JP4 Bezeichnung entspricht der 4 poligen Buchsen am hinteren Bereich der Platine. PIN 4 ist an die Versorgungsspannung und PIN 3 an Masse verbunden. PIN 1 & PIN2 sind für die Signalübertragung zuständig. Die D5, D6 und D7 Dioden sind vor den Buchsen angelötet in der Mitte liegt der Generator. Hinter der Wii Cam sind die drei Widerstände und zwei Kondensatoren.

Stückliste:

- 1 x Wii Cam aus Nintendo Wii Controller
- 1 x Quartzgenerator (24MHz, QG1)
- 2 x C1, C2 Kondensator (100nF)
- 2 x C3, C4 Elektrolytkondensator (10uF, gepolt)
- 2 x D5, D6 Diode (1N4148)
- 1 x D4 Schottkydiode (BAT42)
- 1 x R1 Widerstand 30kΩ (or-sw-or)
- 2 x R2, R3 Widerstand 2,2kΩ (rt-rt-rt)
- 4 polige Buchse

3.3.3. Quellcode

```
void WiiCamMain(){
  readData();
  convertdata();
  lcd_clear();

  //Anzeige der Blobs als X/Y-Wertepaare
  //erste Zeile: Blob 1
  if(uiX==1023||uiY==1023){
    lcd_putsf("No Object!");
    bObject=0;
  }else{
    //konvertiert int-Ausgabe des Empfängers in cArray
    itoa( uiX-512, ucWert);
    lcd_puts(ucWert);
    itoa( uiY-374, ucWert);
    lcd_gotoxy(8, 0);
    lcd_puts(ucWert);
    itoa( ucS, ucWert);
    lcd_gotoxy(16, 0);
    lcd_puts(ucWert);
    bObject=1;
  }
  delay_ms(30);
}
```

1

2

Abbildung 12: Wii Cam Code

In Abbildung 1 vergleicht das Programm die Werte der X und Y Variablen. Wenn die beiden Werte bei dem Vergleich bei 1023 (Schnittwert = konstant) liegen, bedeutet, dass kein Objekt vor dem Roboter steht. Doch wenn die Werte nicht 1023 ergeben, werden die X, Y und S (Größenwert des Objekts) Werte konvertiert um diese auf dem LCD Display auszugeben (Abbildung 2)

3.4. Hauptplatine



Abbildung 13: Hauptplatine

An der Hauptplatine befinden sich zwei wichtige Komponenten die für das Motten-/Wanzenverhalten imitierend zuständig sind - die LDRs.

Ein LDR (Fotowiderstand) ist ein lichtempfindliches elektrisches Bauelement.

Die IR-LED auf der Hauptplatine sendet ein IR-Licht aus. Wenn das Licht gegen ein Objekt stößt prallt es zurück. Die IR-Empfänger erkennen das Licht und wissen so, dass ein Objekt vor dem Roboter ist. Die IR-Sensoren werden aktiv und können von allen Seiten Objekte erkennen. Somit kann der Roboter einen Zusammenstoß mit dem Objekt verhindern.

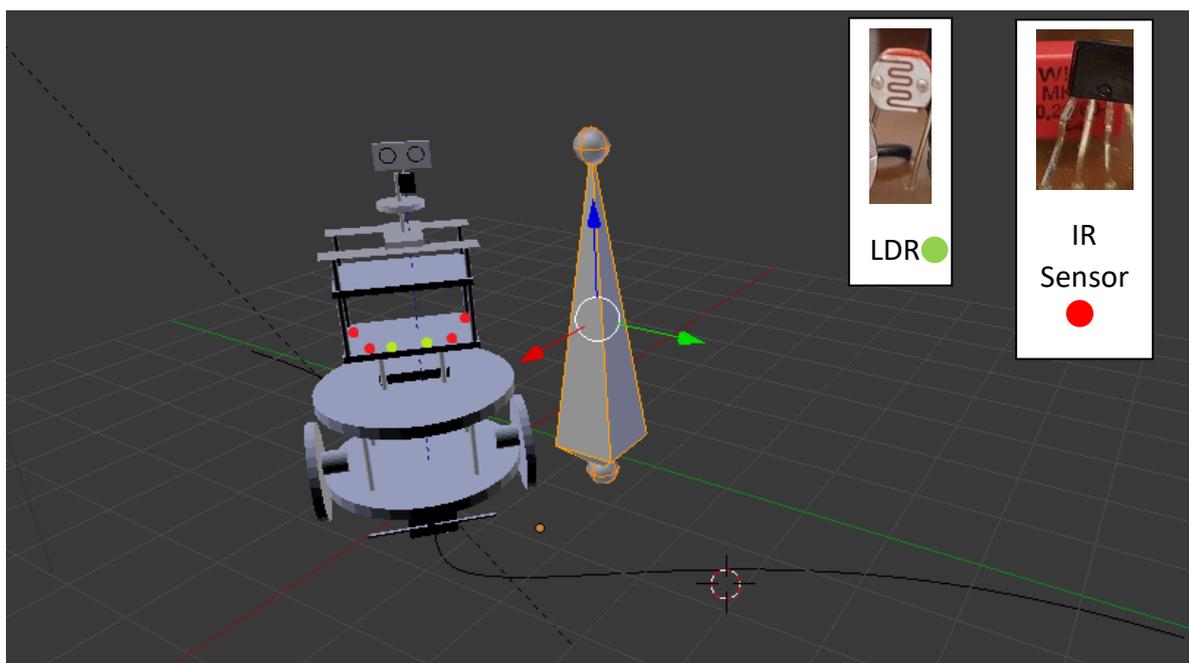


Abbildung 14: Animation

3.4.1. Schaltplan

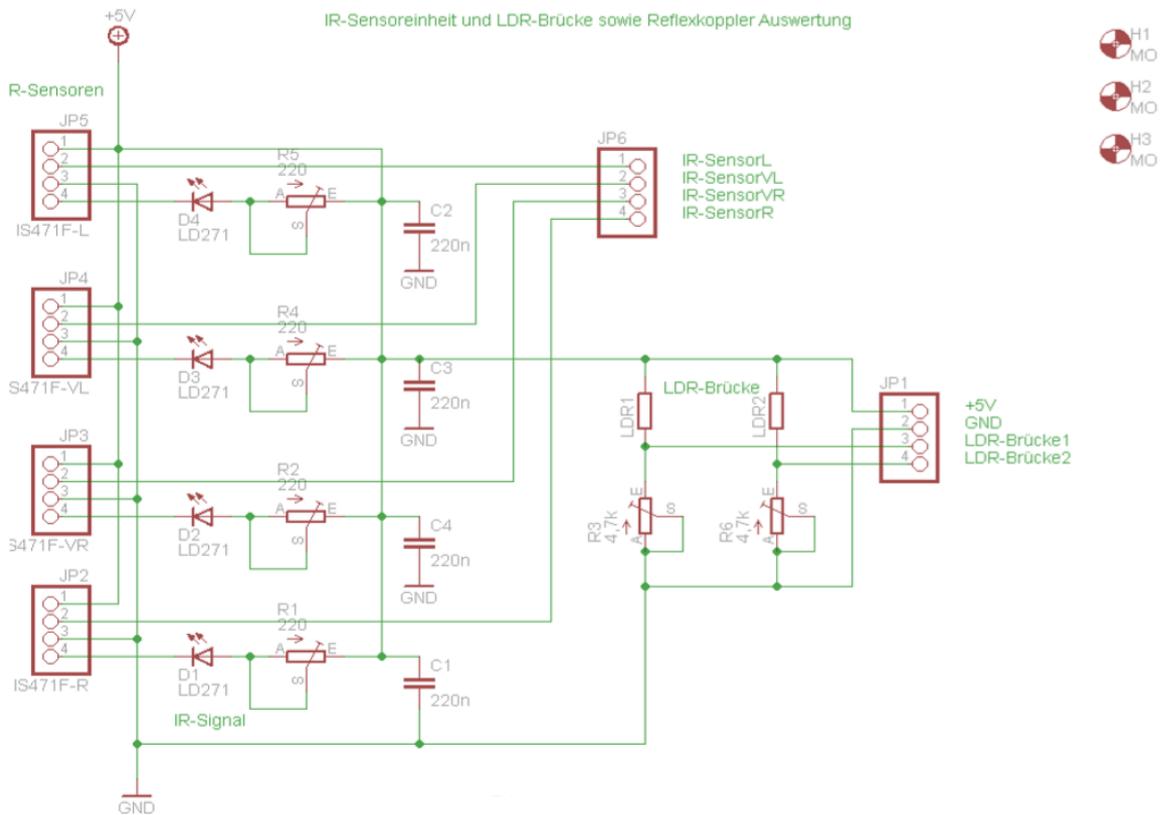


Abbildung 15: Schaltplan der Hauptplatine (Quelle: projekt roboter_3d 2017 Funkmodul_IOT.pdf)

Auf der rechten Seite der Schaltung befinden sich die zwei wichtigsten Komponenten der Hauptplatine. JP6 ist die 4 polige Buchse an der die IR-Sensoren verbunden sind und JP1 ist die 4 polige Buchse für die LDRs. An den vier IR-Sensoren ist jeweils ein Kondensator, woran eine Masse angeschlossen ist. Kondensatoren leiten nur bei Hochfrequenzen und deshalb ist diese Schaltung dazu da Hochfrequenzen abzufangen. An den IR-Sensoren, als auch an der LDR Brücke ist an jedem Sensor ein Potentiometer. Die Diode an der Platine sendet ein IR-Licht und wenn es an einem Objekt stößt, prallt sie wieder zurück.

3.4.2. Bestückungsliste

Das Wichtigste an der IR-Sensoreinheit ist die Pinbelegung der Verbindungsstecker und die zwei Sensoren auf der Platine.

Auf JP6 sind die IR-Sensoren angeschlossen:

1. PIN1 mit dem linken Sensor
2. PIN2 vorne mit dem linken
3. PIN3 vorne mit dem rechten
4. PIN4 mit dem rechten

Auf JP1 sind die LDRs angeschlossen:

1. PIN1 mit dem linken
2. PIN2 mit dem rechten

Auf JP7 ist der rechte Servo angeschlossen:

1. PIN1 mit der Versorgungsspannung
2. PIN2 mit Transistor Emitter
3. PIN3 mit IR-LED

Auf JP8 ist der linke Servo angeschlossen:

1. PIN1 mit der Versorgungsspannung
2. PIN2 mit Transistor Emitter
3. PIN3 mit IR-LED

Auf JP9 kommen die Encodersignale der Servos

1. PIN1 mit Encoder links
2. PIN2 mit Encoder rechts

Stückliste:

<ul style="list-style-type: none">➤ 2 x LDR07 (Lichtempfindliche Widerstand)➤ 4 x IS471F (Hindernisdetektor)➤ 4 x D1-D4, IR-LED LD271 + Reflektor➤ 1 x IC1, 74HC14 (Schmitt- Trigger)➤ 1 x IC2, LM358 (Operationsverstärker)➤ 2 x R11, R12, Widerstand 10k Ω (br-sw-or)➤ 2 x R13, R14, Widerstand 1k Ω (br-sw-rt)➤ 2 x R9, R10, Widerstand 180Ω (br,gr,br)➤ 4 x R1, R2, R4, R5 Trimpoti 220Ω➤ 2 x R7, R8, Trimpoti 10kΩ	<ul style="list-style-type: none">➤ 2 x R3, R6, Trimpoti 4,7kΩ➤ 4 x C1-C4, Kondensator 220nF➤ 1 x C5, Kondensator 100nF➤ 1 x 4 polige Buchse➤ 2 x 3 polige Buchse➤ 4 x 2 polige Buchse➤ 1 x Platine IR-Sensor➤ 2 x 14 + 8 polige IC-Fassung➤ 2 x LED1, LED2, LED 3mm, rot
--	---

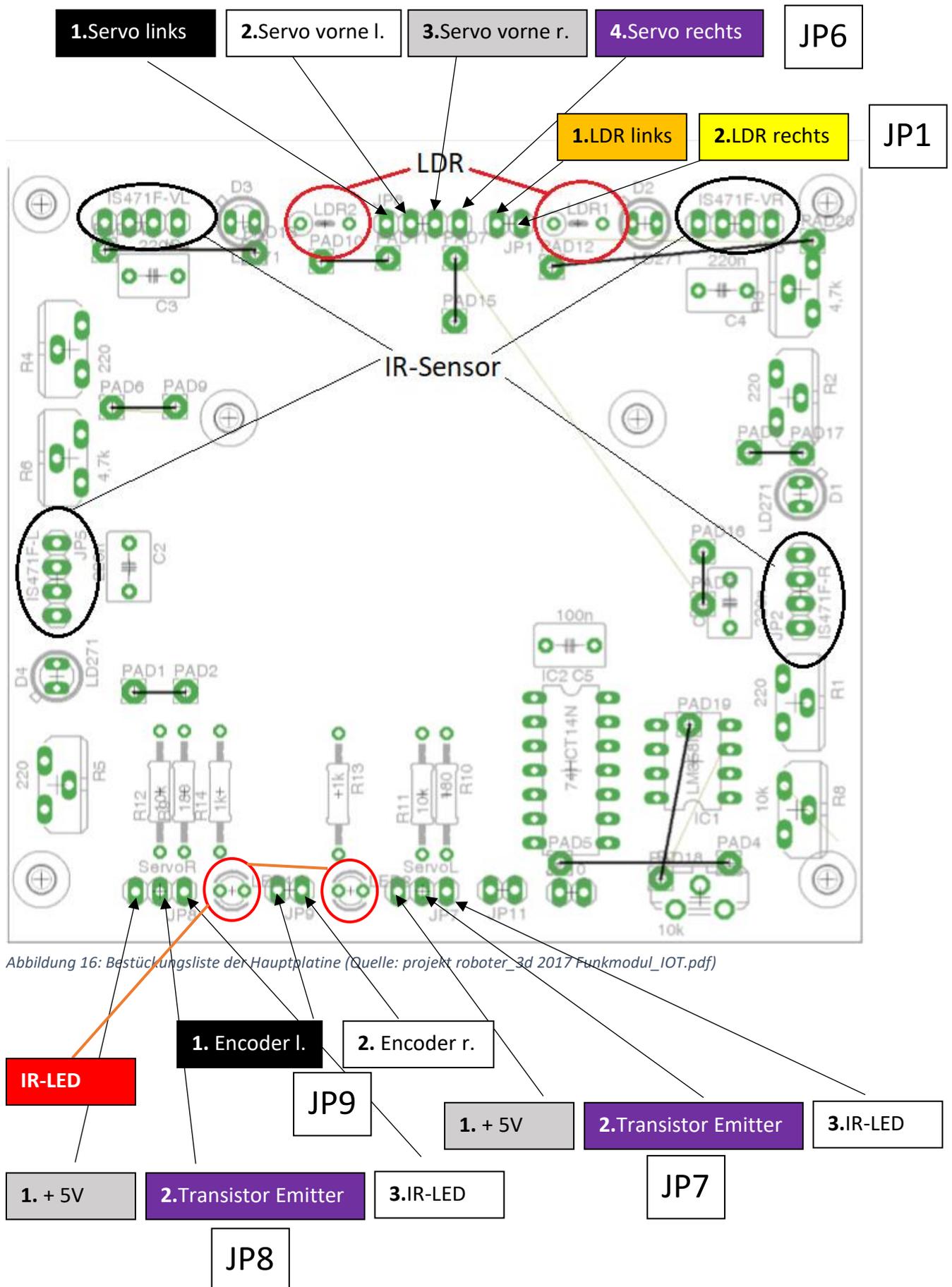


Abbildung 16: Bestückungsliste der Hauptplatine (Quelle: projekt roboter_3d 2017 Funkmodul_IOT.pdf)

