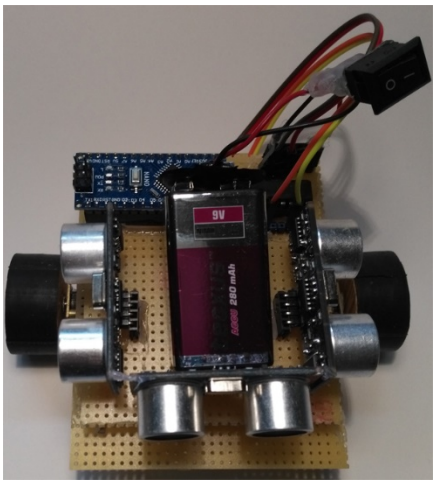
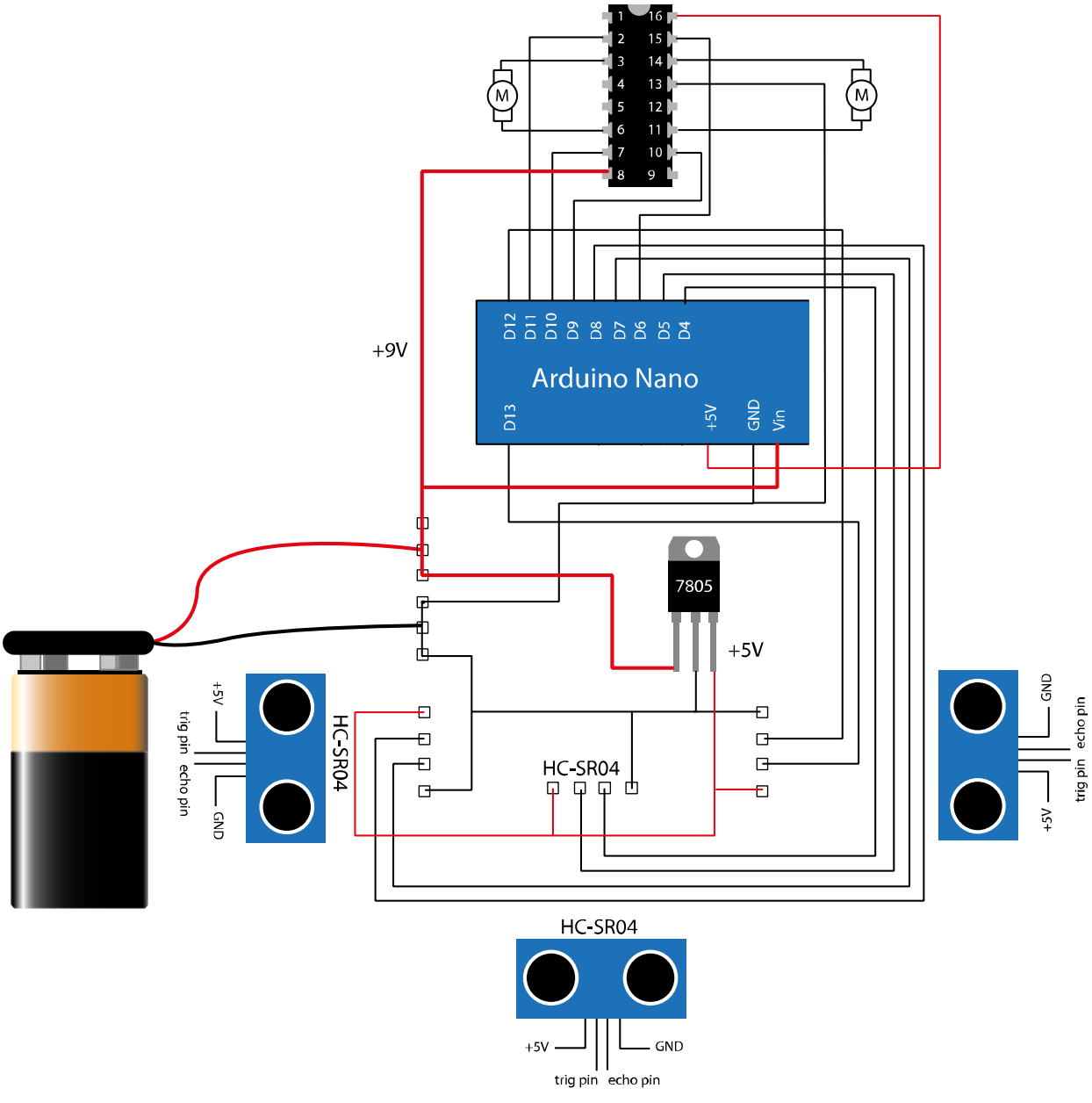
Durchs Labyrinth mit 3 Ultraschall-Sensoren

