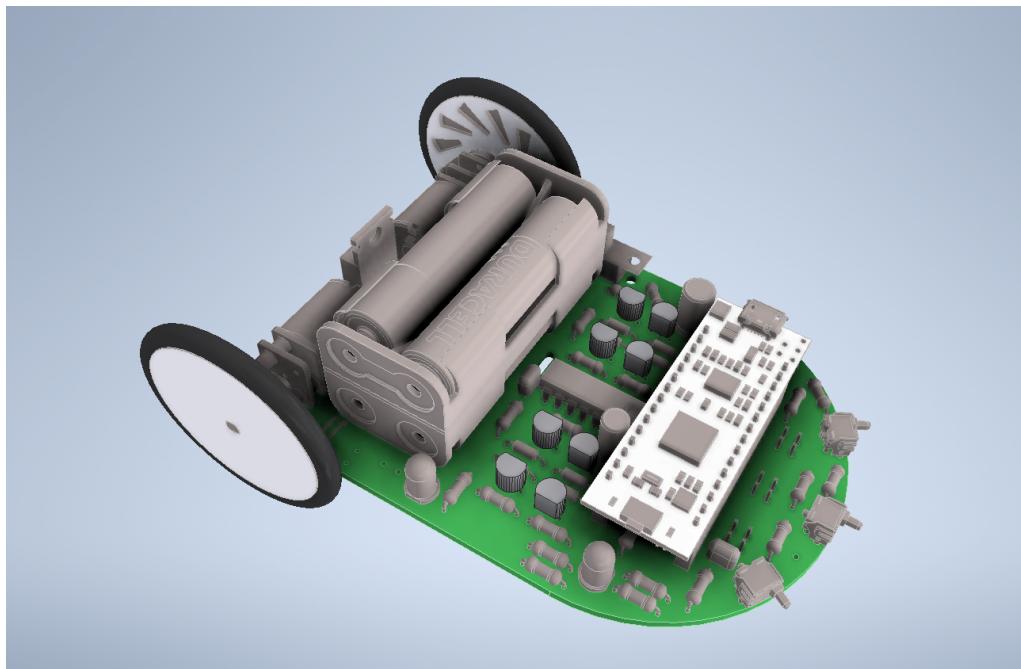


Basispraktikum Mobile Roboter

Armuro Manual

Sommersemester 2024



1 Einleitung

Im Basispraktikum Mobile Roboter am Lehrstuhl für Hochperformante Humanoide Technologien (H^2T) wird Stück für Stück ein kleiner Roboter *Armuro* aufgebaut und in Betrieb genommen. Der Roboter wird über einen Mikrocontroller gesteuert und kann mithilfe verschiedener Sensoren seine Umgebung und seinen eigenen Zustand wahrnehmen. Er verfügt über zwei Motoren, mit deren Hilfe er sich fortbewegen kann.

Ziel des Praktikums ist es, Grundwissen auf den Gebieten der Schaltungstechnik und der Programmierung von Mikrocontrollern in C zu erlernen. Außerdem werden Kenntnisse zur Ansteuerung der Sensoren und Aktoren des Roboters und der reflexbasierten autonomen Verhaltensgenerierung vermittelt. Am Ende soll der Roboter in der Lage sein, einen kleinen Hindernisparcour zu bewältigen.

Diese Anleitung enthält die wichtigsten Punkte zum Aufbau des Roboters und der Einrichtung der Software, die zum Programmieren des Mikrocontrollers benötigt wird. Sie soll durch die erste Hälfte des Praktikums begleiten und ergänzt die täglichen Kolloquien und Übungsblätter, in denen tiefere Kenntnisse zu den einzelnen Themen vermittelt werden.

2 Elektronik

In diesem Kapitel werden die Kenntnisse zum Aufbau des *Armuro*-Roboters vermittelt. Zuerst werden die einzelnen Bauteile behandelt und eine Einführung in den Lötvorgang gegeben. Danach folgt eine Anleitung, in der Schritt für Schritt durch den Aufbau des *Armuro* geführt wird. Zum Schluss wird ein Testprogramm vorgestellt, das nach Fertigstellung auf den Roboter gespielt wird, um seine Funktionsfähigkeit zu überprüfen.

