Inhalt:

1. Erklärung

Das Arduino Fahrzeug besitzt 3 Ultraschall Hc-SR-04 Sensoren, welche vorne und hinten angebracht sind, und somit ein sehr großes Sichtfeld des Fahrzeuges ermöglichen. Diese Sensoren führen ca. 9 Messungen pro Sekunde aus um die Bewegung von sich schnell bewegenden Objekte optimal zu erfassen. Zwischen den Messungen überprüft das Fahrzeug, ob sich Objekte in der geplanten Fahrstrecke befinden. Wenn ein Objekt sich in einer Distanz von 15 bis 30 cm Entfernung befindet, wird automatisch ein Ausweichmanöver in eine andere Richtung gestartet. Wenn sich ein Objekt in einer Reichweite von unter 15 cm befindet, versucht das Fahrzeug nach hinten auszuweichen. Sollte sich aber hinter dem Fahrzeug keine 15cm Platz befinden, so bleibt das Fahrzeug stehen, und muss durch einen Knopfdruck wieder entsichert werden. Die Status LED auf dem Fahrzeug, zeigt zudem den Status des Fahrzeuges an. Ist diese grün, so bedeutet dies, dass das Fahrzeug ohne Probleme funktioniert. Ist die Status LED rot bedeutet dies, dass das Fahrzeug entweder keinen Ausweg aus einer Situation gefunden hat, oder dass es durch Knopfdruck in den sicheren Modus gesetzt wurde.