3.4.3. Quellcode

```
//State 4 - Wanzenverhalten
```

```
void state4(){
  iPwmC=1500;
  fnRadOn();
  1 if((adc_data[3]>450) && (adc_data[1]>450) &&
    (fnCompare(adc_data[3],adc_data[1]))){
    DirectionR=0;
    DirectionL=0;
    statel();
  }
  2 }else if((adc_data[3]>450) && (adc_data[3]>adc_data[1])){
    DirectionR=1;
    DirectionL=0;
    statel();
  }
  3 }else if((adc_data[1]>450) && (adc_data[1]>adc_data[3])){
    DirectionR=0;
    DirectionL=1;
    statel();
  }
}
state0();
```

Abbildung 17: Wanzenverhalten-Code

Beim Wanzenverhalten wird die Geschwindigkeit des Roboters klein gestellt. Wenn der Roboter eine hohe Lichtquelle entdeckt, entfernt er sich von diesem. In [Abbildung 1](#) vergleicht das Programm die entnommene Werte der LDRs und fährt von der Lichtquelle. Die [Abbildung 2](#) zeigt den Quellcode von dem linken LDR. Wenn der Wert dieser LDR zu hoch ist dreht sich der Roboter auf die andere Seite. Die [Abbildung 3](#) zeigt den Quellcode von dem rechten LDR. Hier wird dieselbe Methode verwendet nur auf den rechten LDR bezogen.

```
//State 5 - Mottenverhalten
```

```
void state5(){
  iPwmC=1500;
  fnRadOn();
  1 if((adc_data[3]>450) && (adc_data[1]>450) &&
    (fnCompare(adc_data[3],adc_data[1]))){
    DirectionR=1;
    DirectionL=1;
    statel();
  }
  2 }else if((adc_data[3]>450) && (adc_data[3]>adc_data[1])){
    DirectionR=0;
    DirectionL=1;
    statel();
  }
  3 }else if((adc_data[1]>450) && (adc_data[1]>adc_data[3])){
    DirectionR=1;
    DirectionL=0;
    statel();
  }
}
state0();
}
```

Abbildung 18: Mottenverhalten- Code

Beim Mottenverhalten wird die Geschwindigkeit des Roboters klein gestellt. Wenn der Roboter eine hohe Lichtquelle entdeckt, nähert es sich zu diesem. In [Abbildung 1](#) vergleicht das Programm die entnommene Werte der LDRs und fährt zu der Lichtquelle. Die [Abbildung 2](#) zeigt den Quellcode von dem linken LDR. Wenn der Wert dieser LDR zu hoch ist dreht sich der Roboter zur Seite der Lichtquelle hin und fährt zu diesem weiter. Die [Abbildung 3](#) zeigt den Quellcode von der rechten LDR. Hier ist dieselbe Methodik auf den rechten LDR bezogen.

3.5. Motorsteuereinheit

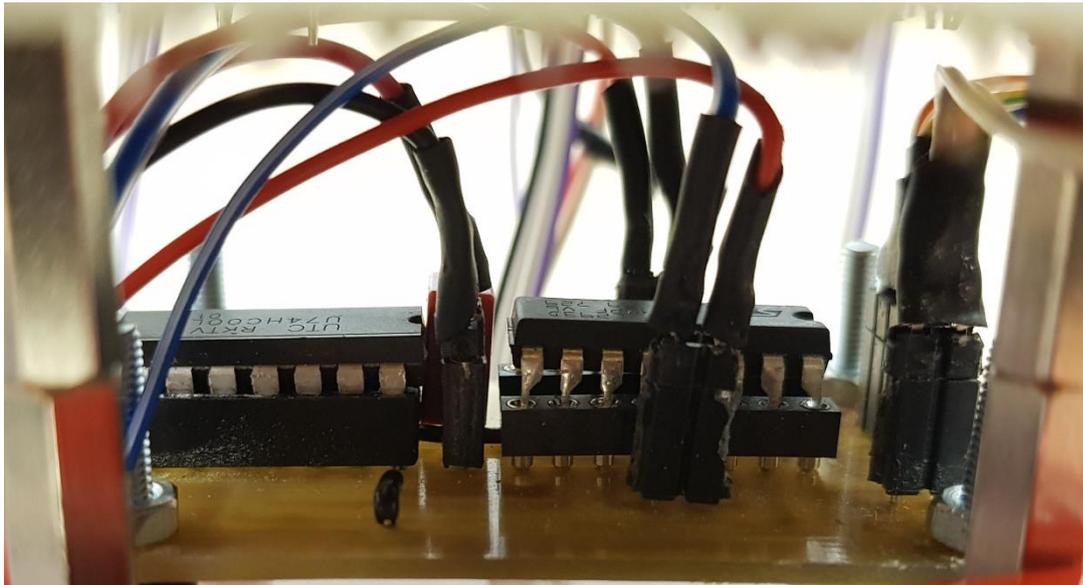


Abbildung 19: Motosteuerereinheit

Die Motorsteuereinheit kann die Geschwindigkeit und Drehrichtung der beiden Motoren steuern. Die Servos werden mit PWM (Pulsweitenmodulation) gesteuert. Somit kann man die Geschwindigkeit anpassen und auch so fahren lassen wie man es möchte.

Als erstes wird ein Signal vom Megaboard zum Motorsteuereinheit gesendet. Die jeweiligen Signale werden von den ICs in Anspruch genommen und die Servos werden mit der gewünschten Schnelligkeit angesteuert.

3.5.1. Schaltplan

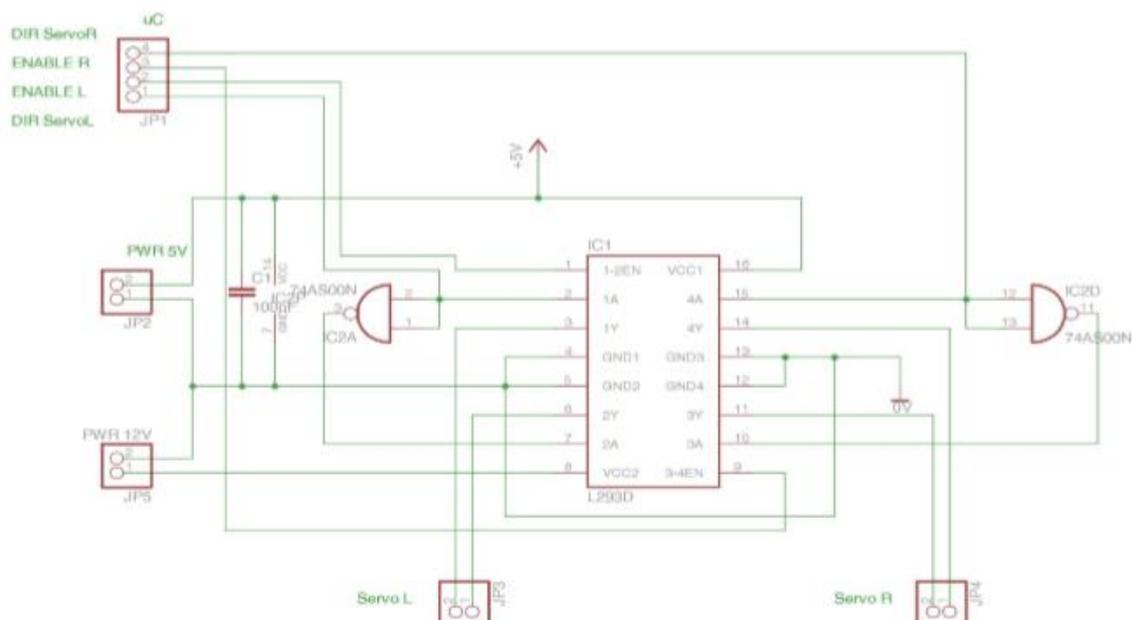


Abbildung 20: Schaltplan der Motorsteuereinheit (Quelle: projekt roboter_3d 2017 Funkmodul_IOT.pdf)

Um die Motoren zu betreiben, werden neben der Versorgungsspannung (5 Volt) auch 12 Volt an der Motorsteuereinheit angeschlossen (siehe JP2 & JP5). Die ICs werden direkt über einen Anschlussclip, der wiederum mit dem Batteriehalter verbunden ist, versorgt. Die normale Spannungsversorgung der Platine wird mit dem Megaboard verbunden. Die Eingänge 1A, 2A, 3A und 4A an der IC L293D steuern die Transistoren und somit auch die Servos. Die Servos laufen mit den Schaltern 1A und 4A nach vorne und mit den Schaltern 2A und 3A nach hinten. Die dazwischen vorhandene Schaltung der Schalter nennt sich H-Brücke.

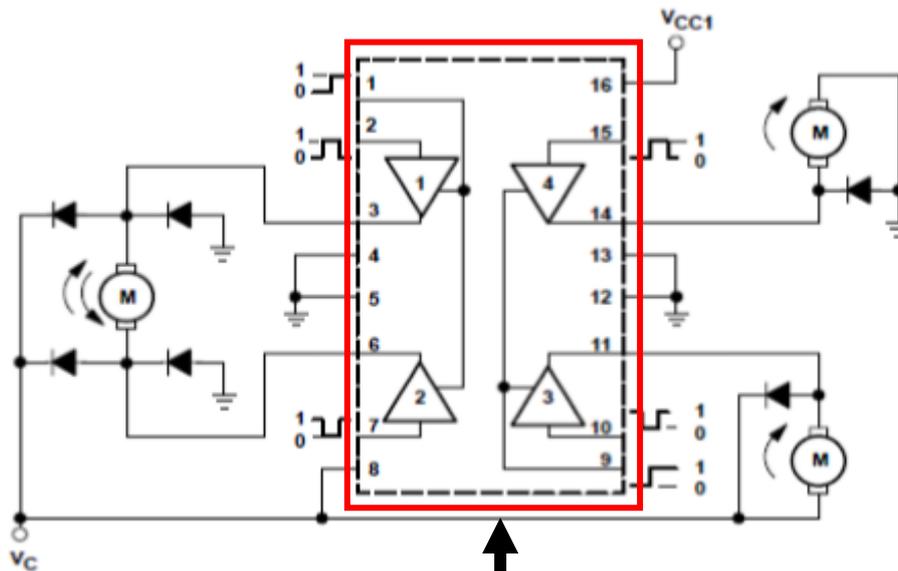


Abbildung 21: IC L293D (Quelle: projekt_roboter_3d 2017 Funkmodul_IOT.pdf)

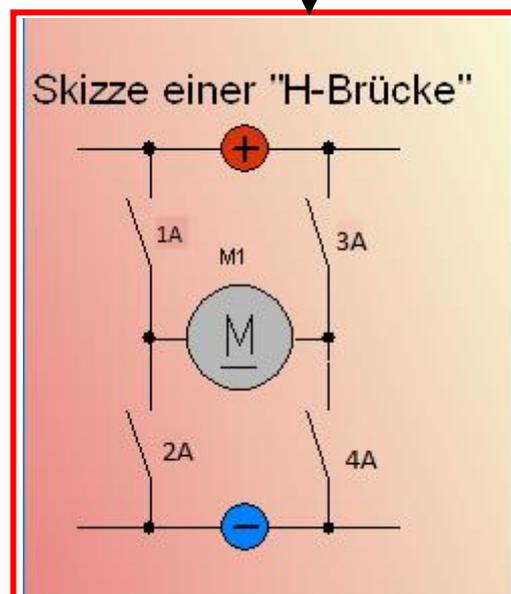


Abbildung 22: H-Brücke (Quelle: <http://www.stripfenstrolch.de/1-2-9-motortreiber-298.html>)

3.5.2. Bestückungsliste

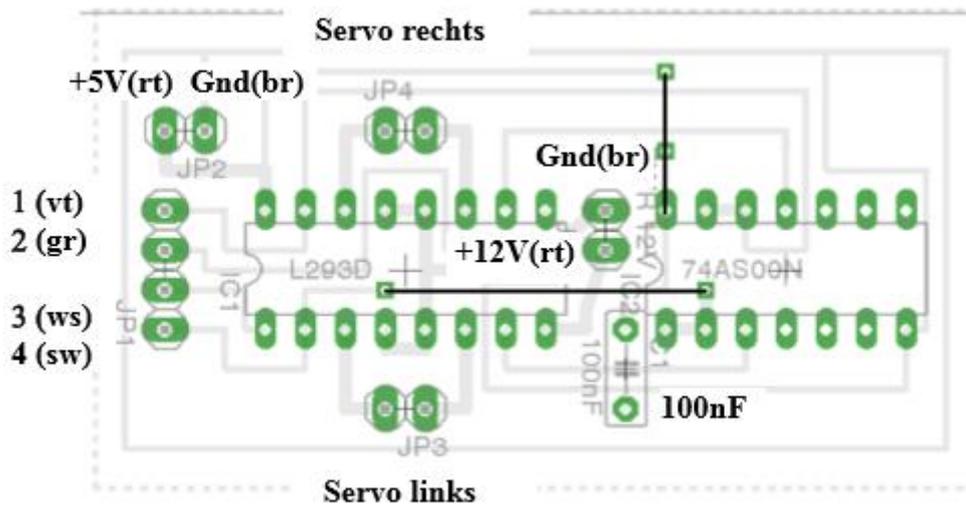


Abbildung 23: Bestückungsliste der Motorsteuereinheit (Quelle: projekt_roboter_3d 2017 Funkmodul_IOT.pdf)

Die wichtigsten Bauteile der Motorsteuereinheit sind die ICs, der Motortreiber (IC L293) und der Logik Baustein (74HC00) der vier NANDS enthält. Die Buchsen JP3 und JP4 führen jeweils zum linken und rechten Servo. Über den PIN1 der Buchse JP1 wird die Platine mit der Versorgungsspannung versorg. PIN2 ist mit Masse verbunden. Vom Megaboard aus kommen die Signale an den Pins PIN 3 und PIN 4 an.

Stückliste:

- 1 x IC1, IC L293D (Motortreiber)
- 1 x IC2, IC 74HC00 (NAND-Gatter)
- 1 x C1, Kondensator 100 nF
- 1 x 14 + 16 polige IC-Fassung
- 1 x 4 polige Buchse
- 1 x 2 polige Buchse
- 1 x Platine Motortreiber

3.5.3. Quellcode

1	<pre>case 32: //PWM - langsam if (iPwmC>500) iPwmC-=500; if (iPwmC<500) iPwmC=300; if (iState!=iStates) iState=iStates; break;</pre>
2	<pre>case 33: //PWM - schneller if (iPwmC<2300) iPwmC+=500; if (iPwmC>2300) iPwmC=2300; if (iState!=iStates) iState=iStates; break;</pre>
3	<pre>case 16: //vorwärts DirectionR=1; DirectionL=1; fRadOn(); statel(); break;</pre>
4	<pre>case 22: //recht drehene 90 grad DirectionR=0; DirectionL=1; fRadOn(); iEncoderR=iEncoderR+NINE; iEncoderL=iEncoderL+NINE; do{ statel(); }while(iEncoderR<=iEncoderR && iEncoderL<=iEncoderL); iState=0; break;</pre>
5	<pre>case 17: //rückwärts DirectionR=0; DirectionL=0; fRadOn(); statel(); break;</pre>
6	<pre>case 21: //links drehen 90 grad DirectionR=1; DirectionL=0; fRadOn(); iEncoderR=iEncoderR+NINE; iEncoderL=iEncoderL+NINE; do{ statel(); }while(iEncoderR<=iEncoderR && iEncoderL<=iEncoderL); iState=0; break;</pre>

Abbildung 24: Aktorik-Code

In **Abbildung 1** ist die Funktion „PWM-langsam fahren“ beschrieben. Nachdem die Funktion aufgerufen wird, wird die Geschwindigkeit des Roboters eine Stufe runtergesetzt. In **Abbildung 2** ist die Funktion „PWM-schnell fahren“. Hier sehen wir dieselbe Methode, doch der Unterschied liegt, dass die Geschwindigkeit eine Stufe hochgesetzt wird. In **Abbildungen 3 & 5** sind die Funktionen „vorwärts & rückwärts“. Entweder kann der Roboter vorwärts **oder** rückwärts fahren. Die Funktion in **Abbildung 4** befiehlt dem Roboter sich um 90 Grad nach rechts zu drehen. In **Abbildung 6** sehen wir dieselbe Funktion, doch der Unterschied liegt, dass sich der Roboter diesmal 90 Grad nach links dreht.