Das Fahrzeug

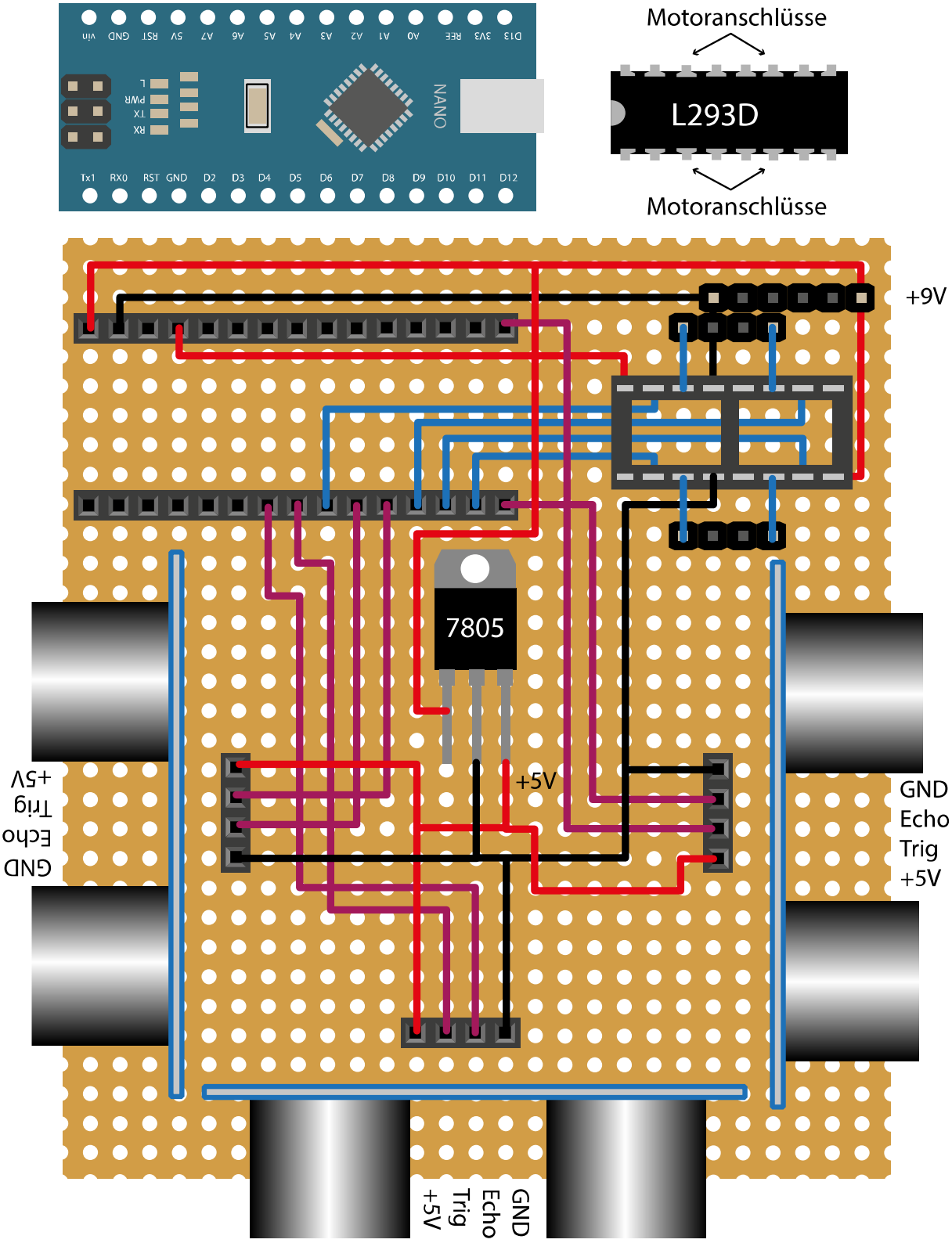


Bei dem Fahrgestell handelt es sich um das des Robino I, für das hier statt eines Holzbrettchens eine Lochrasterplatine verwendet wurde. Um Kabelgewirr und mögliche Wackelkontakte zu vermeiden, sind die benötigten Anschlüsse zwischen Arduino (Nano) Motorentreiber (L 293 D) und Ultraschallsensoren (HC-SR054) mithilfe von Buchsen- und Steckleisten realisiert worden. Die Teile können so jederzeit ausgetauscht werden.
Die Stromversorgung wird von einer 9V-Blockbatterie sichergestellt.
Die Reduzierung der Spannung von 9 auf 5 Volt übernimmt ein Spannungsregler (L7805).



Ein Ein/Aus-Schalter vereinfacht das Starten und Stoppen.

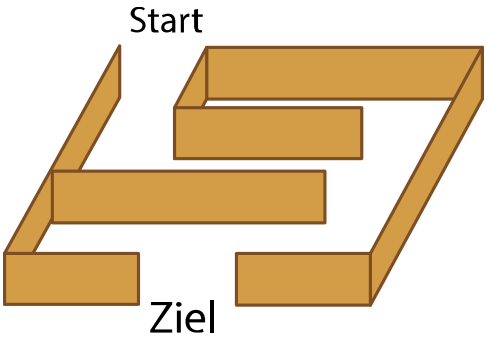
Die Umsetzung des Schaltplans auf die Lochrasterplatine kann der folgenden Abbildung entnommen werden, wobei zu bedenken ist, dass sich die Verbindungsleitungen auf der Rückseite der Platine befinden.