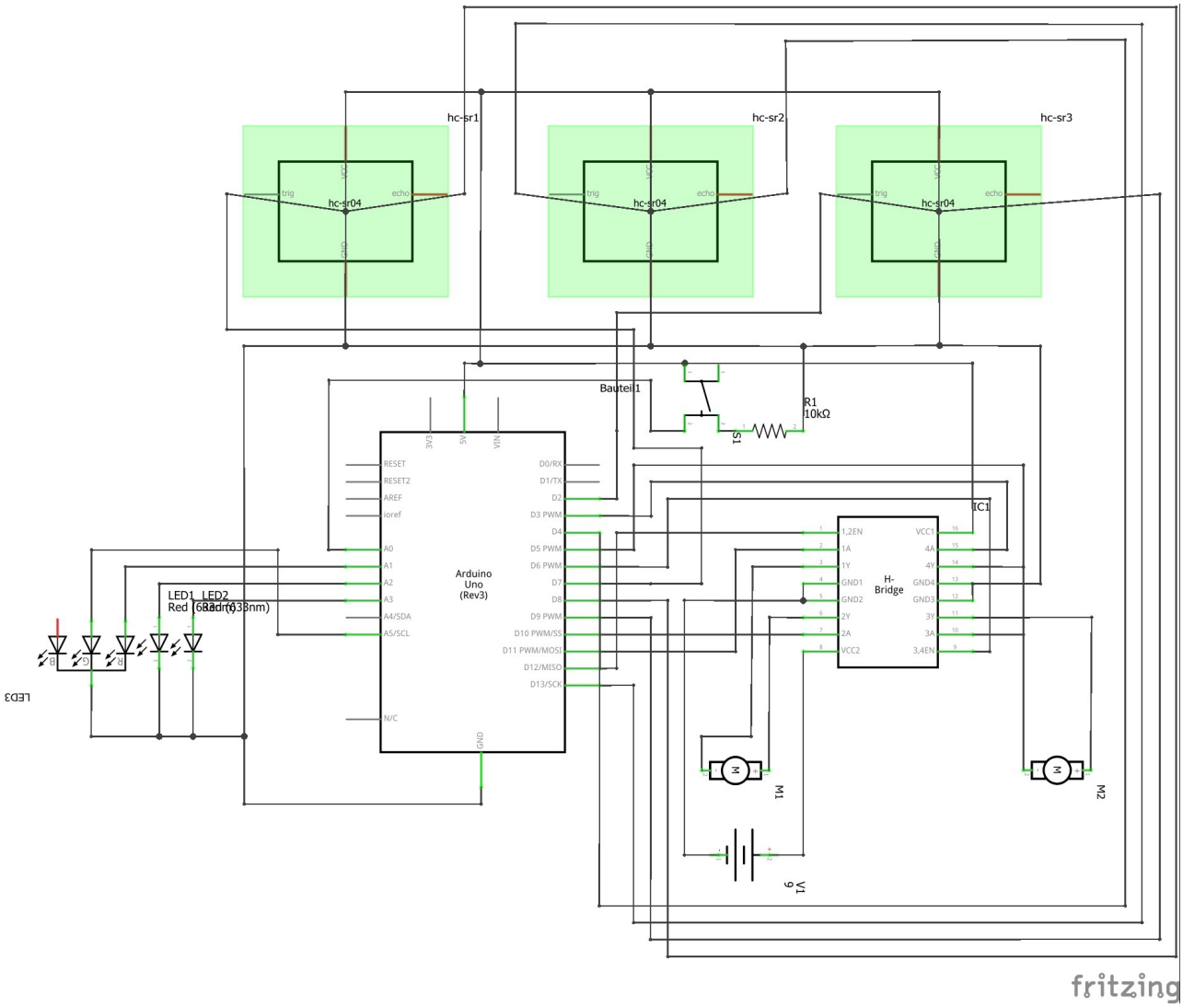
2. Teileliste

Anzahl	Bezeichnung		Anzahl	Bezeichnung
3	HC-SR04		1	RGB Led (gemeinsam Minus)
2	DC Motor		1	9V Clip
1	H-Brücke		1	9V Block
1	Taster		1	Breadboard
1	10kΩ Widerstand		4	Schrauben M4
1	Arduino Uno		12	Muttern M4
2	LED Orange		1	Holzplatte 90 x160 mm