3.5.4. Timer 2

Hier haben sie einen Einblick über den Timer 2. Dies ist der Timer der Pulsweitenmodulation. Informationen über den Timer selbst und über die Initialisierung sind im folgendem aufgelistet.

Timer 2:

- 8 Bit
- Clock source: System Clock
- Clock value: 16000,000 kHz
- Mode: Normal top=0xFF
- Timer Period: 0,016 ms
- $TCCR2=(0\ll WGM20) | (0\ll COM21) | (0\ll COM20) | (0\ll WGM21) |$
- $(0\ll CS22) | (0\ll CS21) | (1\ll CS20)$
- $TCNT2=0x00$
- $TCCR2 : (0\ll CS22) | (0\ll CS21) | (1\ll CS20) = 001 = \text{no prescale} = \mathbf{16\ MHz}$

3.6. Liniensensor

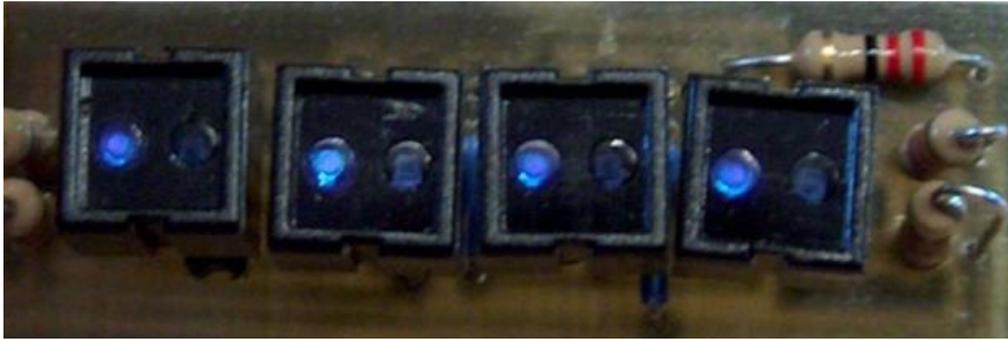


Abbildung 25: Liniensensor

Der Liniensensor ist ein Bauteil, das Linien auf dem Boden folgen kann. Die einzelnen Liniensensoren sind Reflexkoppler. Ein Reflexkoppler ist ein zusammengesetzter Baustein aus einer IR-Diode und einem Fotosensor. Der Reflexkoppler sendet ein LED Licht aus und misst, wie viel davon wieder ankommt. Da die Sensoren nebeneinander gereiht sind, kann man die Veränderungen einer Linie erkennen und dieser folgen.

3.6.1. Schaltplan

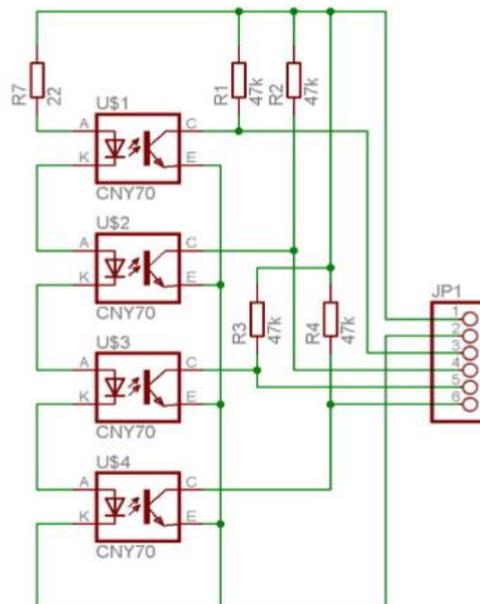


Abbildung 26: Schaltplan der Liniensensor (Quelle : projekt_roboter_3d 2017 Funkmodul_IOT.pdf)

Die Liniensensoreinheit besteht insgesamt aus vier Reflexkopplern (US1, US2, US3, US4), vier Widerstände (R1, R2, R3, R4) und einen Jumper (JP1). Vor jedem Collector Eingang eines Sensors bzw. Transistors ist ein Vorwiderstand. Alle Transistoren sind an Masse verbunden. Die Dioden werden mit 22Ω begrenzt. Die Dioden und Transistoren sind in Reihe geschaltet.

3.6.2. Bestückungsliste

Das Wichtige an der Liniensensoreinheit ist die Anschlussbelegungen des Pfostenverbinders (JP1). PIN1 ist an der Versorgungsspannung(5V) verbunden und PIN 2 an Masse.

Die Sensoren:

1. PIN3 mit Sensor1
2. PIN4 mit Sensor2
3. PIN5 mit Sensor3
4. PIN6 mit Sensor4

Stückliste:

- 4 x CNY70 (Reflexkoppler)
- 8 x 2 polige Steckkontakte
- 1 x 6 polige Buchse
- 4 x R1-4, Widerstand 47k Ω (gb-vl-or)
- 1 x R7, Widerstand 22 Ω (rt-rt-sw)
- 1 x Platine Liniensensor

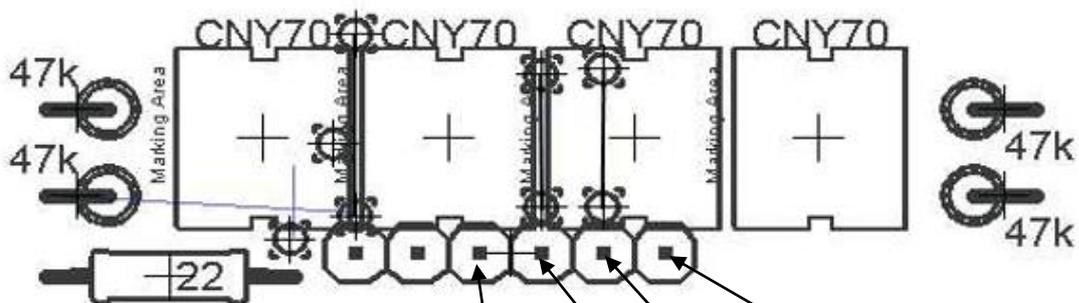


Abbildung 27: Bestückungsliste der Liniensensor (Quelle: projekt roboter_3d 2017 Funkmodul_IOT.pdf)



3.6.3. Quellcode

```
//State 6 - Linie finden & folgen
void state6(){
    iPwmC=2100;
    /*if(adc_data[0]<600 && adc_data[2]<600 &&
    adc_data[4]<600 && adc_data[6]<600 && bLinie){
        iPwmC=1500;
        DirectionR=1;
        DirectionL=0;
        bRadR=1;
        bRadL=1;
        statel();}*/
    if(adc_data[0]<600 && adc_data[2]<600 &&
    adc_data[4]<600 && adc_data[6]<600){
        DirectionR=1;
        DirectionL=1;
        fnRadOn();
        statel();
        bLinie=0;
    }
    else{
        bLinie=1;
        DirectionR=1;
        DirectionL=1;
        //Linie gefunden, Roboter richten
        if(adc_data[0]>600 && adc_data[2]>600 &&
        adc_data[4]>600 && adc_data[6]>600){
            bRadR=0;
            bRadL=1;
            statel();
        }
        //grade linie folgen
        /*if(adc_data[0]<600 && (adc_data[2]>600 |
        adc_data[4]>600) && adc_data[6]<600){
            bRadR=1;
            bRadL=1;
            statel();
        }*/
        //rechter sensor, rechts biegen
        if(adc_data[0]>600 && ( adc_data[2]<600 |
        adc_data[4]<600 | adc_data[6]<600)){
            bRadR=0;
            bRadL=1;
            statel();
        }
        //linker sensor, links biegen
        if((adc_data[0]<600 | adc_data[2]<600 |
        adc_data[4]<600) && adc_data[6]>600){
            bRadR=1;
            bRadL=0;
            statel();
        }
    }
}
```

Abbildung 28: Liniensensor Code

Der Quellcode in **Abbildung 1** sucht nach einer Linie. Wenn die ADC-Werten kleiner als 600 sind, bedeutet, dass die untere Fläche weiß ist. Sobald die Werte über 600 sind, bedeutet es dass die untere Fläche schwarz ist. In **Abbildung 2** folgt der Roboter der gefundenen Linie. Außerdem prüft der Quellcode in **Abbildung 3** nach Kurven der Linie. Wenn die Linie anfängt in eine andere Richtung zu gehen, wird der Linie in die Kurve weiter gefolgt. An der **Abbildung 4** wird die gleiche Methode ausgeführt nur diesmal durch den linke Sensor.

3.7. Schallwellensensor

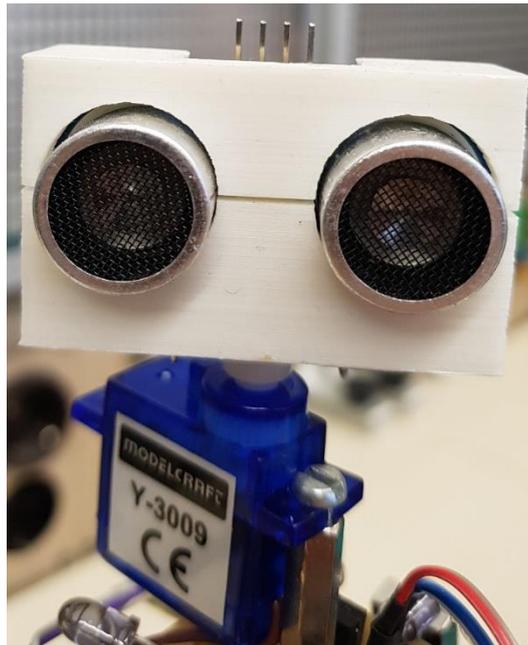


Abbildung 29: Schallwellensensor

Das Ultraschall Modul HC-SR04 ist auf dem blauen Servo montiert. Er sammelt Informationen über Objekte in der Umgebung. Er kann Entfernungen im Bereich zwischen 2cm und ca. 3m mit einer Genauigkeit von 3mm messen. Der Schallwellensensor wird normal mit einer Versorgungsspannung (5V) versorgt.

Energieversorgung	VCC +5 / GND 0V
Signalpegel (Trigger, Echo)	TTL Pegel
messbare Distanz	2cm – ca. 300cm
Messintervall	0,3cm
Messungen pro Sekunde	maximal 50
Abmessungen (länge, breite, tiefe)	45 x 21 x 18

Pinbeschreibung:

1. VCC, Versorgungsspannung 5V
2. Triggereingang, TTL-Pegel
3. Echo, Ausgang Messergebnis, TTL-Pegel
4. GND, 0V

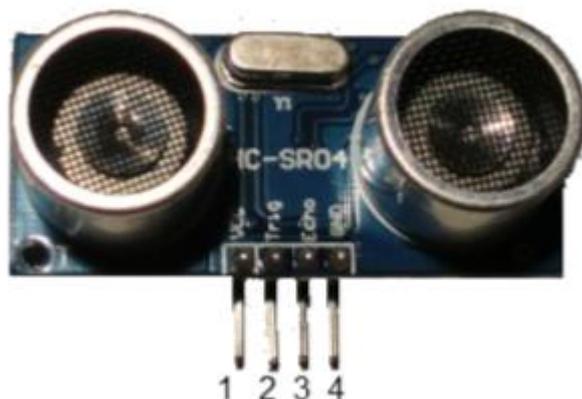


Abbildung 30: Anschlusspins der Schallwellensensor (Quelle: HC-SR04_ultraschallmodul_beschreibung_3.pdf)

3.7.1. Auswertung der Messung

Um die Entfernung zum Messobjekt zu berechnen, wird erstmal die Schallgeschwindigkeit in Luft von 343m/S herangezogen. 343m/S entsprechen 34,3cm pro Millisekunden. Dann wird die gemessene Laufzeit (T = 10.40ms) mit der Schallgeschwindigkeit multipliziert. Die komplette Rechnung wird nochmal durch 2 geteilt:

$$(10,4\text{ms} \times 34,3\text{cm}) / 2 = 178,3\text{cm}$$

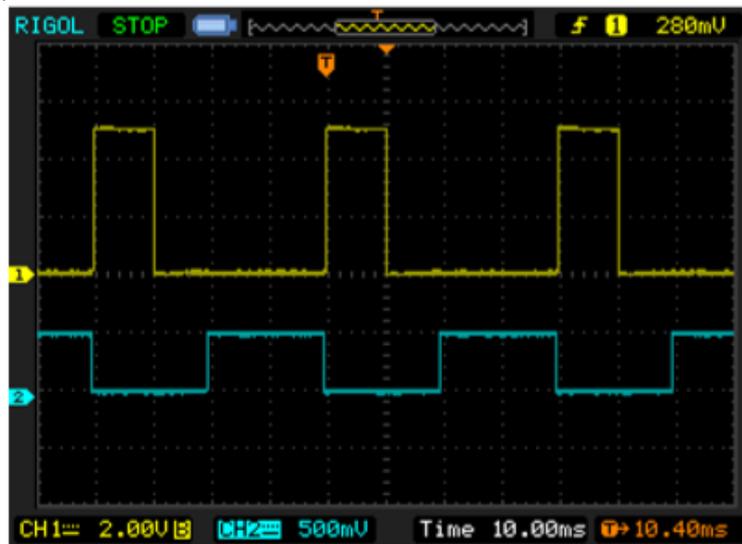


Abbildung 31: Messung der Objekte (Quelle: HC-SR04_ultraschallmodul_beschreibung_3.pdf)

3.7.2. Timer 1

Hier haben sie einen Einblick über den Timer 1. Dies ist der Timer des Servos. Informationen über den Timer selbst und über die Initialisierung sind im folgendem aufgelistet.

Timer 1:

- 16 Bit
- Clock source: System Clock
- Clock value: 250,000 kHz
- Mode: CTC top=OCR1A
- Input Capture on Falling Edge
- Timer Period: 4 us
- Compare A Match Interrupt: On

- $TCCR1B = (0 \ll ICNC1) | (0 \ll ICES1) | (0 \ll WGM13) | (1 \ll WGM12) | (0 \ll CS12) | (1 \ll CS11) | (1 \ll CS10)$
- $TCCR1B : (0 \ll CS12) | (1 \ll CS11) | (1 \ll CS10) = 011 = /64 = 16 \text{ Mhz} / 64 = \mathbf{250\ 000ms}$

3.7.3. Timer 3

Hier haben sie einen Einblick über den Timer 3. Dies ist der Timer des Schallwellensensors. Informationen über den Timer selbst und über die Initialisierung sind im folgendem aufgelistet.

