

Dokumentation Arduino Car

Julia Petz

28 March 2018

Contents

1 Einleitung	1
2 Planung	1
2.1 Skizze des Autos	1
2.2 Schaltplan	3
3 Bauen des Autos	3
4 Testen der Einzelteile	5
4.1 Servo	5
4.2 Ultraschallsensor	6
4.3 DC-Motoren	7
5 Links	8
5.1 Tutorials	8
5.2 Foren	8

1 Einleitung

In diesem Dokument wird die Arbeit (Planen, Bauen und Programmieren) eines selbstfahrenden Autos mithilfe eines Arduino Uno dokumentiert.

- [pdf-Datei](#)

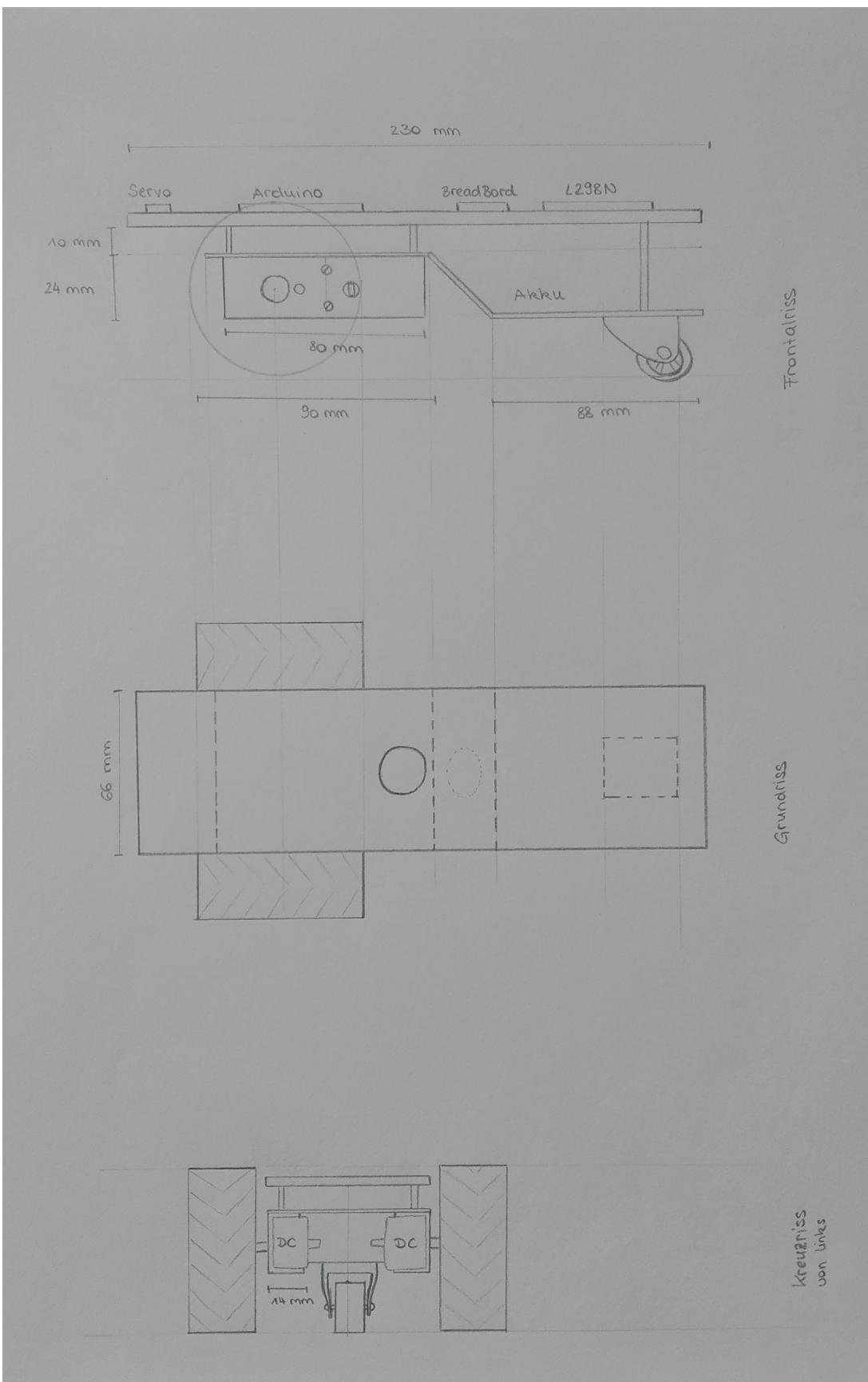