Der Arduino Nano und der Motorentreiber L 293 D sind in der Abbildung so dargestellt, dass sie in dieser Ausrichtung in die Buchsenleisten  zu stecken sind. Für die Anschlüsse der Batterie und der Motoren werden Stiftleisten  verwendet. Aus ihnen sind einige Stifte herausgezogen worden (graue Felder). Die verbliebenen Stifte sind als helle Felder dargestellt.

Das Labyrinth

Zum Bau eines einfachen Labyrinths eignet sich Kartonpappe, die in 7 – 10 cm breite Streifen geschnitten und wie in der Abbildung zusammengeklebt werden. Die Breite der Wege ist dem Fahrzeug anzupassen. Hier waren es 16 cm.



Zielführender Algorithmus nach der „Linken-Hand-Regel“:
Geradeaus – 90 ° Linkskurve – geradeaus – 180 ° Linkskurve – geradeaus – Kehrtwendung – geradeaus – 90 ° Rechtskurve – geradeaus – 90° Rechtskurve – geradeaus – 90 ° Linkskurve.

Der Programmcode

Die Durchfahrt mit dem gewählten Fahrzeug verlässlich zu machen, beinhaltet einige Herausforderungen, die sich bei der praktischen Erprobung zeigen. Der nachfolgende Programmcode sollte deshalb nur als eine Annäherung an die gewünschte Lösung betrachtet werden:

```
#define m_links_a 11
#define m_links_b 10
#define m_rechts_a 9
#define m_rechts_b 6
int trigger_vorn = 5;
int echo_vorn = 4;
int trigger_rechts = 8;
int echo_rechts = 7;
int trigger_links = 13;
int echo_links = 12;

void setup() {
  pinMode(trigger_vorn, OUTPUT);
  pinMode(echo_vorn, INPUT);
  pinMode(trigger_rechts, OUTPUT);
  pinMode(echo_rechts, INPUT);
  pinMode(trigger_links, OUTPUT);
  pinMode(echo_links, INPUT);
  pinMode(m_links_a, OUTPUT);
  pinMode(m_links_b, OUTPUT);
  pinMode(m_rechts_a, OUTPUT);
  pinMode(m_rechts_b, OUTPUT);

  analogWrite(m_links_a, 255);
  analogWrite(m_links_b, 0);
  analogWrite(m_rechts_a, 255);
  analogWrite(m_rechts_b, 0);
}
void rechts_1() {
  analogWrite(m_rechts_a, 0);
  delay(100);
  analogWrite(m_rechts_a, 255);
}
```

```
void links_1() {
  analogWrite(m_links_a, 0);
  delay(100);
  analogWrite(m_links_a, 255);
}
void rechts_2() {
  analogWrite(m_rechts_a, 0);
  analogWrite(m_rechts_b, 255);
  delay(800);
  analogWrite(m_rechts_a, 255);
  analogWrite(m_rechts_b, 0);
}
void links_2() {
  delay(600);
  analogWrite(m_links_a, 0);
  analogWrite(m_links_b, 255);
  delay(750);
  analogWrite(m_links_a, 255);
  analogWrite(m_links_b, 0);
  delay(800);
}
void loop()  {
  long dauer_vorn, dauer_rechts, dauer_links, strecke_links, strecke_rechts,
strecke_vorn;
  digitalWrite(trigger_vorn, LOW);
  delayMicroseconds(2);
  digitalWrite(trigger_vorn, HIGH);
  delayMicroseconds(5);
  digitalWrite(trigger_vorn, LOW);
  dauer_vorn = pulseIn(echo_vorn, HIGH);
  strecke_vorn = dauer_vorn / 29 / 2;
  digitalWrite(trigger_rechts, LOW);
  delayMicroseconds(2);
  digitalWrite(trigger_rechts, HIGH);
  delayMicroseconds(5);
  digitalWrite(trigger_rechts, LOW);
  dauer_rechts = pulseIn(echo_rechts, HIGH);
  strecke_rechts = dauer_rechts / 29 / 2;
  digitalWrite(trigger_links, LOW);
  delayMicroseconds(2);
  digitalWrite(trigger_links, HIGH);
  delayMicroseconds(5);
  digitalWrite(trigger_links, LOW);
  dauer_links = pulseIn(echo_links, HIGH);
  strecke_links = dauer_links / 29 / 2;

  if (strecke_links < 7) // Spur halten
    rechts_1();

  if (strecke_rechts < 7) // Spur halten
    links_1();

  if (strecke_rechts > 19) // links bleiben
    links_1();

  if (strecke_links > 19) // nach links abbiegen
    links_2();

  if (strecke_vorn < 7) // nach rechts abbiegen
    rechts_2();
}
```