Timer 3:

- 16 Bit
- Clock source: System Clock
- Clock value: 250.000 kHz
- Mode: CTC top=OCR3A
- Noise Canceler: On
- Input Capture on Rising Edge
- Timer Period: 0.1 s
- Input Capture Interrupt: On
- Compare A Match Interrupt: On

- $TCCR3B = (1 \ll ICNC3) | (1 \ll ICES3) | (0 \ll WGM33) | (1 \ll WGM32) | (0 \ll CS32) | (1 \ll CS31) | (1 \ll CS30)$
- $OCR3AH = 0x61$
- $OCR3AL = 0xA8$
- $TCCR3B : (0 \ll CS32) | (1 \ll CS31) | (1 \ll CS30) = 011 = /64 = 16\text{Mhz} / 64 = \mathbf{250\ 000ms}$

3.8. Radencoder

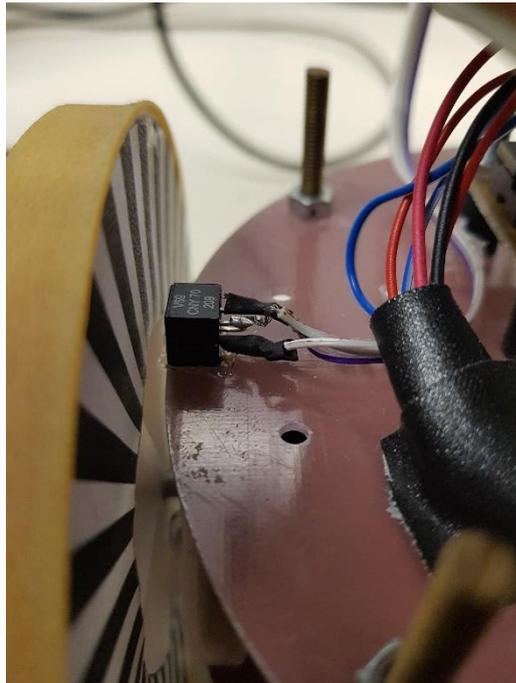


Abbildung 32: Radencoder

Der Radencoder ist ein Reflexkoppler. Ein Reflexkoppler ist ein zusammengesetzter Baustein aus einer IR-Diode und einem Fotosensor. Der Reflexkoppler sendet ein LED Licht aus und misst, den Wert der wieder ankommt. Es gibt jeweils für die weißen und schwarzen Farben konstante Werte. Wenn das Rad sich dreht wechseln die Farben von schwarz auf weiß. Bei jeder Farbe Wechselung berechnet der Reflexkoppler einen Puls. Mit den Pulsen wird eine Wegmessung implementiert. Die Wegmessungen werden später im Programm für Berechnungen weiterverwendet.

3.9. Bumper

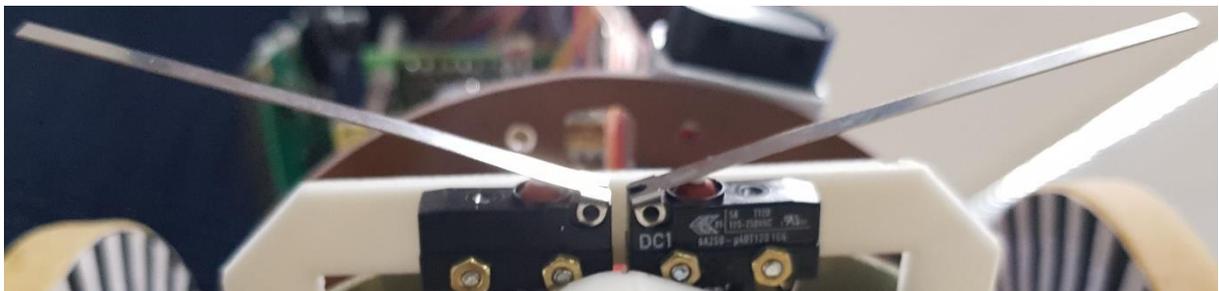


Abbildung 33: Bumper

Der Zustand des Roboters ist immer auf High (== TRUE). Sobald der Tastschalter betätigt wird, wird der Zustand des Bumpers auf Low gesetzt. Somit weiß der Roboter das der Taster betätigt wurde. Damit kann der Roboter erkennen ob er gegen ein Objekt gefahren ist.

3.10. Master Mega128

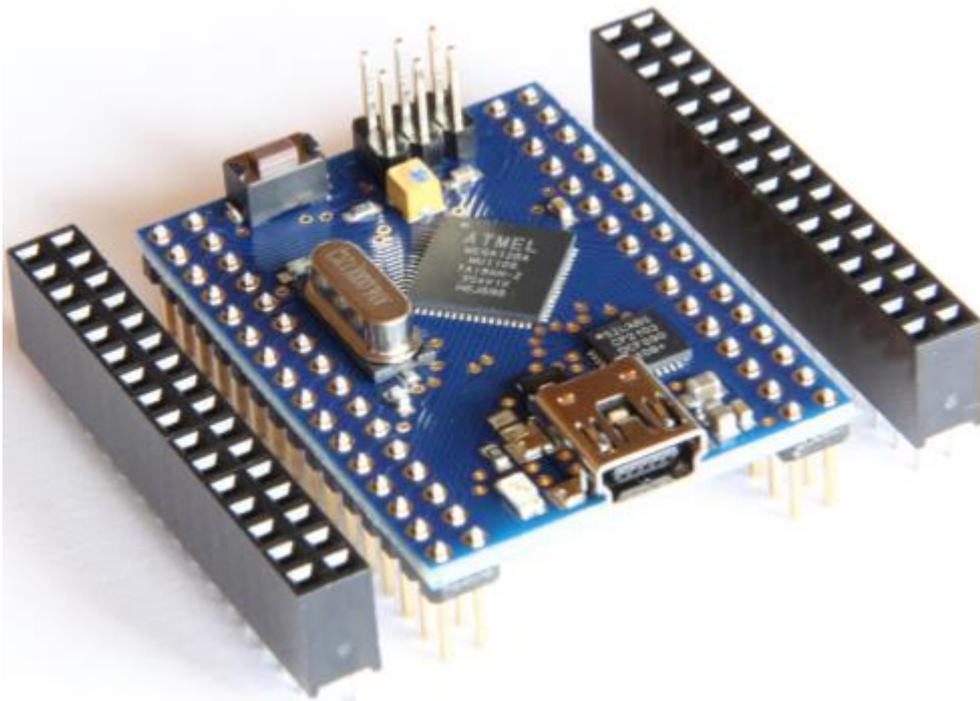


Abbildung 34: Mega128 (Quelle: projekt_roboter_3d 2017 Funkmodul_IOT.pdf)

Ein Mikrocontroller ist ein Ein-Chip Computersystem. Microcontroller haben einen CPU und Peripheriefunktionen. Peripheriefunktionen sind außenstehende Hardwarekomponenten welche mit dem Mikrocontroller verbunden sind und dadurch angesprochen werden können z.b Wii Cam mit dem Mega128.

Jeder Schüler in der Klasse hat den Microcontroller bevor wir mit dem Projekt angefangen haben, in einer 3 bis 4-wöchigen Zeitraum im Rechner-/Systemtechnik Unterricht zusammengebaut. Nachdem die Gruppen erstellt wurden, wurde der Microcontroller eines Schülers für den Roboter ausgewählt.

Stückliste Mega128 Usb V2:

<ul style="list-style-type: none">➤ 1 x Kohle-Schichtwiderstand 22Ω➤ 1 x Kohle-Schichtwiderstand 10Ω➤ 1 x JP1-1, JP1-2, JP2-1, JP2-2, JP3, JP4 (Präzisionskontakte)➤ 1 x RN1, RN2 (Widerstandsnetzwerke)➤ 2 x P1, P2 (Einstellpotentiometer)➤ 1 x Dip-Schalter (4-polig)➤ 1 x Buchsenleiste (1x10-polig)➤ 1 x Stiftleiste (2x10-polig)➤ 1 x LED (Bargraphanzeige)➤ 1 x Wannenstecker (2x7-polig)➤ 1 x Elektrolytkondensator 470uF/16V➤ 1 x Drehimpulsgeber	<ul style="list-style-type: none">➤ 2 x Kurzhubtaster➤ 1 x Anschlussklemme (2-polig)➤ 1 x Pololu 5V Step-Up Wandler➤ 1 x Spannungsregler LM2940➤ 2 x Stiftleiste, 2x16 pol.➤ 1 x Kondensator 470nF➤ 1 x Mikrocontroller-Modul MEGA128-USB➤ 4 x Distanzbolzen 10 mm➤ Kohle-Schichtwiderstand 1 kΩ➤ 2x16 pol. Buchsenleiste➤ 1 x Elektrolytkondensator 10uF/35V
---	---

Stückliste LCD Modul:

<ul style="list-style-type: none">➤ 1 x Flachbandkabel (14-polig)➤ 1 x Pfostenbuchse (14-polig)➤ 1 x LCD-Modul
--

Stückliste Funkmodul RFM12:

<ul style="list-style-type: none">➤ 1 x Kohle Schichtwiderstand 10kΩ➤ 3 x Kohle Schichtwiderstand 1kΩ➤ 1 x Kondensator100nF➤ 1 x Spannungsregler➤ 1 x Elektrolytkondensator 1μF➤ 1 x Elektrolytkondensator 3,3μF➤ 1 x Stiftleiste (1x7-polig)➤ 1 x Funkmodul

3.10.1. Pin Belegungen

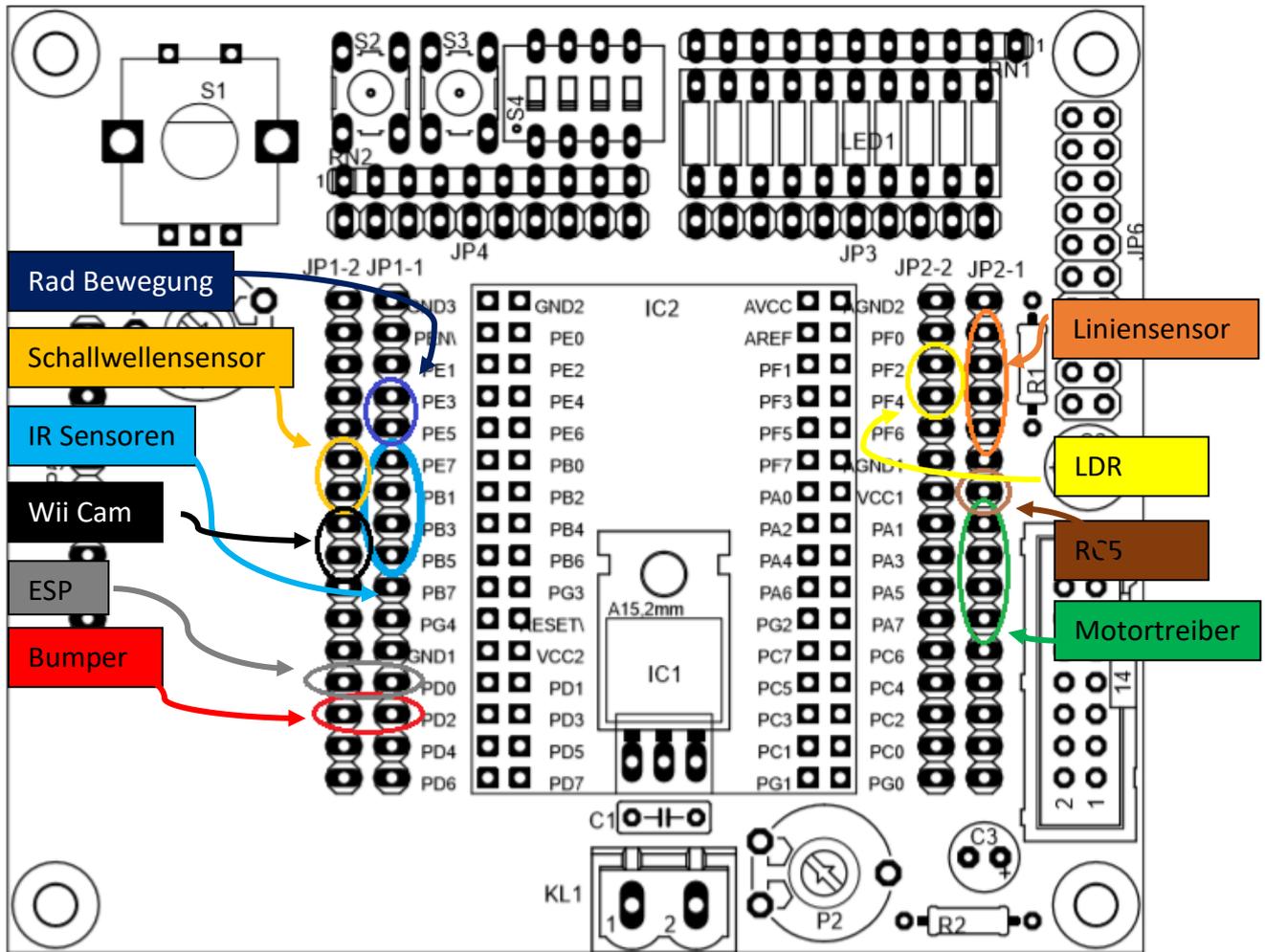
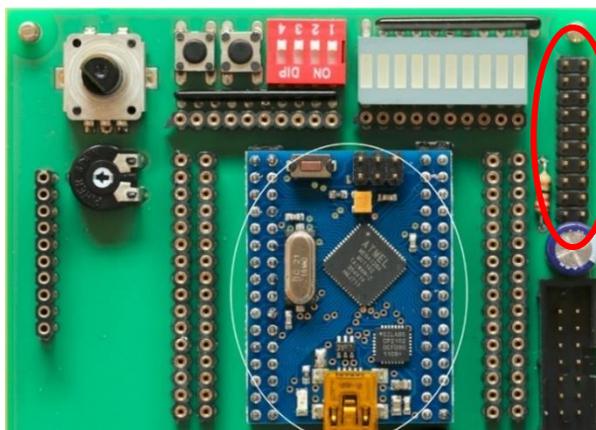


Abbildung 35: Portbelegung auf dem Microcontroller (Quelle: projekt_roboter_3d 2017 Funkmodul_IOT.pdf)

Alle Komponenten sind mit dem Megaboard verbunden. Die Portbelegungen unterscheiden sich zwischen PORTS und PINS. Wenn ein Port als PIN definiert wurde bedeutet, dass die Signale von Komponenten zum Megaboard kommen. Zum Beispiel der ESP sendet Signale zum Megaboard. Wenn ein Port als PORT definiert wird, bedeutet, dass die Signale an Komponente geschickt werden.



Außerdem sind alle Komponenten mit der Versorgungsspannung durch das Megaboard versorgt.

Versorgungsspannung

Abbildung 36: Microcontroller (Quelle: projekt_roboter_3d 2017 Funkmodul_IOT.pdf)

