Dokumentation   
Lieferroboter

Elektrotechnik

Inhalt

[1. Bauteil – Liste 1](#_Toc24022393)

[1.1. Boards 1](#_Toc24022394)

[1.2. Aktoren 1](#_Toc24022395)

[2. Pin – Zuordnungen 2](#_Toc24022396)

[2.1. Arduino A1 🡪 Motortreiber X1 2](#_Toc24022398)

[2.2. Arduino A1 🡪 IR Modul A2 2](#_Toc24022399)

[2.3. Arduino A1 🡪 IR Modul A3 2](#_Toc24022400)

[3. Schaltplan 1](#_Toc24022401)

[4. Steckplatine 2](#_Toc24022402)

[5. Medien 1](#_Toc24022403)

[6. Programmcode 2](#_Toc24022404)

1. Bauteil – Liste
   1. Boards

|  |  |
| --- | --- |
| **Boardbezeichnung** | **Beschreibung** |
| Arduino Uno R3 (A1) | Steuerung des Systems |
| Motortreiber L-9110H (T1) | Steuerung der DC Motoren |
| IR – Board (A2) bestehend aus IR – LED und Fotowiderstand | Erkennung der Orientierungslinie (links) |
| IR – Board (A3) bestehend aus IR – LED und Fotowiderstand | Erkennung der Orientierungslinie (rechts) |

* 1. Aktoren

|  |  |
| --- | --- |
| **Aktorbezeichnung** | **Beschreibung** |
| M1: JOY-IT Getriebemotor (COM-Motor01) | Radsteuerung vorne links |
| M2: JOY-IT Getriebemotor (COM-Motor01) | Radsteuerung vorne rechts |
| M3: JOY-IT Getriebemotor (COM-Motor01) | Radsteuerung hinten links |
| M4: JOY-IT Getriebemotor (COM-Motor01) | Radsteuerung hinten rechts |

1. Pin – Zuordnungen
2. 1. Arduino A1 🡪 Motortreiber X1

|  |  |
| --- | --- |
| **Pin – Arduino A1** | **Pin – Motortreiber X1** |
| 3 | A1A |
| 2 | A1B |
| 5 | B1A |
| 4 | B1B |
| 5V | VCC |
| GND | GND |