2 Planung

Die Planung erfolgte aufgrund der verfügbaren Bauteile:

- Arduino Uno
- Ultraschallsensor
- Servo
- 2 Farbsensoren
- 2 Abstandssensoren
- 2 DC-Motoren inklusive Rad
- Motorschutz
- Taster

2.1 Skizze des Autos

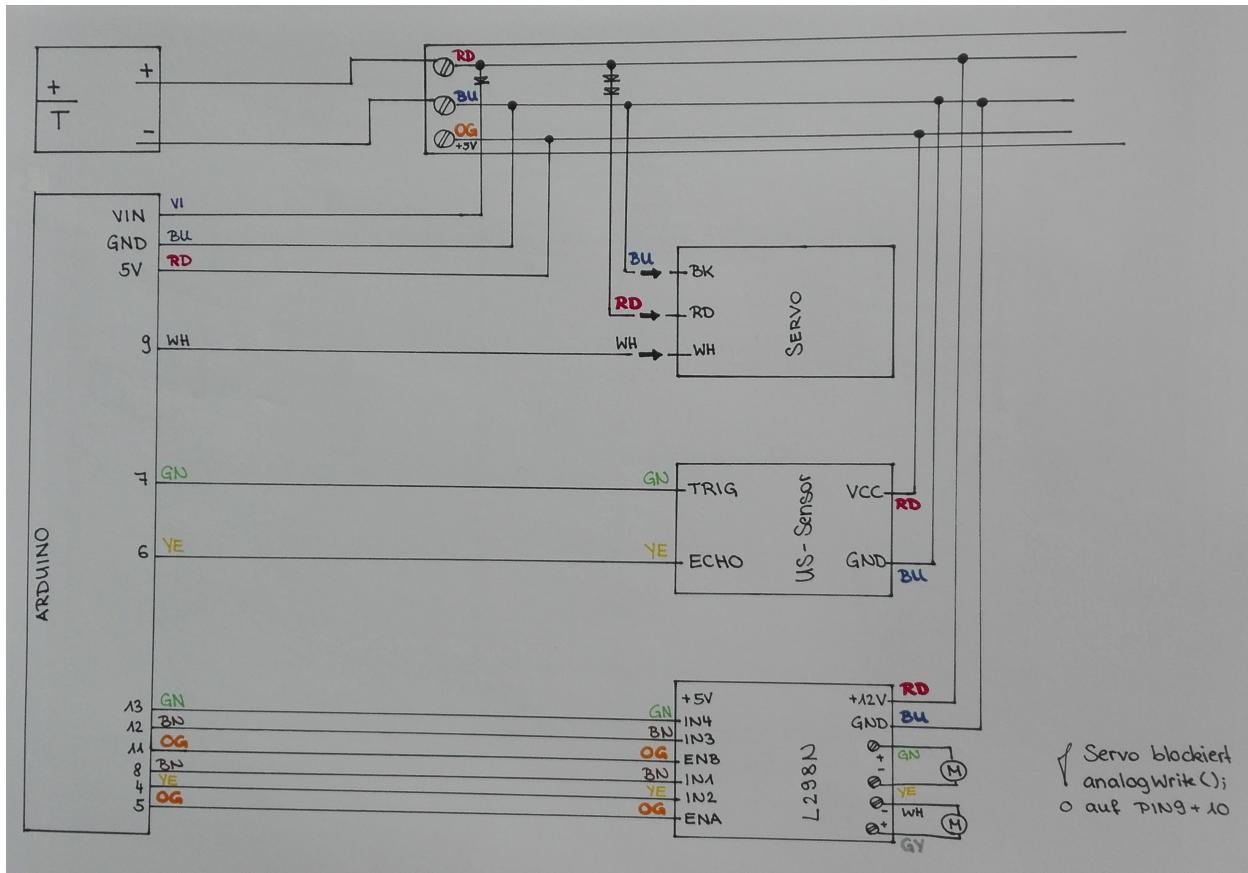
Die folgende Skizze beinhaltet eine maßstäbliche Grobplanung des Aufbaus des selbstfahrenden Autos. Momentan wird dabei nur die Planung eines Autos mit Servo und Ultraschallsensor behandelt und kein Line-Follower.