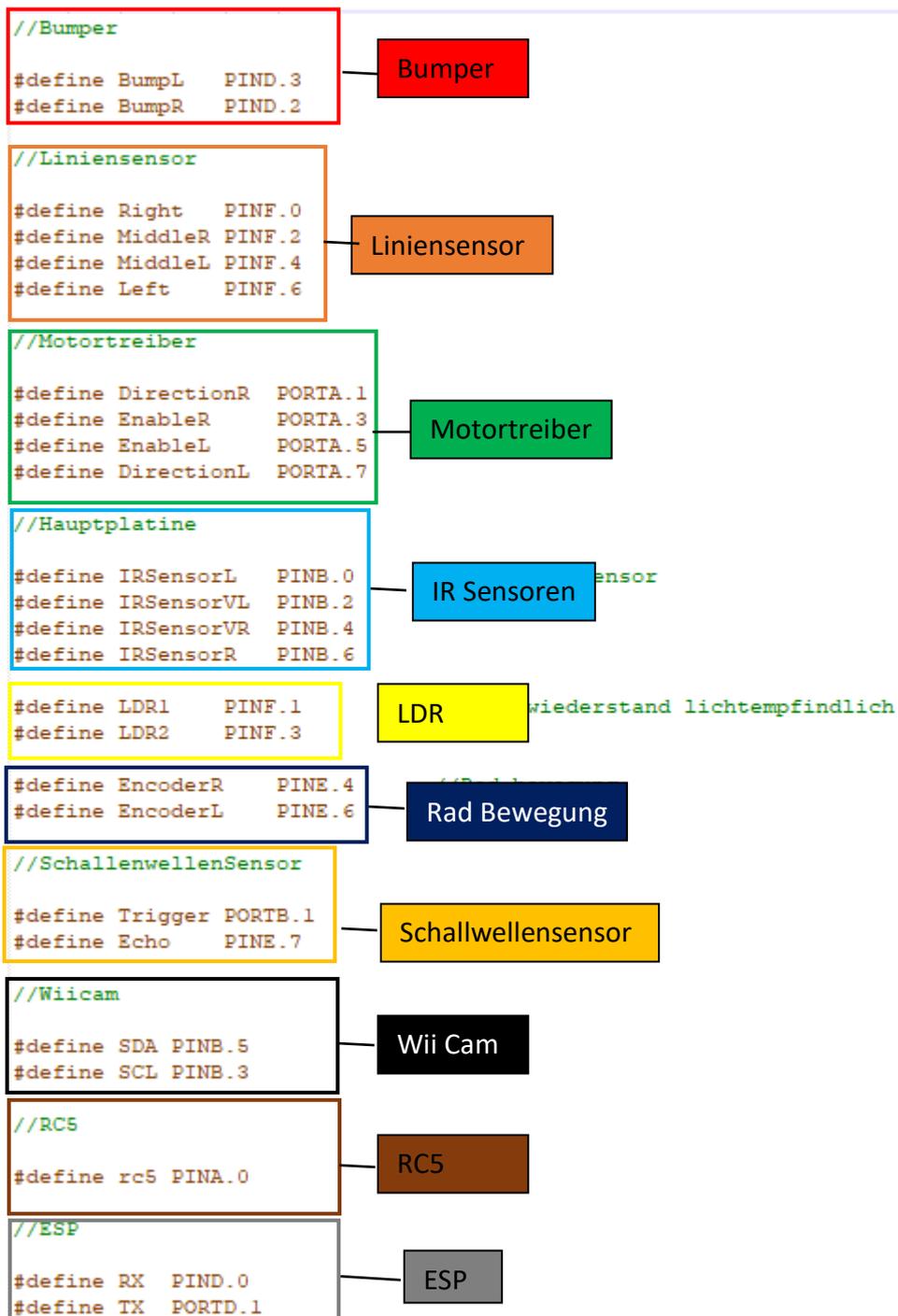


Abbildung 37: Makros der Portbelegungen

3.11. ESP8266

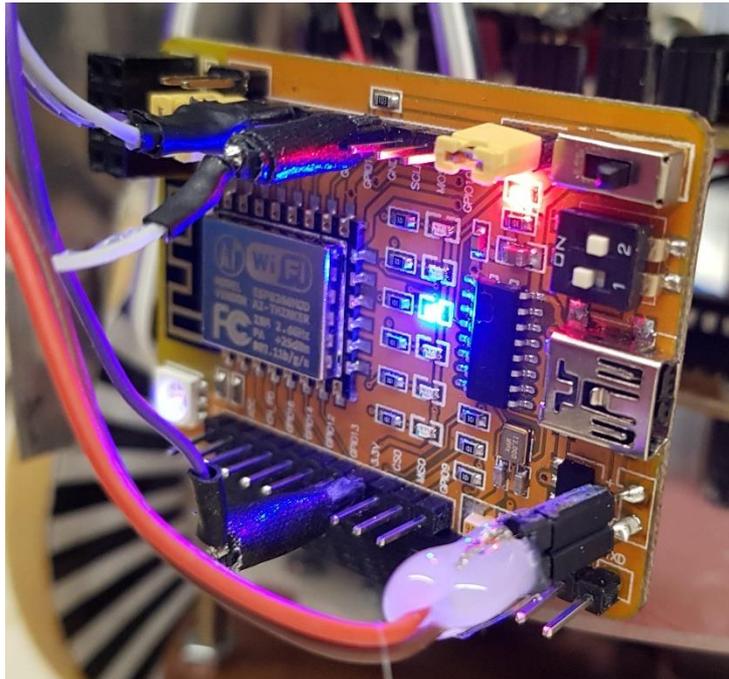


Abbildung 38: ESP8266

Der ESP8266 ist ein WIFI-Modul, das eine WIFI-Funktionalität enthält. Das Modul ist programmierbar mit UART- und SPI Schnittstelle. Eine UART Schnittstelle ist die gängige serielle Schnittstelle an PCs und Mikrocontrollern. Eine SPI Schnittstelle ist ein Bussystem und bestehend aus drei Leitungen für eine Datenübertragung zwischen verschiedenen ICs. Der ESP8266 kann auch selbst als Access-Point mit DHCP verwendet werden. Auf dem ESP8266 ist die Website drauf programmiert. Über die Website kann der Roboter gesteuert werden.

Physikalische Schnittstelle:

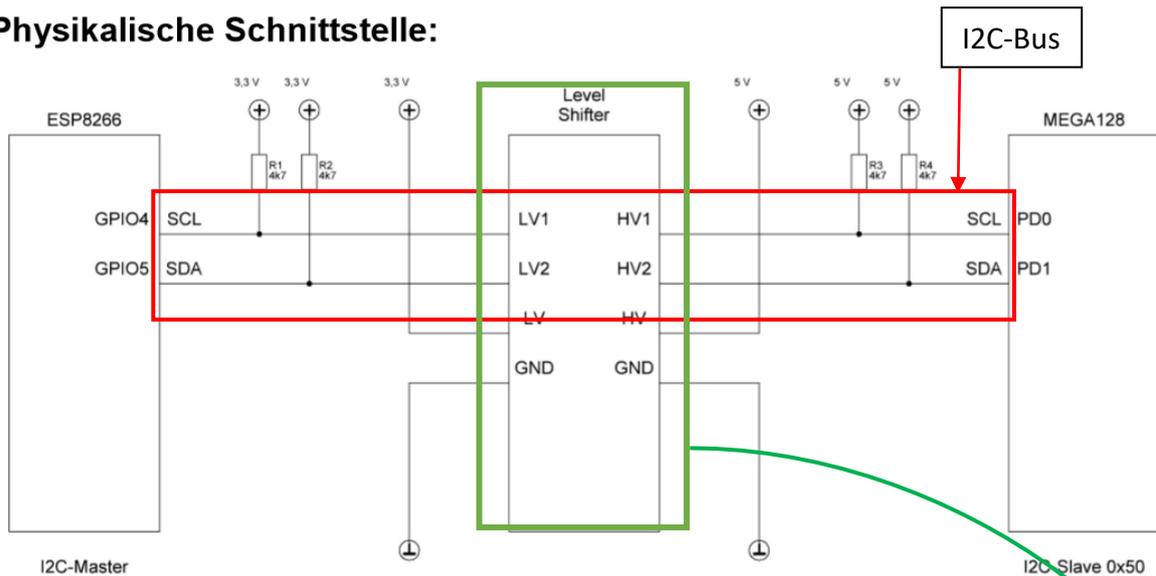
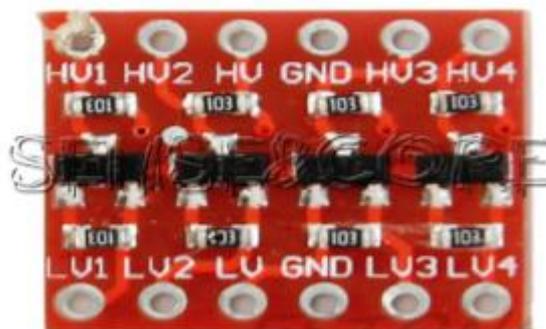


Abbildung 39: Schnittstelle & Signalaustausch (Quelle: ESP8266_I2C_MEGA128.PDF)

Zwischen dem ESP8266 und dem Mega128 werden Daten über den I2C-Bus ausgetauscht. Der ESP8266 fungiert bei der Kommunikation als I2C-Master. Die Signale SDA und SCA werden auf den Ports GPIO4 und GPIO5 des Logic Level Converters gesendet. Die Signale werden über einen PullUp-Widerstand ($4,7k\Omega$) mit $+3,3V$ verbunden.

Der Mega128 wirkt bei der Kommunikation als I2C-Slave. Die Signale werden über den Ports PD0 und PD1 gesendet. Diese sind auch über einen PullUp-Widerstand ($4,7k\Omega$) mit $+5V$ verbunden.

Zwischen der Mega128 und dem ESP8266 befindet sich der Logic Level Converter. Dies führt die Rolle einer Brücke zwischen ESP8266 und der Mega128. Die Signale gelangen durch den Logic Level Converter zu den Controllern.



3.12. LCD Display

```
// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTA Bit 0
// RD - PORTA Bit 1
// EN - PORTA Bit 2
// D4 - PORTA Bit 4
// D5 - PORTA Bit 5
// D6 - PORTA Bit 6
// D7 - PORTA Bit 7
// Characters/line: 16
lcd_init(16);
}
```

Abbildung 40: LCD Initialisierung

Die Abkürzung LCD steht ausgesprochen für "Liquid Crystal Display" und bedeutet wörtlich übersetzt Flüssigkristallbildschirm. Das LCD Display was wir an unserem Roboter benutzen besteht aus 16 Bits, die je 8 Bits pro Zeile enthalten. Diese Bits werden durch 12 angelötete Pins angesprochen. Diese Pins werden an den Mikrocontroller angeschlossen, um in diesem Fall die States und die Befehle anzuzeigen.

```
// Global enable interrupts
#asm("sei")
i2c_init();
lcd_putsf("--Der--Geraet--");
```

Abbildung 41: Beispiel Code

In diesem Beispiel können wir sehen, dass mit dem Befehl „lcd_putsf()“ eine beliebige Zeile bzw. Nachricht am LCD ausgegeben kann.



Abbildung 42: LCD Ausgabe

4. Steuerung des Roboters

Um den Roboter über die Fernbedienung zu steuern wurde ein RC5-Empfänger im Roboter verbaut.

Der Roboter soll die Möglichkeit verfügen, Daten in ein Netzwerk zu senden und empfangen. Dafür wird der ESP8266-Chip verwendet. Dieser hat eine eigene IP-Adresse und öffnet ein Soft-Access-Point. Der ESP wurde über die Arduino-IDE programmiert. Die Kommunikation zwischen dem ESP und dem Mikrocontroller findet über die I2C Schnittstelle statt.

Der Gerät

Wert vom MLGA128
Integer: ...
Sent Befehl

Bewegen:

Vor

Links Rechts

Zurück

Stop PWM Hoch PWM Runter

Mode:

Drehen L-Fahren

Ausweichen Wanze

Motte Linie

Objekt

Sensorwerte:

WiiCam_x: ...
WiiCam_y: ...
WiiCam_z: ...
1.Reflexkoppler: ...
1.Reflexkoppler: ...
2.Reflexkoppler: ...
3.Reflexkoppler: ...
4.Reflexkoppler: ...
IRSensorL: ...
IRSensorR: ...
IRSensorVL: ...
IRSensorVR: ...
LDR1: ...
LDR2: ...
Schallwellens: ...

Connection
ReadyState: 1

Hier kann der Roboter gesteuert und die PWM hoch oder runter gestellt werden.

Hier werden die Sensorwerte die vom Mikrocontroller gesendet werden dargestellt.

In dem Carvis werden die Daten der WiiCam optisch dargestellt.

In der Textarea wird der User darüber informiert, welche Befehle gesendet werden.

In diesem Bereich kann man zwischen den einzelnen Funktionen des Roboters (Stats) wechseln.

4.1. Funktion

Der Roboter kann über die Webseite angesteuert werden. Dies funktioniert über die jeweiligen Buttons. Außerdem kann man auch zwischen den einzelnen Modi wie z.B. Wanzen-/Mottenverhalten und L-Fahren wechseln. Die Sensordaten können auch empfangen werden. Dazu werden die Daten der Wii Cam visuell dargestellt.

4.2 Tastenbelegung Fernbedienung

- Taste 0 = Roboter stoppen
- Taste 1 = Roboter drehen lassen
- Taste 2 = L-Fahren
- Taste 3 = Objekte ausweichen
- Taste 4 = Wanzenverhalten
- Taste 5 = Mottenverhalten
- Taste 6 = Linie verfolgen
- Taste 7 = Anderen Roboter verfolgen
- Taste 8 = Kartierung
- Taste Hoch = geradeaus fahren
- Taste Runter = Zurück fahren
- Taste Links = links abbiegen
- Taste Rechts = rechts abbiegen
- Taste Kanal runter = PWM langsamer
- Taste Kanal hoch = PWM schneller

5. Arduino Verbindung

```
// IP initialization
IPAddress local_IP(192, 168, 2, 100);
IPAddress gateway(192, 168, 2, 1);
IPAddress subnet(255, 255, 255, 0);

Serial.print("Setting soft-AP configuration ... ");
Serial.println(WiFi.softAPConfig(local_IP, gateway, subnet) ? "Ready" : "Failed!");

Serial.print("Setting soft-AP ... ");
Serial.println(WiFi.softAP("Geraet") ? "Ready" : "Failed!");

Serial.print("Soft-AP IP address = ");
Serial.println(WiFi.softAPIP());

Serial.print("Station IP address = ");
Serial.println(WiFi.localIP());
```

Abbildung 43: IP Initialisierung

Um den Webserver zu starten wird eine IP, ein Gateway und eine Subnetzmaske benötigt. Die IP wird hier auf 192.168.2.100 gesetzt und das Gateway auf 192.168.2.1.. Die Subnetzmaske ist 255.255.255.0. Die SSID des Access-Points wird hier „Geraet“ genannt. User wird darüber im seriellen Monitor informiert.

```
// WebSocket
WebSocket.begin();
WebSocket.onEvent(WebSocketEvent);
Serial.println("WebSocket gestartet!");

// WebServer
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request) {
  request->send_P(200, "text/html", html);
});
server.begin();
Serial.println("HTTP server gestartet!");
```

Abbildung 44: WebSocket & Webserver mit HTML Code wird gestartet

Hier wird der WebSocket und der Webserver mit dem dazugehörigem HTML-Code gestartet. Dies wird auch in der Konsole angezeigt.

```

tx_buffer.data.iValue = itest;

// transmit data from tx_buffer to the slave and receive response in rx_buffer
i2c_result = i2c_master_trans(TWI_SLAVE_ADDR, tx_buffer.bytes, BUFFER_SIZE, rx_buffer.bytes, BUFFER_SIZE);

// display the transaction on the terminal
if (i2c_result == 0)
{
    // I2C transaction was performed without errors
    Serial.printf("I2C transaction result: %i\r\n", i2c_result);
    Serial.printf("Transmitted to MEGA128:%2i \r\n", tx_buffer.data.iValue);
    Serial.printf("Received from MEGA128: %4i\r\n", rx_buffer.data.iValue);
    Serial.printf("Content of Receive Buffer: ");
    for (int i = 0; i < BUFFER_SIZE; i++)
        Serial.printf("%2X ", rx_buffer.bytes[i]);
    Serial.printf("\r\n");
}
else
{
    // I2C transaction was performed with errors
    Serial.printf("I2C transaction error: %i\r\n\r\n", i2c_result);
}
}

```

Abbildung 46: Transaktion

Hier wird der Inhalt des tx_buffers, über den I2C-Bus an die Slave-Adresse(Mikrocontroller) gesendet. Im Gegenzug bekommt der ESP den rx_buffer. Die Transaktionen werden im seriellen Monitor angezeigt.

```

if (text == "16") {
    itest=16; //Wert '16'(Vor) an ATMEGA senden
    i2c_exchange_data();
}

if (text == "21") {
    itest=21; //Wert '21'(Links) an ATMEGA senden
    i2c_exchange_data();
}

if (text == "22") {
    itest=3; //Wert '22'(Rechts) an ATMEGA senden
    i2c_exchange_data();
}

if (text == "17") {
    itest=17; //Wert '17'(Zurück) an ATMEGA senden
    i2c_exchange_data();
}

```

Abbildung 45: Website sendet zur ESP8266

Text der von der Webseite an den ESP8266-CHIP gesendet wird, wird in der Variable 'itest' gespeichert. Danach wird die Funktion i2c_exchange_data() ausgeführt. Dort wird 'itest' an den Mikrocontroller übergeben.

```

void sendJasonString()
{
    DynamicJsonBuffer jsonBuf;
    JsonObject& root = jsonBuf.createObject();
    root["action"] = "request";
    root["integer"] = rx_buffer.data.iValue;
    root["WiiCam1"] = rx_buffer.data.WiiCam1;
    root["WiiCam2"] = rx_buffer.data.WiiCam2;
    root["WiiCam3"] = rx_buffer.data.WiiCam3;
    root["Reflexkoppler1"] = rx_buffer.data.Reflexkoppler1;
    root["Reflexkoppler2"] = rx_buffer.data.Reflexkoppler2;
    root["Reflexkoppler3"] = rx_buffer.data.Reflexkoppler3;
    root["Reflexkoppler4"] = rx_buffer.data.Reflexkoppler4;
    root["IRSensorL"] = rx_buffer.data.IRSensorL;
    root["IRSensorR"] = rx_buffer.data.IRSensorR;
    root["IRSensorVL"] = rx_buffer.data.IRSensorVL;
    root["IRSensorVR"] = rx_buffer.data.IRSensorVR;
    root["LichtDR1"] = rx_buffer.data.LichtDR1;
    root["LichtDR2"] = rx_buffer.data.LichtDR2;
    root["Schallwellen"] = rx_buffer.data.Schallwellen;
    char buf[512];
    size_t size = root.printTo(buf, sizeof(buf));
    websocket.sendTXT(0, buf, size);
}

```

Abbildung 47: ESP8266 sendet an Website

Inhalt des Structs, mithilfe des JsonObject, von ESP an Webseite übergeben.