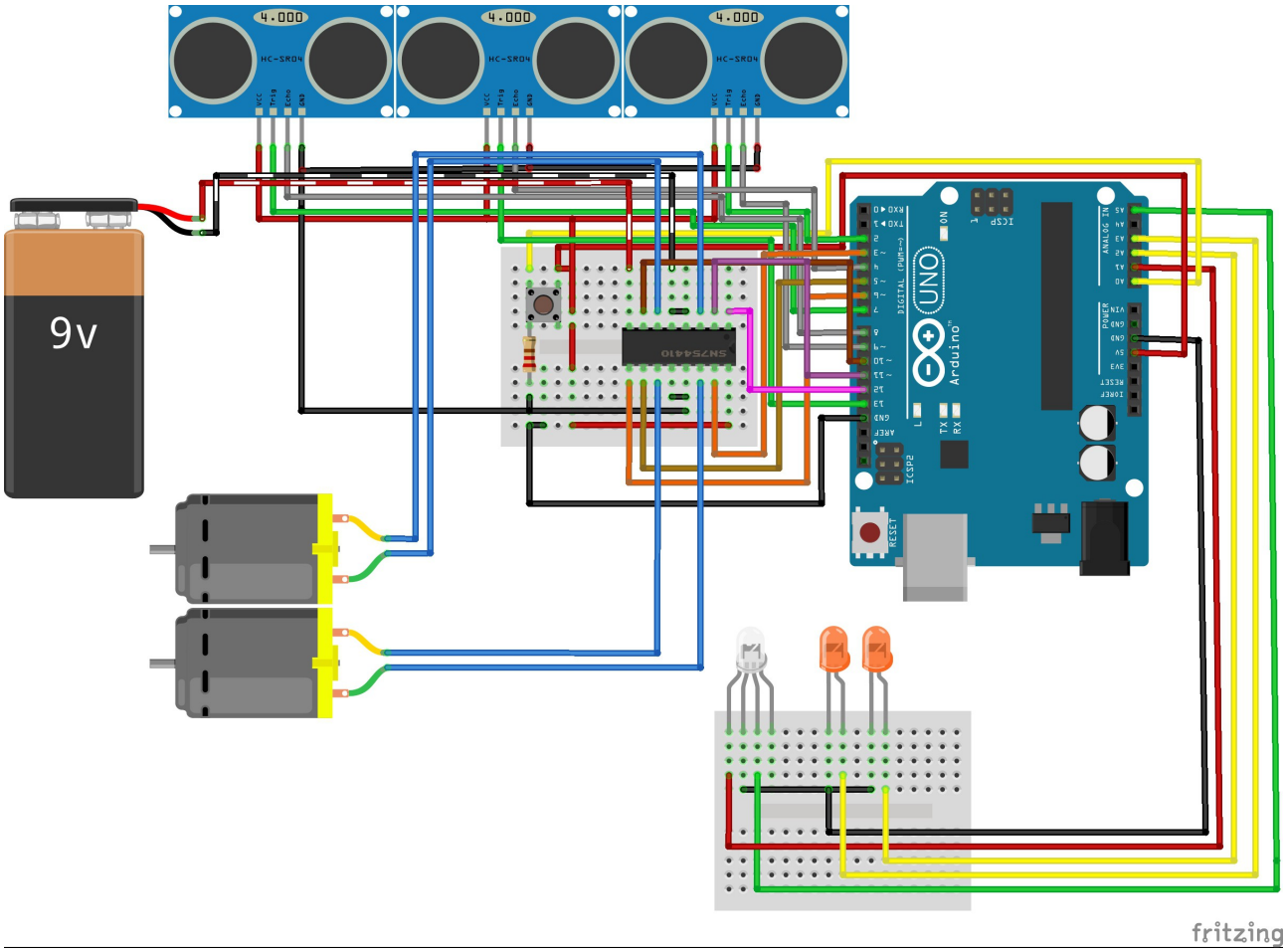
3. Pin Belegung am Arduino

Pin	Belegung
D0	frei
D1	frei
D2	Trigger Sensor 1
D3	Speed Motor1
D4	Echo Sensor 2
D5	Speed Motor 2
D6	Enable Motor 2
D7	Trigger Sensor 3
D8	Echo Sensor 3
D9	Echo Sensor 1
D10	Speed Motor 2
D11	Speed Motor 1
D12	Enable Motor 1
D13	Trigger Sensor 2
A0	Input Taster
A1	Status LED - rot
A2	Blinker rechts
A3	Blinker links
A4	frei
A5	Status LED - grün