2.2 Schaltplan

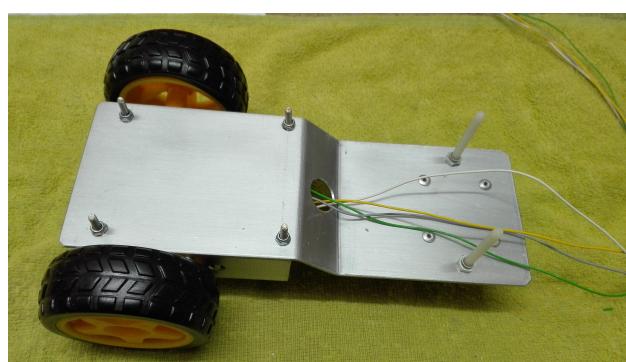
Der momentane ersichtliche Schaltplan beinhaltet die Verkabelung für ein selbstfahrendes Auto mit Ultraschallsensor und Servo. Weiters wird hier ein Akku als Stromquelle angenommen.

Wichtig: Wenn ein Servo verwendet wird, verhindert dieser ein `analogWrite()`; auf Pin 9 und Pin 10. Jedoch kann das Servo sehr wohl an diesen Pins angeschlossen werden.

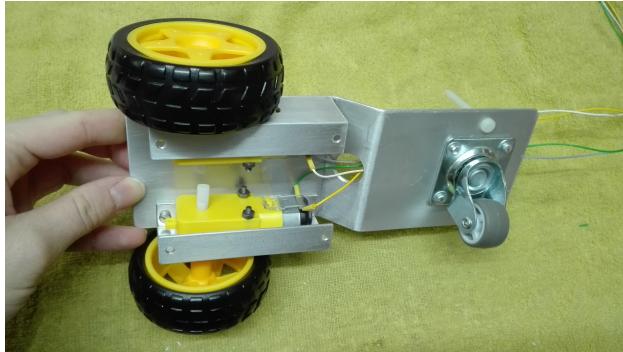


3 Bauen des Autos

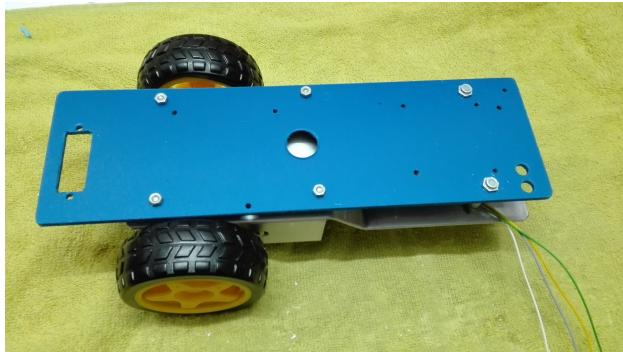
Die Bodenplatte besteht aus Aluminium und wurde "gekantet" (sprich gebogen). Zusätzlich wurde die Platte noch mit Klarlack lackiert, sodass das schwarze Abfärbren des Aluminiums verhindert wird.
Die Aufhängung der DC-Motoren besteht aus einem zugeschnittenen Aluminiumprofil.