6. System Struktur

Sequenzdiagramm:

Das Sequenzdiagramm haben wir für die Verwirklichung der Verbindungen zwischen den Controllern erstellt. Es zeigt zum Beispiel den Ablauf, wenn per Fernbedienung oder per Website, dem Roboter etwas befehlt wird.

Wenn die Powerbank an dem Roboter angeschlossen wird, wird der Roboter aktiv. Wenn der Akteur die Website aufruft, wird die Seite aktiv.

Aktivitätsdiagramm & Strukturgramm:

Das Aktivitätsdiagramm und das Strukturgramm haben wir für den Ablauf nach einer Wahl der States (Aufgabe) erstellt. Es gibt insgesamt 8 States also 8 Aufgaben. Jede einzelne State enthält bestimmte Aufgaben:

1. State = Roboter bewegen
2. State = "L" fahren
3. State = Hindernis umfahren
4. State = Wanzenverhalten
5. State = Mottenverhalten
6. State = Linie befahren
7. State = IR-Objekt folgen
8. State = Umgebung kartieren

Anwendungsfalldiagramm:

Das Anwendungsfalldiagramm haben wir dafür erstellt, um zu visualisieren was der Akteur für Möglichkeiten an dem Roboter hat. Alle Möglichkeiten wurden auch so erweitert, dass die weiteren Aktivitäten durch den einen Aktivität aufgerufen werden.