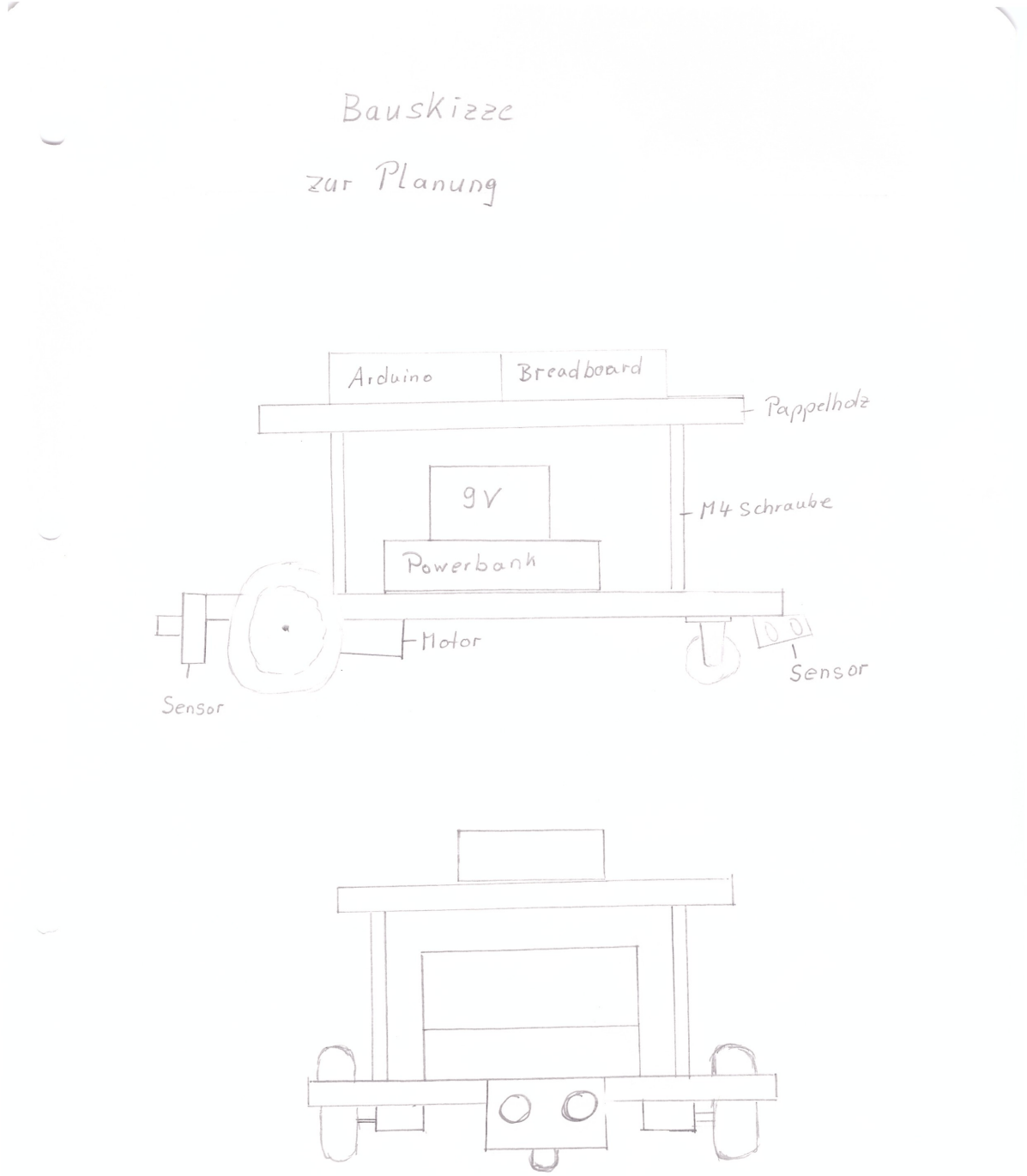
4. Schaltplan



5. Schaltskizze

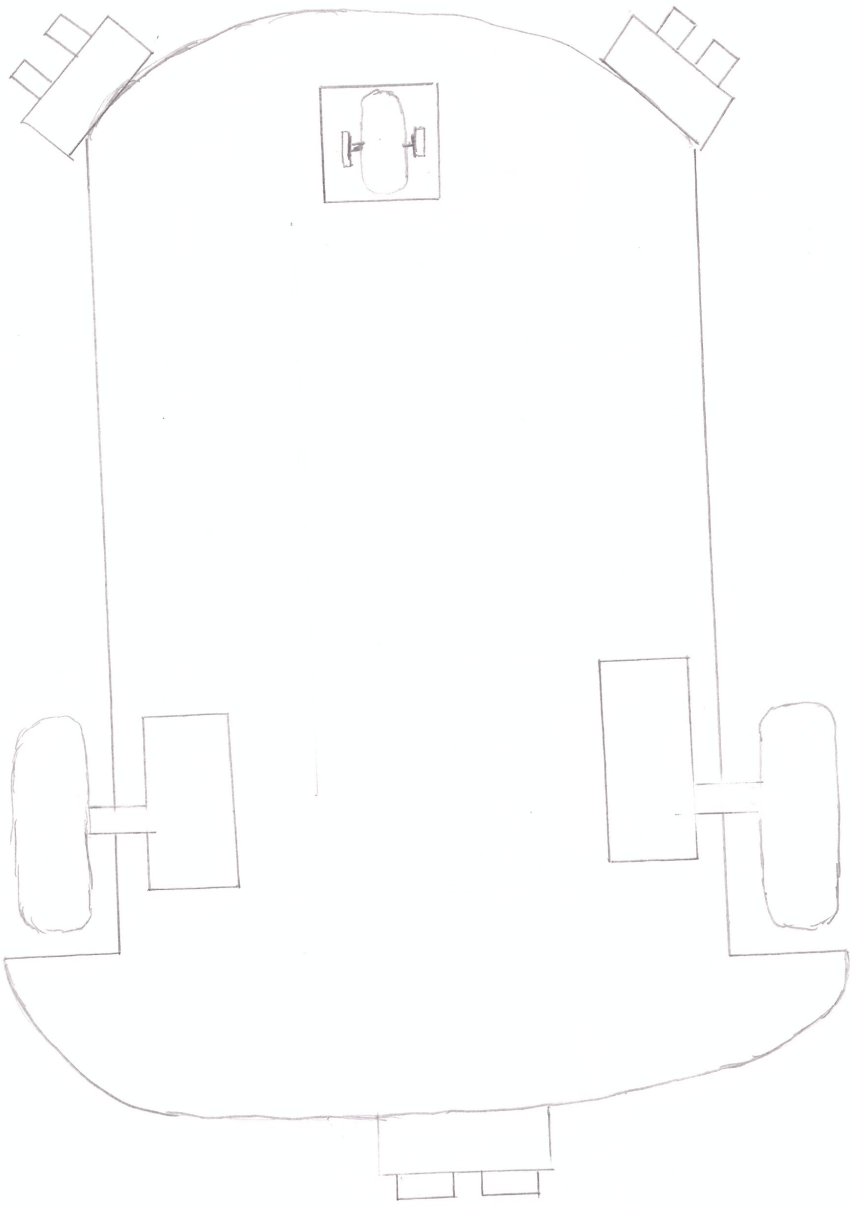


6. Skizze des Fahrzeuges

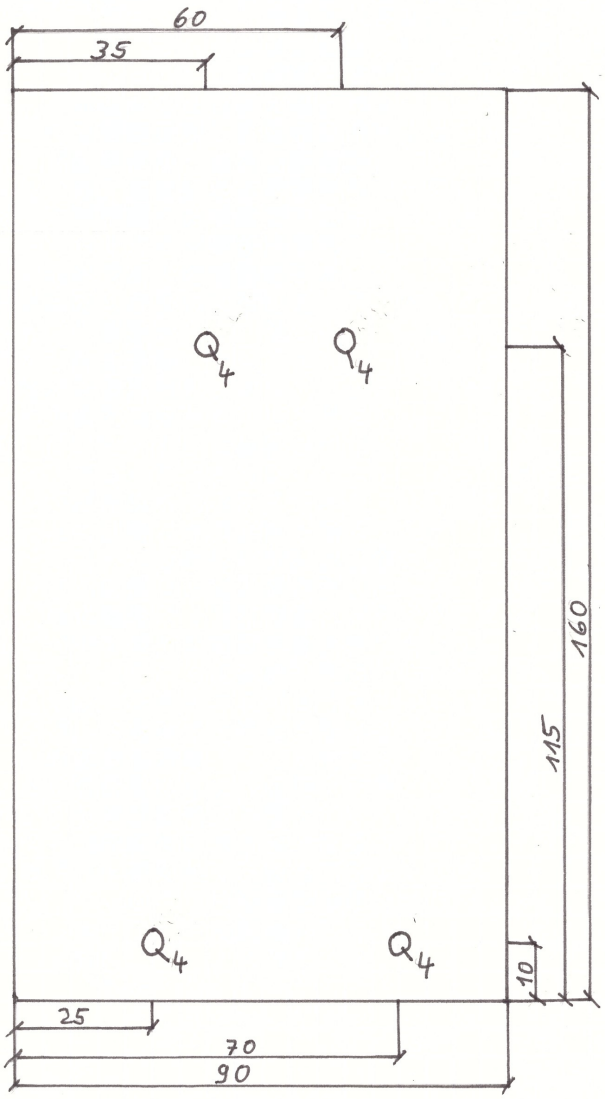


Bauskizze

zur Planung



7. Technische Zeichnung des Anbaus



Zusatzboden aus 5mm Leimholz

8. Code

```
//Sensor definition
#include <NewPing.h>
int MAX_DISTANCE = 30;


NewPing sensor1 (13, 4, MAX_DISTANCE);
NewPing sensor2 (2, 9, MAX_DISTANCE);
NewPing sensor3 (7, 8, MAX_DISTANCE);


//Pin definition

int mot1A_speed_pin = 3;
int mot2A_speed_pin = 5;
int mot1B_speed_pin = 11;
int mot2B_speed_pin = 10;
int mot1_enable_pin = 12;
int mot2_enable_pin = 6;
int input_pin = A0;
int led_rt = A1;
int led_gr = A5;
int links = A3;
int rechts = A4;

//Variablen definition

int distance1;
int distance2;
int distance3;
int richtung = 0;

int dir = 0;
int input = 0;
int mot1_enable = 0;
int mot2_enable = 0;
bool stat = false;


//setup

void setup() {
  pinMode (input, INPUT);
  pinMode (mot1A_speed_pin, OUTPUT);
  pinMode (mot1B_speed_pin, OUTPUT);
  pinMode (mot2A_speed_pin, OUTPUT);
  pinMode (mot2B_speed_pin, OUTPUT);
  pinMode (mot1_enable_pin, OUTPUT);
  pinMode (mot2_enable_pin, OUTPUT);
  pinMode (led_rt, OUTPUT);
  pinMode (led_gr, OUTPUT);
  pinMode (links, OUTPUT);
  pinMode (rechts, OUTPUT);
  dir = 0;
  mot1_enable = 0;
  mot2_enable = 0;
  stat = false;
  Serial.begin(9600);
  delay(200);
  Serial.println("Debug Protokoll:");
  Serial.print("setup finished after ");
  Serial.println(millis());
  Serial.println("------------------------------------------------------------");
  debug();
  digitalWrite(led_rt, HIGH);
  delay(500);
  digitalWrite(led_rt, LOW);