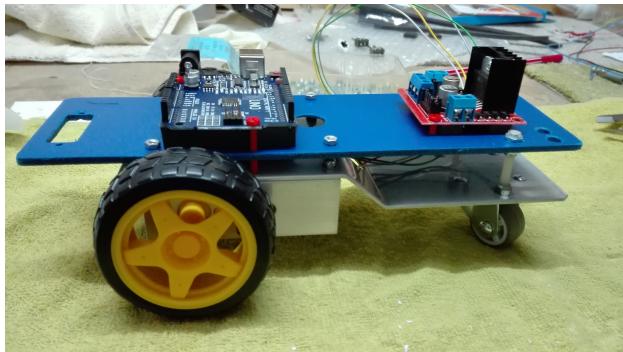
Im oberen Bild sieht man den Unterboden (inlusiver montierter DC-Motoren) von oben. Im folgenden Bild ist die Konstruktion von unten zu sehen.



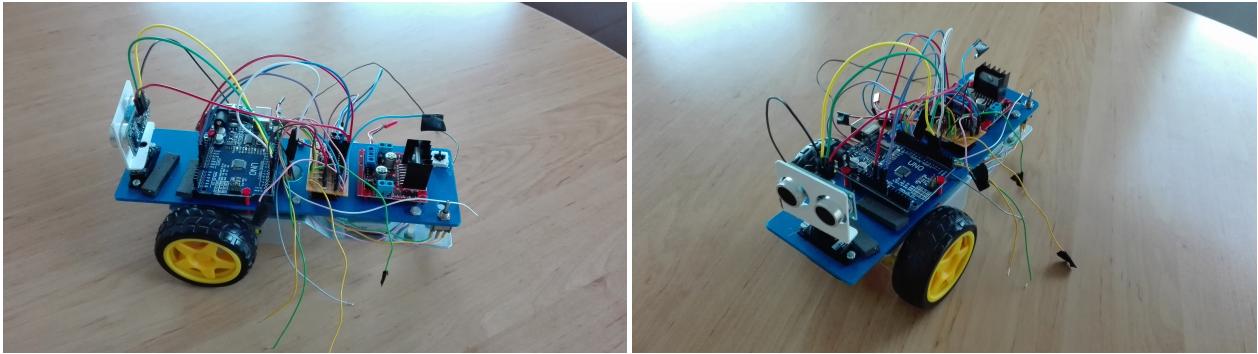
Mithilfe von Schrauben und Muttern wurde die blaue Bodenplatte auf die gewünschte Höhe gebracht und fixiert. Dies ist auf dem folgenden Bild ersichtlich.



Der Arduino Uno und der Motortreiber wurden mithilfe von Plastikschauben auf der blauen Bodenplatte fixiert. Dafür wurden die bereits vorgebohrten Löcher verwendet, welche nicht im Plan aufscheinen.



Da selbstgebastelte und gelötete Bread-Bord wurde mithilfe eines Klettstreifens auf der blauen Bodenplatte befestigt. Für das Servo wurde ein passendes Loch in die Platte geschnitten. Weiters wurde aus einer Plastikschiene noch eine Halterung für den Ultraschallsensor gebastelt. Danach erfolgte die Verkabelung laut dem Schaltplan.



4 Testen der Einzelteile

Momentan scheinen hier die Testcodes für ein Servo, einen Ultraschallsensor und DC-Motoren (unter der Verwendung eines Motorschutzes) auf. Weiterführende Codes stehen noch nicht zur Verfügung.