6.1. Sequenzdiagramm

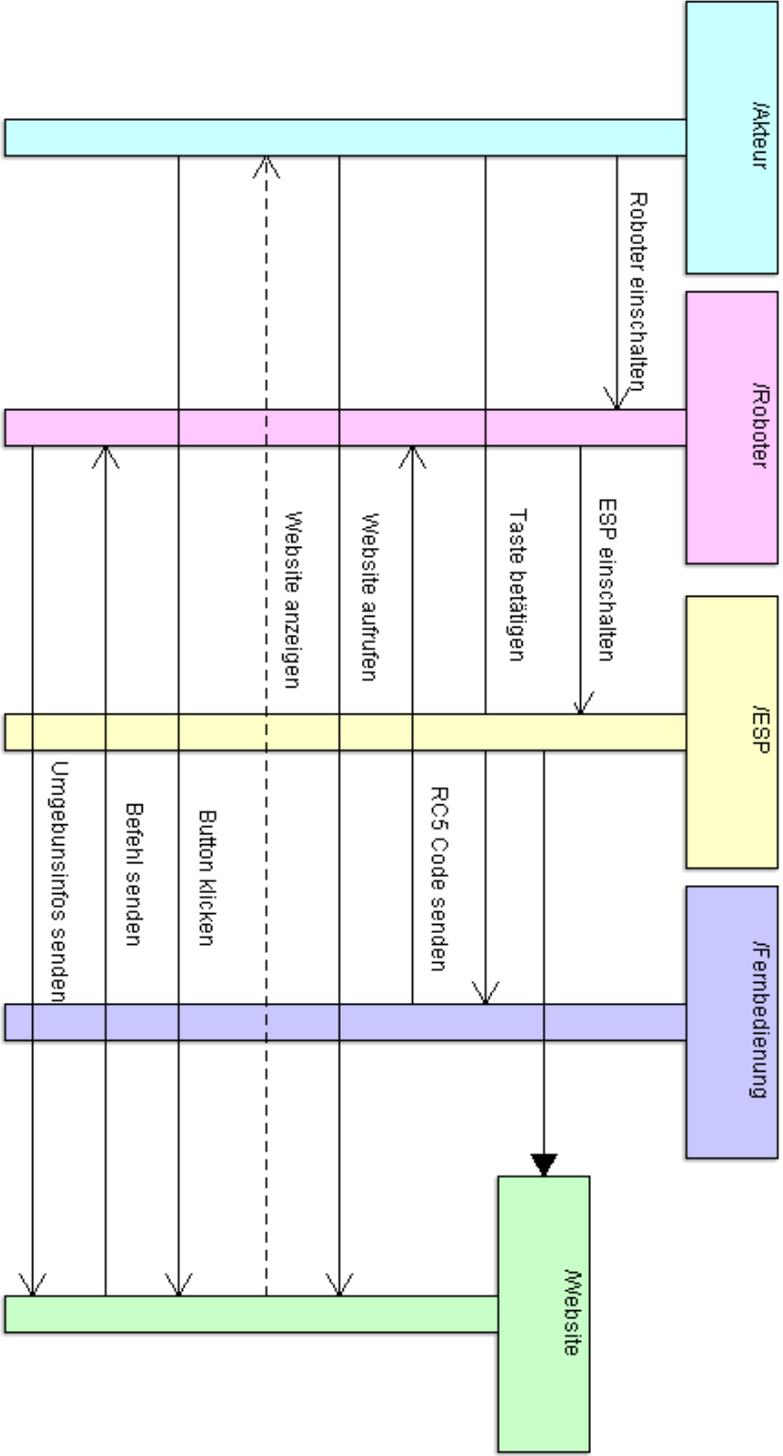


Abbildung 48: Sequenzdiagramm

6.2. Aktivitätsdiagramm

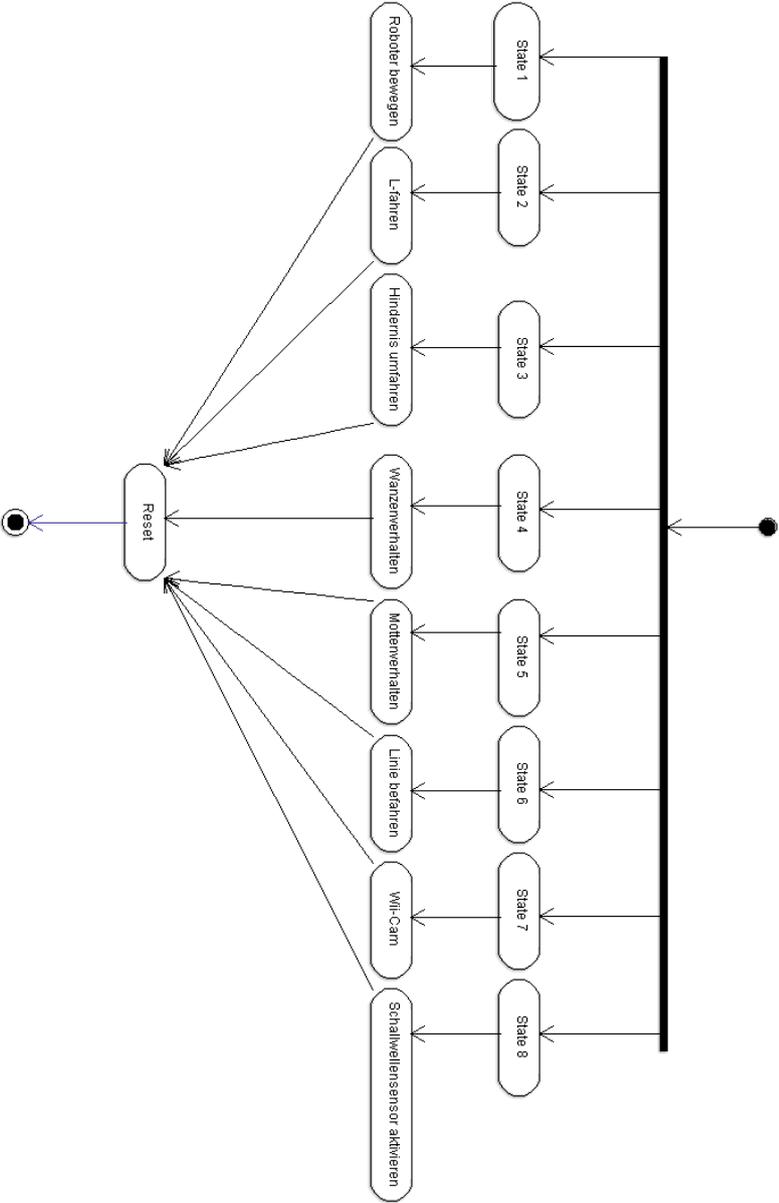


Abbildung 49: Aktivitätsdiagramm

6.3. Anwendungsfalldiagramm

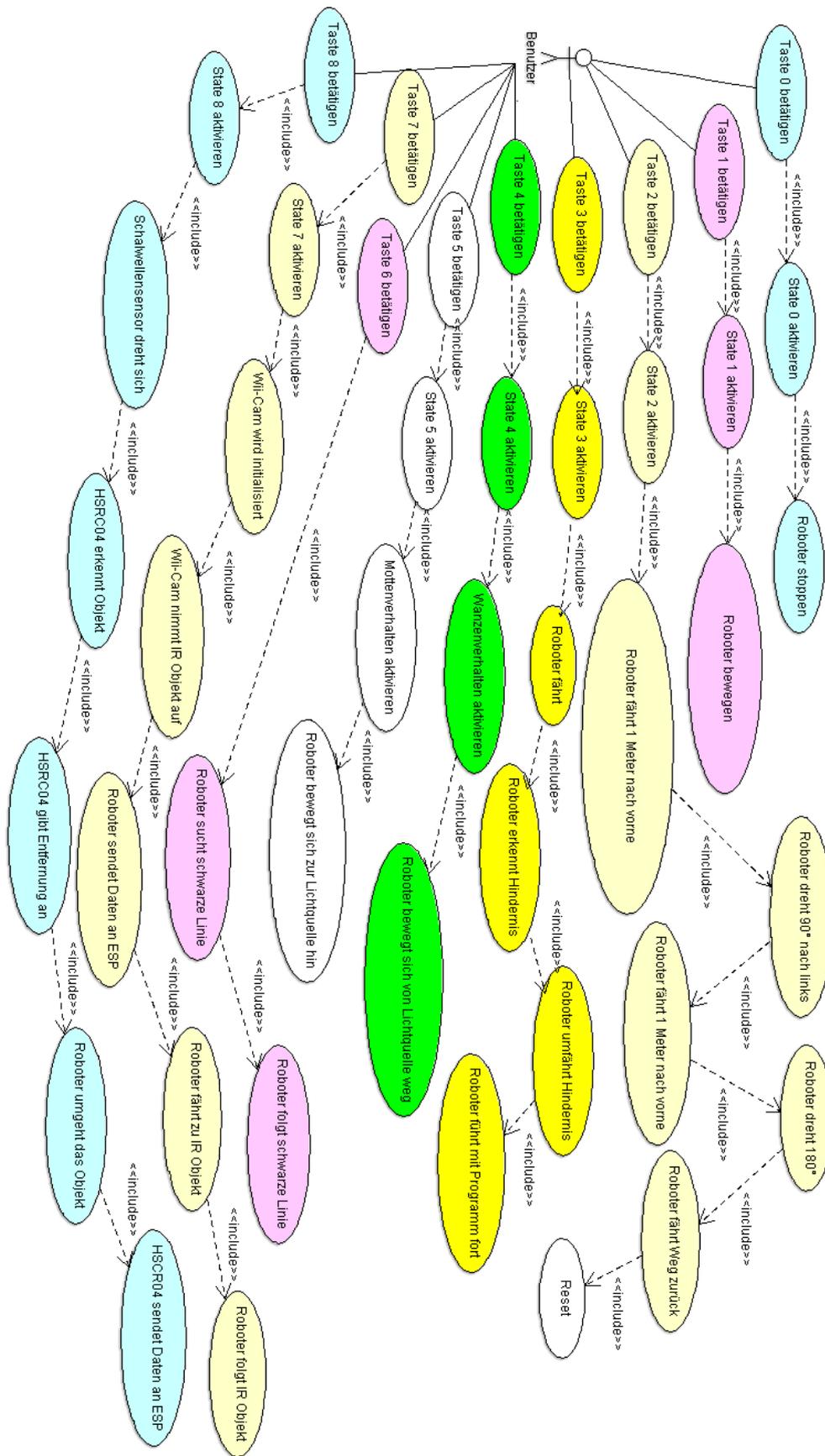


Abbildung 50: Anwendungsfalldiagramm

6.4. Strukturgramm

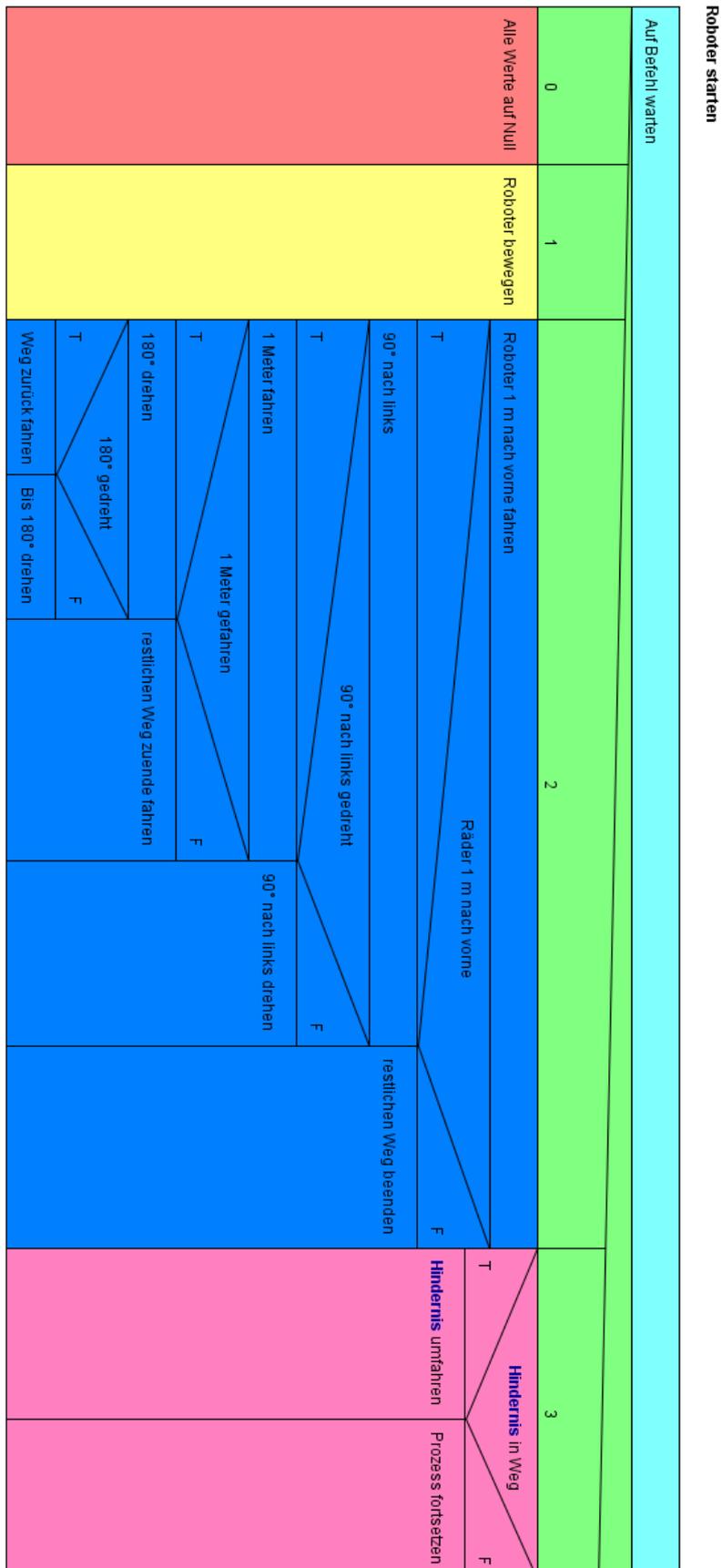


Abbildung 51: Strukturgramm

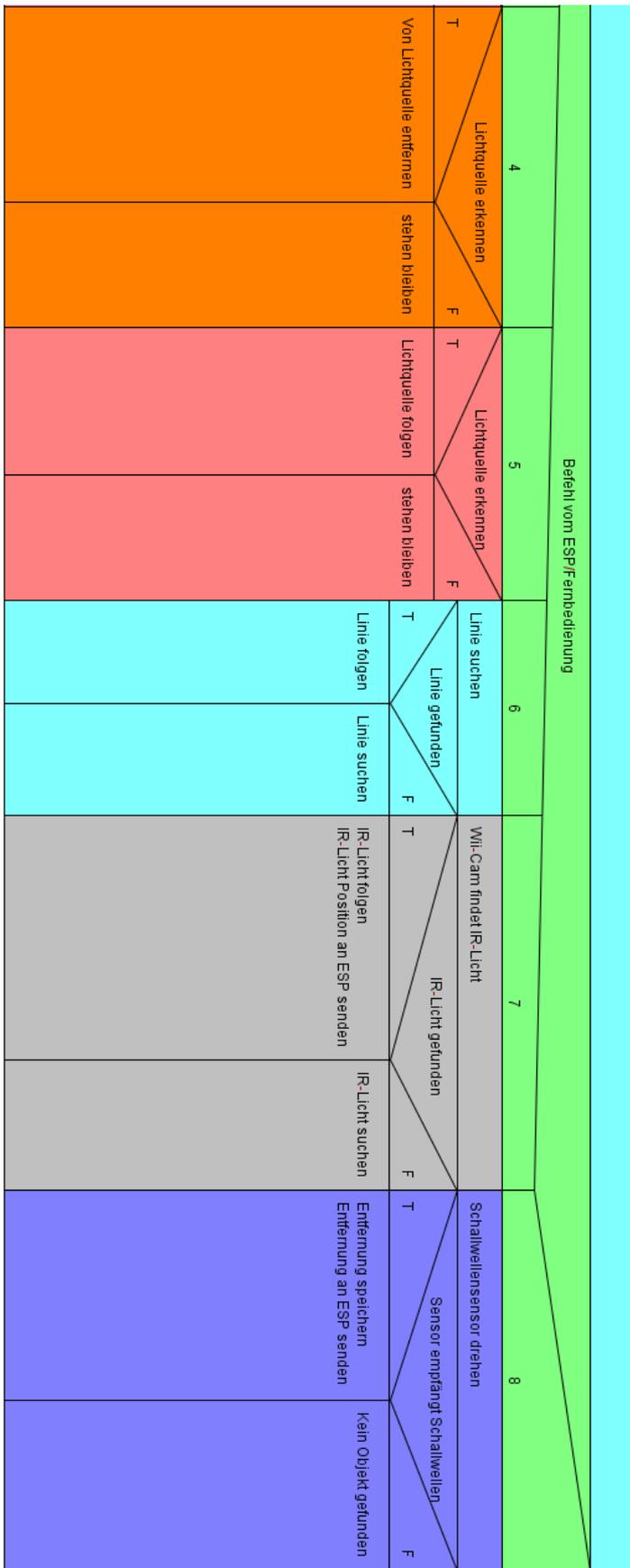


Abbildung 52: Strukturgramm

7. Teamsitzungen

Als Gruppe haben wir uns jede Woche nach der Schule zusammengesetzt, um über unseren Stand gegenseitig zu informieren oder abends per Skype informiert. Falls Fragen aufgetaucht sind, wurden diesen in den Teamsitzungen geklärt.

Bei jeder Teamsitzung wurde ein Protokoll mitgeführt:

- Zeit/Ort
- An/-Abwesende
- Ziele
- Verlauf
- Ergebnisse
- Besonderheiten

Es befinden sich im nachfolgenden insgesamt fünf Teamsitzungsprotokolle welche vom Teamleiter unterzeichnet sind. Die Teamsitzungsprotokolle umfasst die erste Projektwoche. Der ganze Aufbau ist in den Teamsitzungen detailliert beschrieben.

7.1. Teamsitzung 1

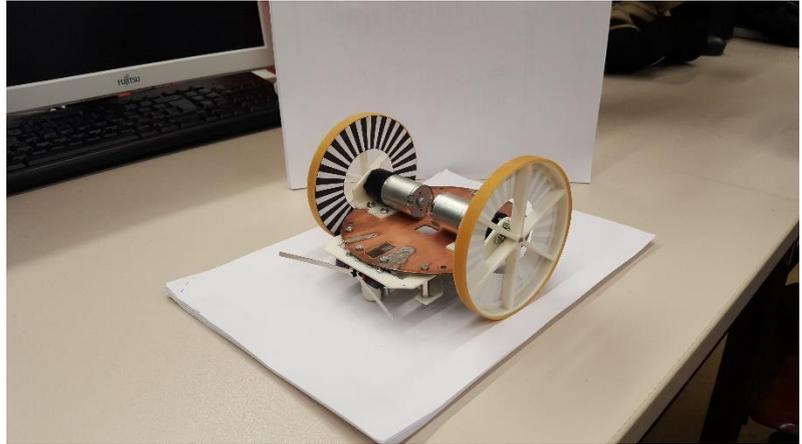
Zeit/Ort:

13. November 2017

08:00 Uhr-15:00 Uhr

Raum 301

Berufskolleg Technik Remscheid



An-/Abwesende:

- Keiner Abwesend

Ziele:

- Leitungen für die Kommunikation zwischen Komponente erstellen
- Untere Ebene des Roboters anfertigen

Verlauf:

Am ersten Tag haben wir die Dokumentationen für das Roboterprojekt ausgeteilt bekommen. Wir haben uns die Materialien (Bauteile) besorgt und angefangen den Roboter zu bauen. Serdar und Benjamin haben die untere Ebene (Antriebsebene) fertig gebaut. Florian und Mehmet haben die Leitungen verzinnt und angefangen die Leitungen mit den Steckern zu löten.

Ergebnisse:

- Montage des Motorhalters
- Montieren des Kugellagers
- Encoder- Winkelscheibe und Räder
- Montieren des Motors mit Rad
- Montage der Halterung für Schalter und Liniensensor von unten

Besonderheiten:

- Keine Besonderheiten

7.2. Teamsitzung 2

Zeit/Ort:

14. November 2017

08:00 Uhr-15:00 Uhr

Raum 301

Berufskolleg Technik Remscheid

An-/Abwesende:

- Keiner Abwesend

Ziele:

- Liniensensor Platine fertig stellen
- Wii Cam Platine fertig stellen
- Motortreiber Platine fertig stellen
- IR-Sensor Platine fertig stellen

Verlauf:

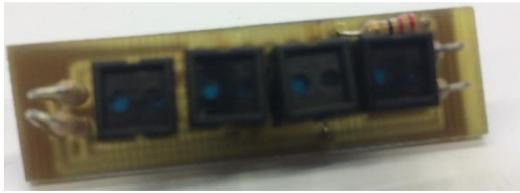
Am zweiten Tag haben alle mit der Erstellung der Platinen angefangen. Mehmet und Serdar haben die Wii Cam und den Liniensensor fertig gestellt. Florian hat mit dem IR-Sensor angefangen aber wurde nicht fertig. Serdar hat mit dem Motortreiber angefangen, doch er hat ein Fehler gemacht und er wusste nicht wie er es beheben sollte. Benjamin hat weiter an dem Motortreiber gearbeitet aber wurde nicht fertig.

Ergebnisse:

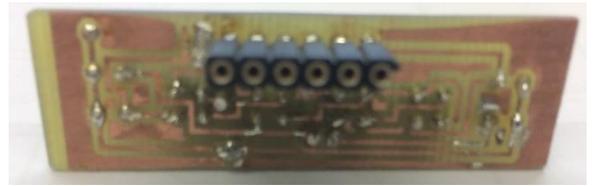
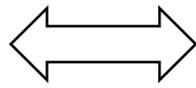
- Liniensensor Platine fertig gestellt
- Wii Cam Platine fertig gestellt

Besonderheiten:

- Keine Besonderheiten

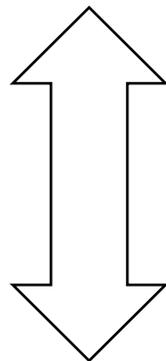
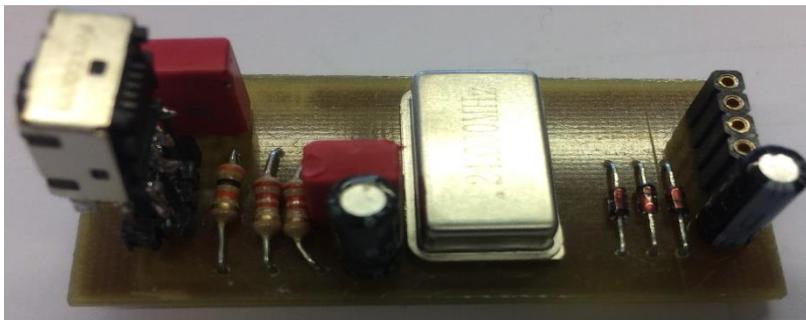


Liniensensor Platine
Oberhalb



Liniensensor Platine
Unterhalb

Wii Cam Platine
Oberhalb



Wii Cam Platine
Unterhalb

7.3. Teamsitzung 3

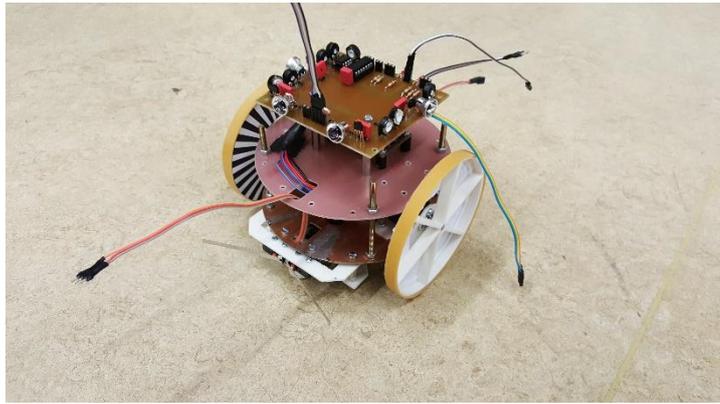
Zeit/Ort:

15. November 2017

08:00 Uhr-15:00 Uhr

Raum 301

Berufskolleg Technik Remscheid



An-/Abwesende:

- Keiner Abwesend

Ziele:

- Motortreiber Platine fertig stellen
- IR-Sensor Platine fertig stellen
- Alle Leitungen messen
- Kalte Lötstellen verbessern

Verlauf:

Am dritten Tag wurden auch die Motortreiberplatine und die IR-Sensorplatine fertig gestellt. Alle Platinen wurden nach kalte Lötstellen untersucht und verbessert. Mit einem Messgerät haben wir die Leitungen überprüft und haben festgestellt das alles in Ordnung war. Danach haben wir bis zum Rest des Tages die Hauptebene fertig gestellt.

Ergebnisse:

- Motortreiber Platine fertig gestellt
- IR-Sensor Platine fertig gestellt
- Alle Leitungen erfolgreich gemessen
- Kalte Lötstellen verbessert

Besonderheiten:

- Keine Besonderheiten

7.4. Teamsitzung 4

Zeit/Ort:

16. November 2017

08:00 Uhr-15:00 Uhr

Raum 301

Berufskolleg Technik Remscheid

An-/Abwesende:

- Keiner Abwesend

Ziele:

- Den Kopf anbringen (Schallwellensensor, IR-Tower etc.)
- Kabel Verlegung
- Restliche Platinen an Roboter befestigen
- Testprogramme erstellen

Verlauf:

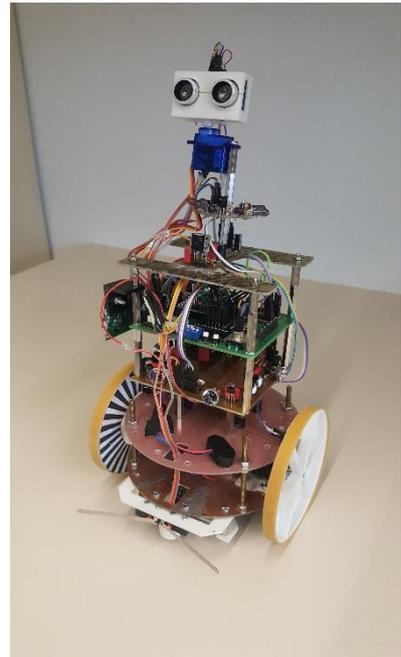
Am vierten Tag war der Roboter komplett fertig. Wir haben alle restlichen Platinen an den Roboter befestigt. Den IR-Tower und Schallwellensensor haben wir auch als letztes montiert und befestigt. Die letzten Kabel haben wir mit den jeweils zugehörigen Ports und Pins verbunden. Den Rest der Zeit haben wir alle außer Serdar Testprogramme geschrieben. Serdar hat als Gruppenleiter sein Zeitplan bearbeitet.

Ergebnisse:

- Schallwellensensor & IR-Tower anmontiert
- Platinen an Roboter befestigt
- Kabel verlegt

Besonderheiten:

- Bumper links funktioniert / rechts Fehler
 - o Lötstelle wurde entdeckt und behoben
- Rechter Rad (Servo) funktioniert nicht



7.5. Teamsitzung 5

Zeit/Ort:

17. November 2017

08:00 Uhr-15:00 Uhr

Raum 301

Berufskolleg Technik Remscheid

An-/Abwesende:

- Keiner Abwesend

Ziele:

- Testprogramme für alle Funktionen erstellen
- Fehler der rechten Servo beseitigen

Verlauf:

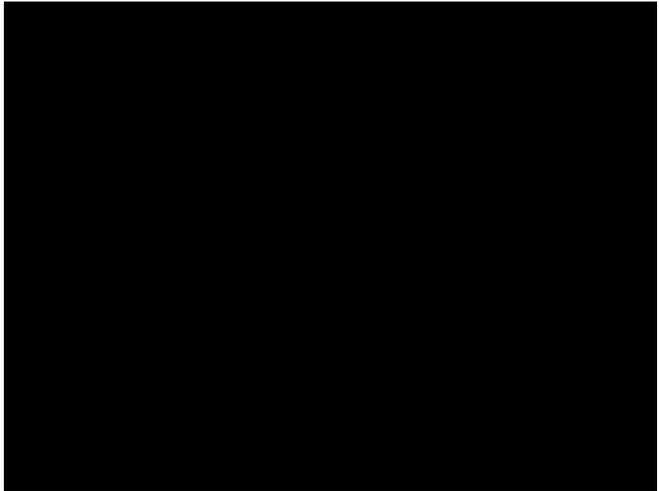
Am fünften Tag haben wir alle Testprogramme fertig gestellt. Die Testprogramme haben wir am Roboter getestet und alle haben auch funktioniert. Wir haben den Fehler beim rechten Rad behoben und den Testprogramm dafür auch fertig gestellt.

Ergebnisse:

- Testprogramme erfolgreich fertig gestellt
- Fehler der rechten Servo behoben

Besonderheiten:

- Für die tolle Arbeit haben wir uns am Ende belohnt (siehe Bild)



8. Zeitplan

Der Zeitplan zeigt den kompletten Arbeitsprozess nochmal in einem Schaubild wieder. Der Zeitplan wurde vom Teamleiter mit dem Programm "Excel" erstellt.