  delay(500);
  digitalWrite(led_rt, HIGH);
  delay(500);
  digitalWrite(led_rt, LOW);
  delay(1000);
  digitalWrite(led_rt, HIGH);
}

void loop() {
  if (Serial.available() > 0) {
    String debug_inc = Serial.readString();
    if (debug_inc = "DEBUG") {
      mess();
```

```
    debug();


   }
  }
  //status steuerung
  state() ;

  if (stat == 1) {

//steuerung

   mess();
   control();

  }

  else {
   stopp();
  }

}

void state() {
  input = digitalRead(input_pin);
  if (stat == true && input == HIGH) {
   stat = false;
   digitalWrite(led_rt, HIGH);
   digitalWrite(led_gr, LOW);
   delay(1000);
   debug();
  }
  else {
   if (stat == false && input == HIGH) {
    stat = true;
    delay(2000);
    digitalWrite(led_rt, LOW);
    digitalWrite(led_gr, HIGH);
    debug();
   }
  }

}



void debug() {
  Serial.print("Debug von Laufzeit: ");
  Serial.print(millis());
  Serial.print("   |   ");
  Serial.print(millis() / 1000);
  Serial.println(" Sekunden");
  Serial.println("  ");
  Serial.print("Status: ");
  Serial.println(stat);
  Serial.print("direction: ");
  Serial.println(dir);
  Serial.print("Motor 1 enable: ");
  Serial.println(mot1_enable);
  Serial.print("Motor 2 enable: ");
  Serial.println(mot2_enable);
  Serial.print("Distance Sensor1:  ");
  Serial.println (distance1);
  Serial.print("Distance Sensor2:  ");
  Serial.println (distance2);
  Serial.print("Distance Sensor3:  ");
  Serial.println (distance3);
  Serial.print("Richtung  ");
  Serial.println(richtung);
  Serial.println("debug done!");
  Serial.println("------------------------------------------------------------");
  Serial.println(" ");
  Serial.println(" ");