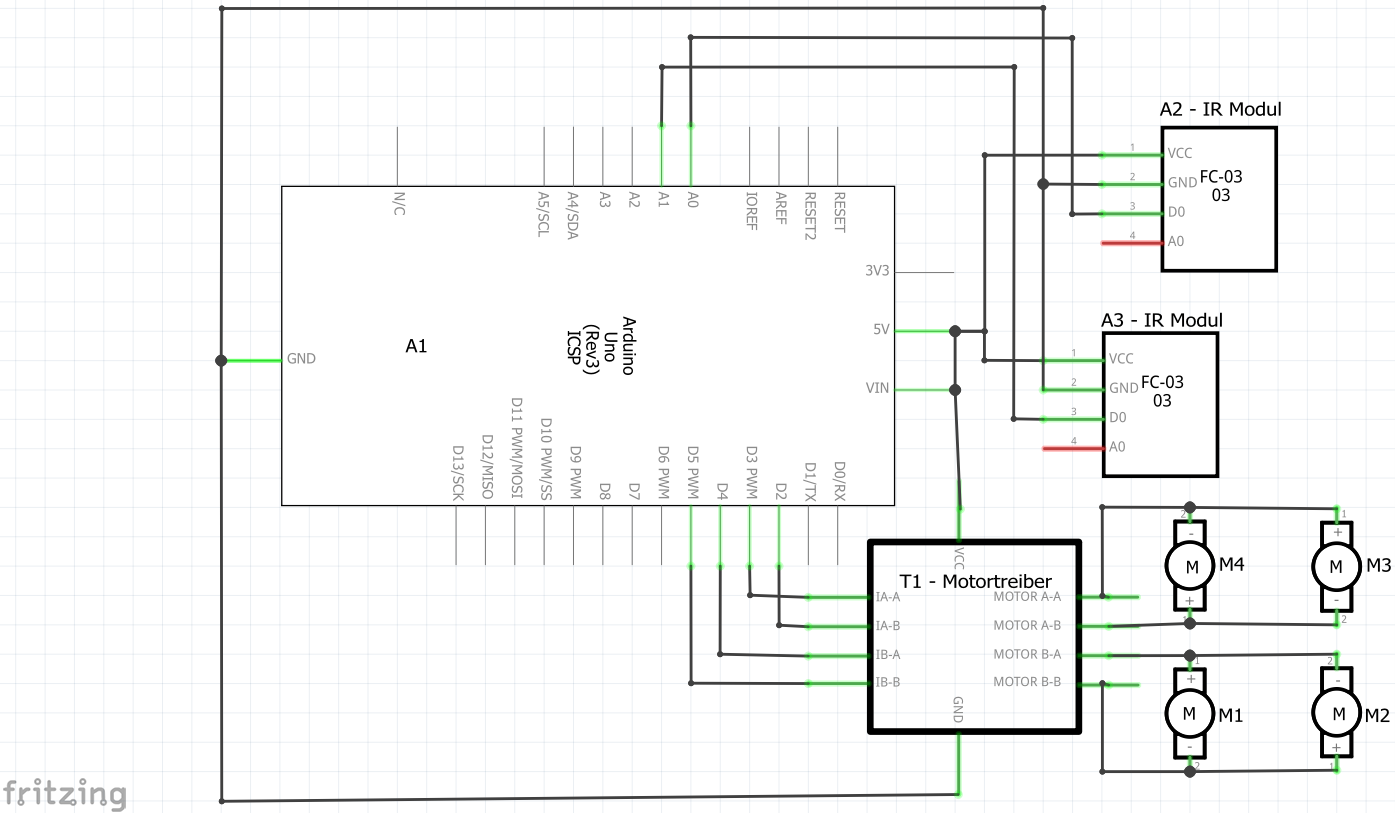
* 1. Arduino A1 🡪 IR Modul A2

|  |  |
| --- | --- |
| **Pin – Arduino A1** | **Pin – Motortreiber X1** |
| 5V | VCC |
| GND | GND |
| A0 | D1 |