Der Zeitplan ist in 3 Teile aufgeteilt. Im ersten Teil sind die Aufgaben aufgelistet. Im mittleren Teil sind die Teammitglieder, die für die Aufgaben zuständig waren aufgelistet. Im letzten Teil sind die einzelnen Zeitabläufe für die jeweiligen Aufgaben dargestellt.

Die Legende:

Legende	
Mehmet	
Bejamin	
Florian	
Serdar	
Serdar & Bejamin	
Mehmet & Florian	
Serdar & Mehmet	
Bejamin & Mehmet	
Bejamin & Florian	
Serdar & Florian	
Alle	

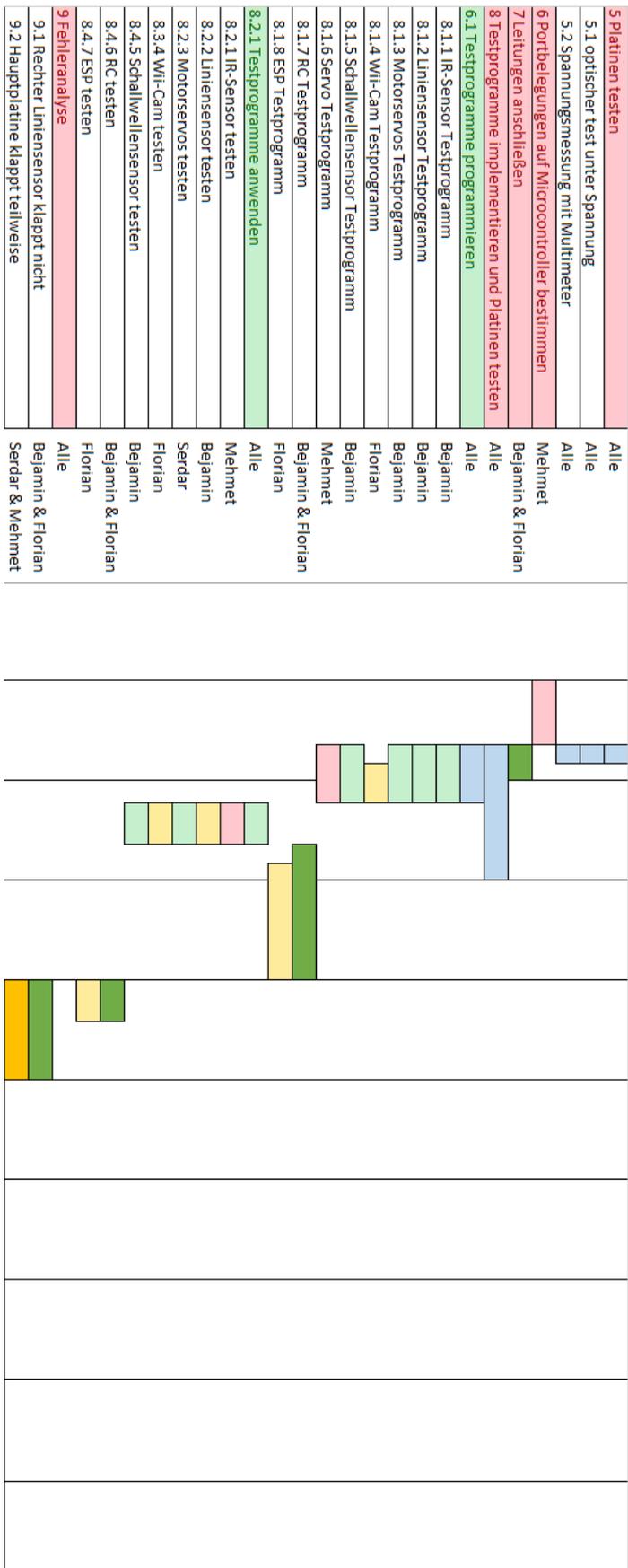


Abbildung 54: Zeitplan

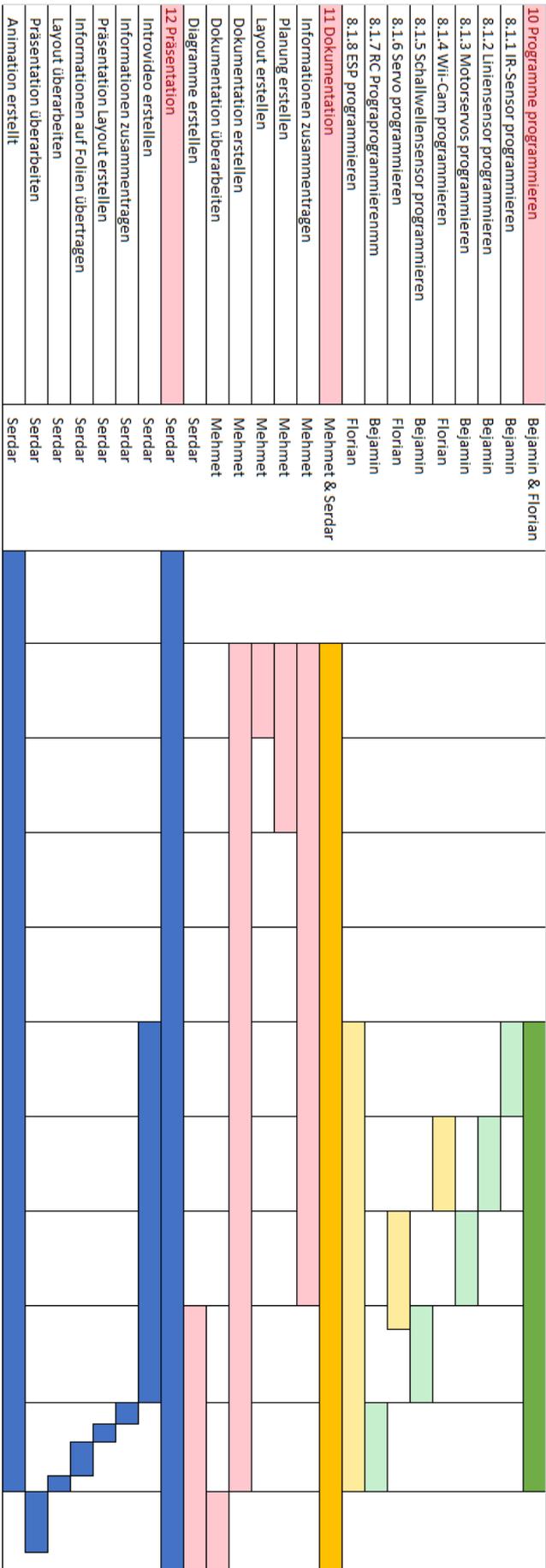


Abbildung 55: Zeitplan

9. Gruppenfazit

Als Gruppe war das Projekt eine interessante Erfahrung, da wir viele Erfahrung ausgetauscht und neue gewonnen haben. Dadurch hat das Projekt viele Wissenslücken geschlossen und neue Kenntnisse erbracht.

Die Gruppenkonstellation war im guten Verhältnis und die Gruppe hat gut miteinander harmoniert. Jedes Gruppenmitglied konnte seine Stärken in dem Projekt zeigen.

Wir hatten alle unsere Höhen und Tiefen aber im Großen und Ganzen hat keiner seine Motivation verloren. Alle hatten das Ziel das Projekt erfolgreich zu beenden und jeder hat versucht darauf hinzuarbeiten.

Dadurch, dass, wir uns oft getroffen und Teamsitzungen gehalten haben, hatten wir einen guten Kommunikationsfluss und keiner blieb ohne Beschäftigung. Es war für jeden etwas zu tun und wenn einer mal Probleme hatte konnte ein anderes Teammitglied aushelfen. Ein weiteres Werkzeug was uns bei der Vervollständigung des Projekts geholfen hat, war die Zusammentragung der erledigten Dokumente, Programme, Diagramme, etc. auf einer Cloud. Dadurch war jeder auf dem neusten Stand und konnte mit der richtigen Information weiterarbeiten.

Da in dem Projekt Soft- auf Hardware getroffen ist, gab es viele Problem Möglichkeiten. Die meisten Probleme die wir während des Projekts hatten, hatte mit der Hardware zutun. Oft war es so dass, ein Komponent welches funktionierte sich verabschiedete und wir mehre Stunden opfern mussten um den Fehler zu finden. Letztendlich schauen acht besser als zwei Augen, daher konnten wir im Team auch diese Probleme lösen.

Es war eine gute Option, dass, wir in der Schule nach und während dem Unterricht an dem Roboter weiter basteln konnten, dadurch hatten wir eine gute Arbeitsatmosphäre bzw. einen Arbeitsplatz und man konnte sich mit anderen Gruppen austauschen. Da unsere Gruppe seit Tag eins an dem Projekt gearbeitet hat, sind wir schnell vorangekommen. Oft war es aber trotzdem so, dass wir Programme oder Kommunikationsschnittstellen nicht testen konnten. Obwohl wir oft diese Möglichkeit hatten in der Schule arbeiten zu können, wäre es besser, wenn, man in der Schule einen Raum öfters zur Verfügung hätten, so könnten man sich auch nach der Schule mit den Teammitgliedern treffen und effizienter arbeiten. Natürlich hat nicht jeder ein Lötkolben und das passende Material Zuhause, daher war es oft ein Problem das man über das Wochenende eine Leitung, die abgegangen war nicht einfach wieder anlöten konnte. Dies hat uns in Verzug gebracht. Mit einem Raum in der Schule würden sich diese Probleme lösen, da alle benötigten Geräte, Werkzeuge und Materialien in der Schule vorhanden sind.

Ein weiterer guter Aspekt ist, dass, wir einen hohen Anteil an praktischer Arbeit hatten. Das bauen des Roboters hat jedem Teammitglied viel Spaß gemacht. Auch wenn manche Sachen wie Platine schneiden oder das anlöten eines sehr kleinen Komponenten schwierig waren hat im Großen und Ganzen der praktische Teil ein hohen Spaß Faktor.

Natürlich gab es auch Probleme beim programmieren, da die Hardware nicht immer so mitgespielt hat wie wir es uns gedacht hatten, aber nach bisschen testen und basteln haben sich auch diese Probleme gelöst.

Am meisten haben wir Zeit zum suchen der Hardware Technischen Probleme gebraucht. Wir mussten die Schaltpläne durchgehen und einzelne Lötstellen oder Komponenten testen.

Für uns war das Projekt sehr erfolgreich und wir konnten es mit vielen neuen Kenntnissen und Erfahrungen abschließen.

10. Abbildungsverzeichnis

Abbildung 1:Roboter	4
Abbildung 2:Die Aufgabendarstellung als Schaubild (Quelle: projekt roboter_3d 2017 Funkmodul_IOT.pdf)	6
Abbildung 3: IR Tower mit RC5-Empfänger.....	10
Abbildung 4: Schaltplan des IR-Towers.....	10
Abbildung 5: Bestückungsliste des IR-Towers (Quelleprojekt roboter_3d 2017 Funkmodul_IOT.pdf:)	11
Abbildung 6: Makros RC5	12
Abbildung 7: RC5 Code	12
Abbildung 8: RC5 Code	12
Abbildung 9: Wii Cam Platine	14
Abbildung 10: Schaltplan der Wii Cam Platine (Quelle: projekt roboter_3d 2017 Funkmodul_IOT.pdf)	14
Abbildung 11: Bestückungsliste der Wii Cam (Quelle: projekt roboter_3d 2017 Funkmodul_IOT.pdf)	15
Abbildung 12: Wii Cam Code.....	16
Abbildung 13: Hauptplatine	17
Abbildung 14: Animation.....	17
Abbildung 15: Schaltplan der Hauptplatine (Quelle: projekt roboter_3d 2017 Funkmodul_IOT.pdf)	18
Abbildung 16: Bestückungsliste der Hauptplatine (Quelle: projekt roboter_3d 2017 Funkmodul_IOT.pdf)	20
Abbildung 17: Wanzenverhalten-Code	21
Abbildung 18: Mottenverhalten- Code	21
Abbildung 19: Motosteereinheit.....	22
Abbildung 20: Schaltplan der Motorsteereinheit (Quelle: projekt roboter_3d 2017 Funkmodul_IOT.pdf)	22
Abbildung 21: IC L293D (Quelle: projekt roboter_3d 2017 Funkmodul_IOT.pdf).....	23
Abbildung 22: H-Brücke (Quelle: http://www.strippenstrolch.de/1-2-9-motortreiber-298.html)	23
Abbildung 23: Bestückungsliste der Motorsteereinheit (Quelle: projekt roboter_3d 2017 Funkmodul_IOT.pdf)	24
Abbildung 24: Aktorik-Code	25
Abbildung 25: Liniensensor	27
Abbildung 26: Schaltplan der Liniensensor (Quelle : projekt roboter_3d 2017 Funkmodul_IOT.pdf)	27
Abbildung 27: Bestückungsliste der Liniensensor (Quelle: projekt roboter_3d 2017 Funkmodul_IOT.pdf)	28
Abbildung 28: Liniensensor Code.....	29
Abbildung 29: Schallwellensensor.....	30
Abbildung 30: Anschlusspins der Schallwellensensor (Quelle: HC- SR04_ultraschallmodul_beschreibung_3.pdf)	30

Abbildung 31: Messung der Objekte (Quelle: HC-SR04_ultraschallmodul_beschreibung_3.pdf)	31
Abbildung 32: Radencoder	33
Abbildung 33: Bumper	33
Abbildung 34: Mega128 (Quelle: projekt roboter_3d 2017 Funkmodul_IOT.pdf)	34
Abbildung 35: Portbelegung auf dem Microcontroller (Quelle: projekt roboter_3d 2017 Funkmodul_IOT.pdf)	36
Abbildung 36: Microcontroller (Quelle: projekt roboter_3d 2017 Funkmodul_IOT.pdf).....	36
Abbildung 37: Makros der Portbelegungen	37
Abbildung 38: ESP8266	38
Abbildung 39: Schnittstelle & Signalaustausch (Quelle: ESP8266_I2C_MEGA128.PDF)	39
Abbildung 41: LCD Initialisierung	40
Abbildung 42: Beispiel Code.....	40
Abbildung 43: LCD Ausgabe	40
Abbildung 45: IP Initialisierung	43
Abbildung 46: Websocket & Webserver mit HTML Code wird gestartet	43
Abbildung 47: Website sendet zur ESP8266	44
Abbildung 48: Transaktion	44
Abbildung 49: ESP826 sendet an Website	45
Abbildung 50: Sequenzdiagramm	47
Abbildung 51: Aktivitätsdiagramm.....	48
Abbildung 52: Anwendungsfalldiagramm	49
Abbildung 53: Strukturgramm.....	50
Abbildung 54: Strukturgramm.....	51
Abbildung 55: Zeitplan	60
Abbildung 56: Zeitplan	61
Abbildung 57: Zeitplan	62

11. Literaturverzeichnis

6_Finite State Machine.pdf

ESP8266_I2C_MEGA128.PDF

ESP8266_WiFi_Module_Quick_Start_Guide_v_1.0.4.pdf

HC-SR04_ultraschallmodul_beschreibung_3.pdf

projekt roboter_3d 2017 Funkmodul_IOT.pdf

Vorarbeiten für Roboterprojekt 2017.pdf

12. Weitere Informationen

Der Gerät

© 2018 Der Gerät, Alle Rechte vorbehalten

Sämtliche Inhalte, Fotos, Texte, Grafiken sind urheberrechtlich geschützt. Sie dürfen ohne vorherige schriftliche Genehmigungen weder ganz noch auszugsweise kopiert, verändert, vervielfältigt oder veröffentlicht werden.



Die Online Version des kompletten Projekts mit ATmega und ESP Code finden Sie unter folgenden Link: <https://goo.gl/SZJTxB>

(Öffnen sie den Link mit Firefox!)