  Serial.println(" ");
}

void vor() {
  digitalWrite(rechts,LOW);
digitalWrite(links,LOW);
  mot1_enable = 255;
  mot2_enable = 255;
  analogWrite(mot1A_speed_pin, 150);
  analogWrite(mot2A_speed_pin, 0);
```

```
  analogWrite(mot1B_speed_pin, 150);
  analogWrite(mot2B_speed_pin, 0);
  digitalWrite(mot2_enable_pin, mot2_enable);
  digitalWrite(mot1_enable_pin, mot1_enable);

}

void back() {

digitalWrite(rechts,HIGH);
digitalWrite(links,HIGH);
  mot1_enable = 255;
  mot2_enable = 255;
  analogWrite(mot1A_speed_pin, 0);
  analogWrite(mot2A_speed_pin, 150);
  analogWrite(mot1B_speed_pin, 0);
  analogWrite(mot2B_speed_pin, 150);
  digitalWrite(mot2_enable_pin, mot2_enable);
  digitalWrite(mot1_enable_pin, mot1_enable);


}

void right() {
  digitalWrite(rechts,LOW);
digitalWrite(links,LOW);
  digitalWrite(rechts, HIGH);
  mot1_enable = 255;
  mot2_enable = 255;
  analogWrite(mot1A_speed_pin, 0);
  analogWrite(mot2A_speed_pin, 150);
  analogWrite(mot1B_speed_pin, 150);
  analogWrite(mot2B_speed_pin, 0);
  digitalWrite(mot1_enable_pin, mot1_enable);
  digitalWrite(mot2_enable_pin, mot2_enable);
  delay(247);
  dir = dir + 90;
  mot1_enable = 0;
  mot2_enable = 0;
  analogWrite(mot1A_speed_pin, 0);
  analogWrite(mot2A_speed_pin, 0);
  analogWrite(mot1B_speed_pin, 0);
  analogWrite(mot2B_speed_pin, 0);
  digitalWrite(mot1_enable_pin, mot1_enable);
  digitalWrite(mot2_enable_pin, mot2_enable);
  digitalWrite(rechts,LOW);

}

void left() {
  digitalWrite(rechts,LOW);
digitalWrite(links,LOW);
  digitalWrite(links, HIGH);
  mot1_enable = 225;
  mot2_enable = 255;
  analogWrite(mot1A_speed_pin, 150);
  analogWrite(mot2A_speed_pin, 0);