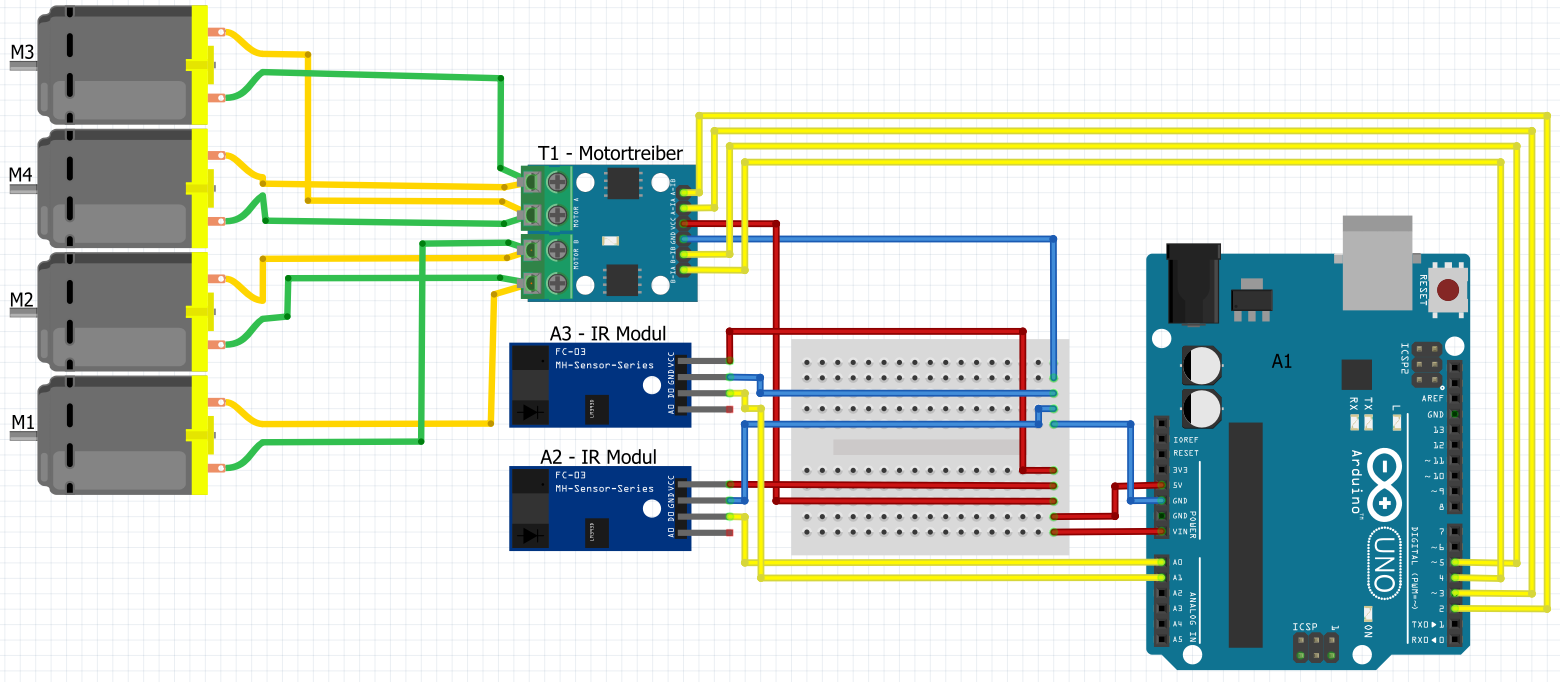
* 1. Arduino A1 🡪 IR Modul A3

|  |  |
| --- | --- |
| **Pin – Arduino A1** | **Pin – Motortreiber X1** |
| 5V | VCC |
| GND | GND |
| A1 | D1 |