2.1 Bauteile

Im *Armuro* werden verschiedenste elektrische und mechanische Bauteile verwendet. Um eine Übersicht über die Bauteile zu gewinnen, werden hier ihre wichtigsten Eigenschaften und Funktionen vorgestellt. Einige dieser Bauteile werden außerdem in den Kolloquien und Übungsblättern noch einmal separat behandelt.

2.1.1 Platine

Die mechanische Grundstruktur des Roboters ist durch die Hauptplatine (Abb. 1) vorgegeben. Auf ihr werden alle elektrischen Bauteile befestigt. Sie hat einen Kern aus elektrisch isolierendem Material,

auf der sich auf beiden Seiten eine dünne Kupferschicht befindet, aus der Leiterbahnen zum Leiten des elektrischen Stroms geätzt wurden. Um um die Leiterbahnen zu schützen, sind diese mit grünem Lötstopplack verdeckt. Auf der Oberfläche sind nur noch die vergoldeten Lötpads zu sehen, die die Verbindungsstellen zwischen den Bauteilen und den Leiterbahnen herstellen sollen. Im Englischen wird der Begriff Platine mit *Printed Circuit Board (PCB)* beschrieben.

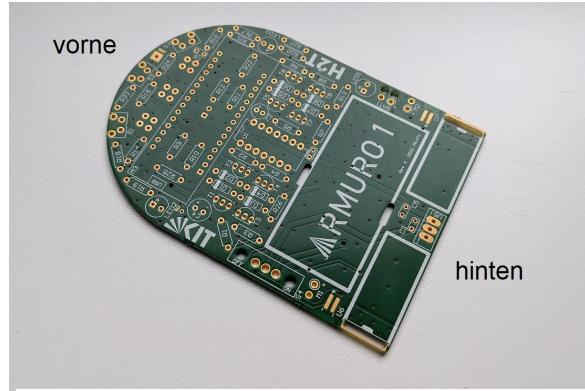


Abb. 1: Hauptplatine des *Armuro*

2.1.2 Widerstände

Ein essenzielles Bauteil für elektronische Schaltungen ist der Widerstand (Abb. 2). Es gilt das ohmsche Gesetz: $R = \frac{U}{I}$, wobei U die Spannung und I die elektrische Stromstärke I sind. R wird in Ohm (Ω) gemessen. Im Bausatz werden Kohleschichtwiderstände mit einer maximalen Verlustleistung von $P = U \cdot I = 0.25 \text{ W}$ verwendet. Auf den Widerständen sind farbige Ringe aufgemalt, die den elektrischen Wert des Widerstandes kodieren. In Abbildung 3 ist die Bedeutung der Farben erklärt.



Abb. 2: Widerstand

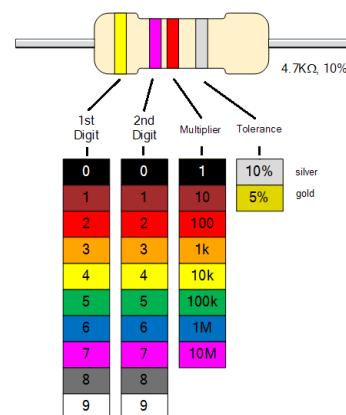


Abb. 3: Farbkodierung eines Widerstands

2.1.3 Kondensatoren

Kondensatoren dienen zum Speichern von elektrischer Ladung Q und werden über ihre elektrische Kapazität $C = \frac{Q}{U}$, die in Farad (F) gemessen wird, beschrieben. Es gibt unterschiedliche technische Realisierungen von Kondensatoren. Bei kleinen Kapazitäten werden oft Keramikkondensatoren (Abb. 4) eingesetzt, bei denen die Platten des Kondensators durch dünne Keramiksichten getrennt werden. Bei größeren Kapazitäten werden meist Elektrolytkondensatoren (Abb. 5) verwendet. Diese Kondensatoren weisen im Gegensatz zu den Keramikkondensatoren eine **Polung** auf, da für Anode und Kathode unterschiedliche Materialien verwendet werden.