  analogWrite(mot1B_speed_pin, 0);
  analogWrite(mot2B_speed_pin, 150);
  digitalWrite(mot1_enable_pin, mot1_enable);
  digitalWrite(mot2_enable_pin, mot2_enable);
  delay(247);
  dir = dir - 90;
  mot1_enable = 0;
  mot2_enable = 0;
  analogWrite(mot1A_speed_pin, 0);
  analogWrite(mot2A_speed_pin, 0);
  analogWrite(mot1B_speed_pin, 0);
  analogWrite(mot2B_speed_pin, 0);
  digitalWrite(mot1_enable_pin, mot1_enable);
  digitalWrite(mot2_enable_pin, mot2_enable);
  digitalWrite (links, LOW);

}

void stopp() {
  mot1_enable = 0;
  mot2_enable = 0;
  analogWrite(mot1A_speed_pin, 0);
  analogWrite(mot2A_speed_pin, 0);
  analogWrite(mot1B_speed_pin, 0);
  analogWrite(mot2B_speed_pin, 0);
  digitalWrite(mot1_enable_pin, mot1_enable);
```

```
  digitalWrite(mot2_enable_pin, mot2_enable);
}

void error() {
  stat = false;
  stopp();
  digitalWrite(led_rt, HIGH);
  digitalWrite(led_gr, LOW);
  digitalWrite(rechts,LOW);
digitalWrite(links,LOW);
}


void mess() {
  distance1 = sensor1.ping_cm();
  distance2 = sensor2.ping_cm();
  distance3 = sensor3.ping_cm();


  delay(100);

}

void control() {

if(distance1 > 0 || distance2 > 0){

    if(distance1 < distance2){
      richtung = 2;
    }
    if(distance1 > distance2){
      richtung = 1;
    }
    if(distance1 < 15 && distance1 > 0 || distance2 < 15 && distance2 > 0){
      if(distance3 < 15 && distance3 > 0){
        error();
      }
      else{
        richtung = 10;
      }
    }
}
else{
  richtung = 0;
}

//richtungs definition
  if (richtung == 1) {

    left();

  }

  if (richtung == 2) {

    right();
  }

  if (richtung == 0) {

    vor();
  }

  if (richtung == 10) {

    back();
  }
}
```