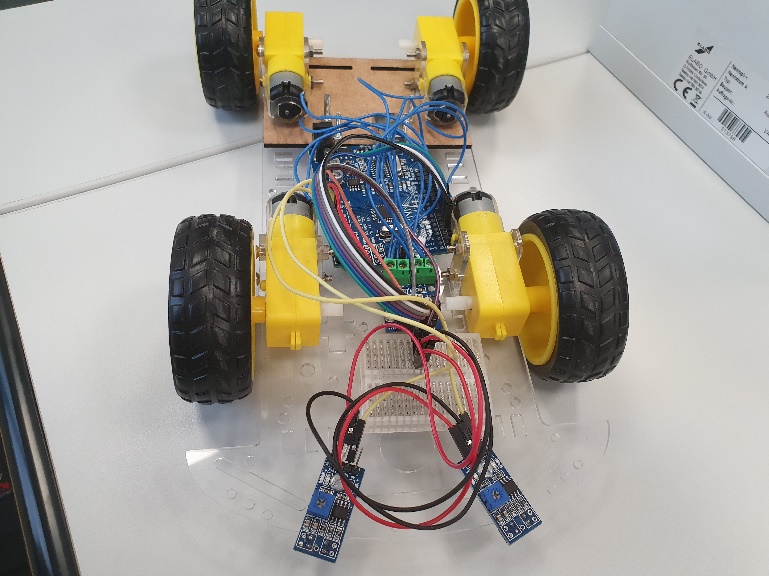
1. Schaltplan

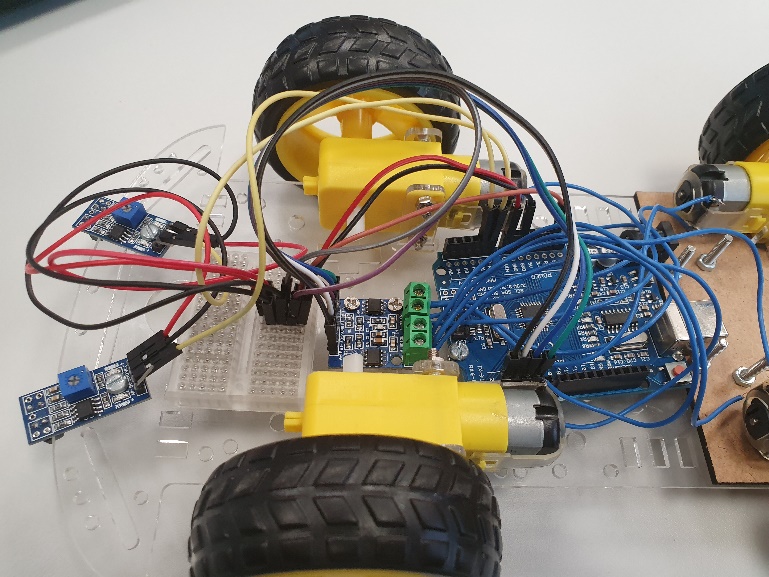


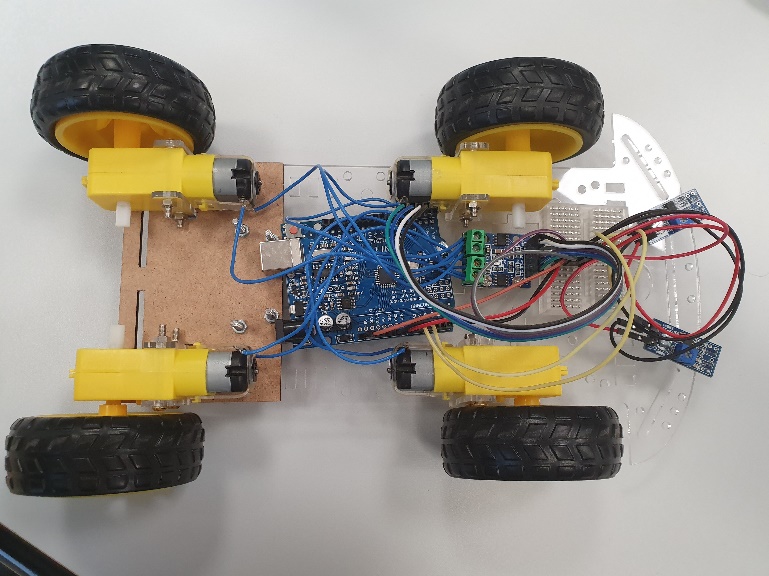
1. Steckplatine

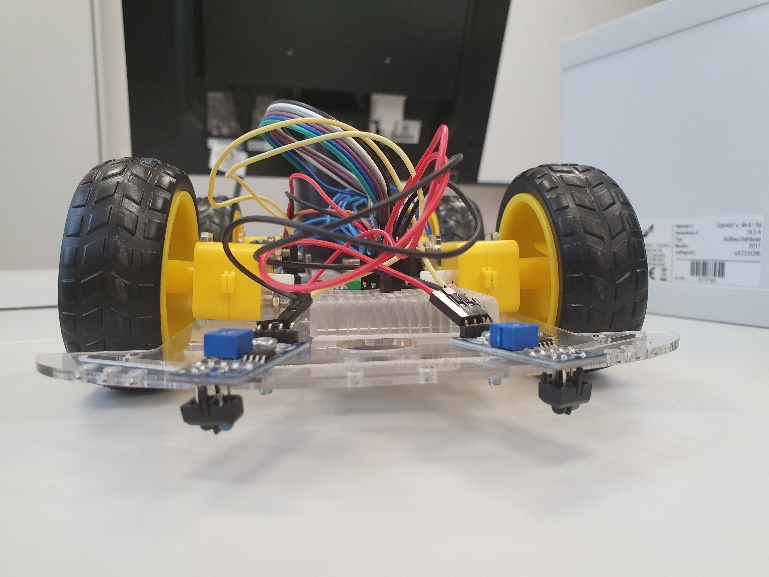


1. Medien









1. Programmcode

Der Programmcode für den Arduino wird als .ino Datei zur Verfügung gestellt. Außerdem ist das gesamte Projekt inkl. Coding hier auf Github verfügbar. Folgend wird der Code auch noch einmals aufgelistet.

1. // Lieferroboter
3. //Pins deklarieren
4. **const** **int** forward\_left = 3;
5. **const** **int** forward\_right = 4;
6. **const** **int** backward\_left = 2;
7. **const** **int** backward\_right = 5;
8. **const** **int** sensor\_left = A0;
9. **const** **int** sensor\_right = A1;
11. //Richtung deklarieren
12. **const** **int** backward = 0b1100;
13. **const** **int** forward = 0b0011;
14. **const** **int** right = 0b1001;
15. **const** **int** left = 0b0110;
16. **const** **int** stop = 0b1111;
18. //Werte deklarieren
19. **const** **int** sensor\_limit = 100;
21. **void** setup() {
22. pinMode(forward\_left, OUTPUT);
23. pinMode(forward\_right, OUTPUT);
24. pinMode(backward\_left, OUTPUT);
25. pinMode(backward\_right, OUTPUT);
26. Serial.begin(9600);
27. }
29. **void** loop() {
31. //Beide über Grenzwert -> Vorwärts
32. **while** (analogRead(sensor\_left) <= sensor\_limit && analogRead(sensor\_right) <= sensor\_limit) {
33. fahren(0b0000);
34. fahren(forward);
36. }
37. //Links unter Grenzwert, rechts über Grenzwert -> Links
38. **while** (analogRead(sensor\_left) <= sensor\_limit && analogRead(sensor\_right) >= sensor\_limit) {
39. fahren(0b0000);
40. fahren(right);
41. }
42. //Links über Grenzwert, rechts unter Grenzwert -> Rechts
43. **while** (analogRead(sensor\_left) >= sensor\_limit && analogRead(sensor\_right) <= sensor\_limit) {
44. fahren(0b0000);
45. fahren(left);
46. }
47. }
49. **void** fahren(uint8\_t value) {
50. digitalWrite(backward\_right, value & 0b1000);
51. digitalWrite(backward\_left, value & 0b0100);
52. digitalWrite(forward\_right, value & 0b0010);
53. digitalWrite(forward\_left, value & 0b0001);
54. }