4.1 Servo

Mithilfe dieses Beispielcodes wurde der Servo getestet:

```
// Einbinden der Servo.h Library für die Steuerung des Servos
#include<Servo.h>

// Pinbelegung des Servos und des Tasters am Arduino
int servoPin = 5;
int tasterPin = 4;

// Servo-Instanz erstellen
Servo scanServo;

void setup() {
    // Bekanntmachung der Pins (Input oder Output)
    scanServo.attach(servoPin,744,2200);
    pinMode(tasterPin, INPUT);
    Serial.begin(9600);
}

void loop() {
    // Abfragen des Tasters
    if(digitalRead(tasterPin)){
        servoMove();
    }
}

// Funktion um den Servo zu bewegen
void servoMove(){
    // Servo wird auf Position mit Grad 0 bewegt
    scanServo.write(0);
    Serial.println("Servo links");
    delay(2000);
}
```

```

// Servo wird auf Position mit Grad 180 bewegt
scanServo.write(180);
Serial.println("Servo rechts");
delay(2000);
// Servo wird auf Position mit Grad 90 bewegt
scanServo.write(90);
Serial.println("Servo mitte");
delay(2000);
}

```

Sobald ein Servo an den Arduino angeschlossen wird, wird ein `analogWrite()`; auf den *PWM-Pins 9 und 10* verhindert. Ein Servo kann an diesen Pins ohne Probleme laufen, jedoch wird beispielsweise kein DC-Motor an diesen Pins funktionieren.

4.2 Ultraschallsensor

Mit dem folgenden Code wurde der Sensor - noch unabhängig vom Servo - getestet.

```

// Pinbelegung des Sensors am Arduino
int trigger=7;
int echo=6;
int timeUS=0;
int distance=0;

void setup() {
    Serial.begin(9600);
    // Bekanntmachung ob Input oder Output vorliegt
    pinMode(trigger, OUTPUT);
    pinMode(echo, INPUT);
}

void loop() {
    // Spannung vom Trigger nehmen, damit später ein
    // rauschfreies Signal vorhanden ist
    digitalWrite(trigger, LOW);
    delay(5);
    // Ultraschallwelle wird losgeschickt
    digitalWrite(trigger, HIGH);
    delay(10);
    // "Welle" endet, indem Spannung entfernt wird
    digitalWrite(trigger, LOW);
    // pulseIn - Mikrocontroller zählt Millisekunden bis
    // das Echo am Sensor angelangt ist
    timeUS = pulseIn(echo, HIGH);
    // Berechnung der Entfernung in Zentimeter
    // (es wird dabei mit der Schallgeschwindigkeit multipliziert)
    distance = (timeUS/2) * 0.03432;
    // Ausgabe der Distanz am seriellen Monitor
    Serial.println(distance);
}

```

4.3 DC-Motoren

```
// Pinbelegung der Motoren auf dem Arduino
// Motor 1 (grün-gelb)
int enA = 10;
int in1 = 9;
int in2 = 8;
// Motor 2 (weiß-grau)
int enB = 11;
int in3 = 12;
int in4 = 13;

int tasterPin = 4;

void setup(){
    // alle Motoren-Pins als Output setzen
    pinMode(enA, OUTPUT);
    pinMode(enB, OUTPUT);
    pinMode(in1, OUTPUT);
    pinMode(in2, OUTPUT);
    pinMode(in3, OUTPUT);
    pinMode(in4, OUTPUT);
    pinMode(tasterPin, INPUT);
}

void demoOne(){
    // die Funktion lässt beide Räder zuerst vorwärts und
    // dann rückwärts laufen
    // Motor 1 einschalten
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    // Geschwindigkeit wird auf 200 gesetzt
    // mögliche Werte sind aus dem Intervall [0;255]
    analogWrite(enA, 200);
    // Motor 2 einschalten
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    // Geschwindigkeit wird auf 200 gesetzt
    analogWrite(enB, 200);

    delay(2000);

    // Richtungswechsel
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);

    delay(2000);

    // Motoren werden ausgeschaltet
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
}
```

```

digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
}

void loop(){
  if(digitalRead(tasterPin)){
    demoOne();
    delay(3000);
  }
}

```

Im nächsten Schritt werden dann die unterschiedlichen “Einzelgeräte” verbunden. Dieser Code - sowie der fertige Code für das selbstfahrende Auto - scheinen jedoch noch nicht auf.

5 Links

5.1 Tutorials

- Funduino
- Arduino Tutorial
- tutorialspoint
- Madgyver
- Arduinospielwiese

5.2 Foren

- Arduino Forum