Abb. 4: Keramikkondensator

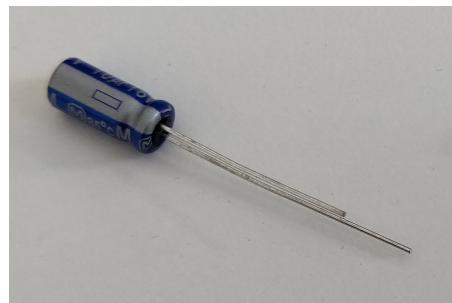


Abb. 5: Elektrolytkondensator

2.1.4 Dioden

Dioden sind elektronische Bauelemente, die den Strom nur in eine bestimmte Richtung leiten. Dabei leiten sie von der Anode zu Kathode hin, umgekehrt sind sie nicht leitend und zeigen ein Sperrverhalten. Daher ist auch hier die Polung der Bauelemente relevant. Abbildung 6 zeigt ein Beispiel für eine Diode. Der schwarze Strich markiert die Seite der Kathode. Leuchtdioden (Abb. 7) bestehen aus Halbleitermaterialien, die leuchten, wenn Strom hindurch fließt. Um die Polung zu markieren, ist die Kathodenseite der hier verwendeten runden LEDs leicht abgeflacht.

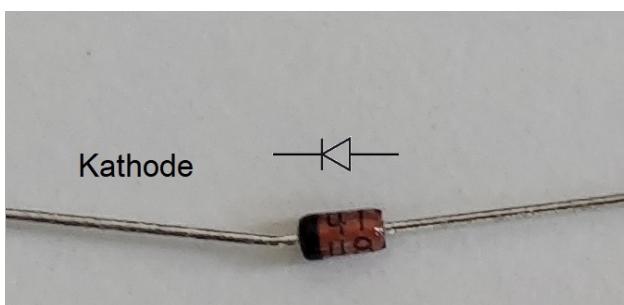


Abb. 6: Diode

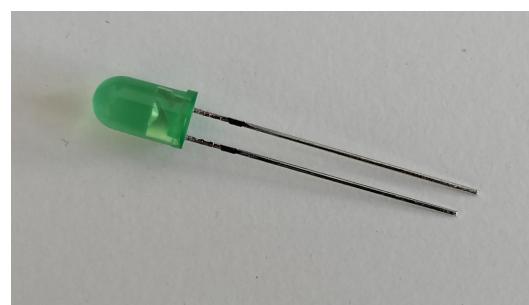


Abb. 7: Leuchtdiode (LED)

2.1.5 Schalter und Taster

Ein Schalter (Abb. 8) kann je nach Einstellung eine elektrisch leitende Verbindung herstellen oder trennen. Im Gegensatz dazu fließt bei einem Taster (Abb. 9) nur dann Strom, wenn er betätigt wird.



Abb. 8: Schalter



Abb. 9: Taster

2.1.6 Transistoren

Ein weiteres wichtiges Bauelement ist der Transistor (Abb. 11), der die Funktion eines elektronischen Schalters übernimmt. Er verfügt über drei Anschlüsse: Kollektor (*C*), Basis (*B*) und Emitter (*E*). Fließt ein Strom I_{BE} durch die Basis zum Emitter, so schaltet der Transistor durch und es kann ebenfalls ein Strom I_{CE} vom Kollektor zum Emitter fließen. I_{BE} kann dabei geringer sein als I_{CE} , der Transistor dient so also als Verstärker: $I_{CE} = \beta \cdot I_{BE}$. Falls kein Strom durch die Basis fließt, sperrt der Transistor und $I_{CE} = 0$.

Bipolare Transistoren gibt es in zwei Varianten, deren Funktion sich leicht unterscheidet: NPN-Transistoren werden durch positive Basis-Ströme durchgeschaltet, PNP-Transistoren durch negative Ströme. Negative Ströme fließen in die entgegengesetzte Richtung wie positive Ströme.

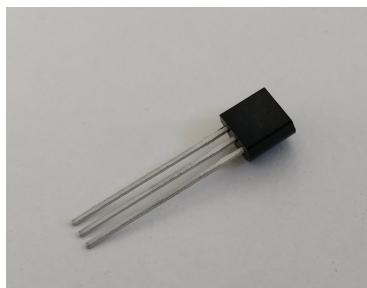
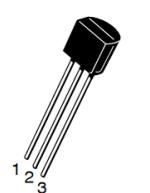
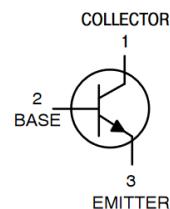


Abb. 10: Transistor



NPN: BC337



PNP: BC327

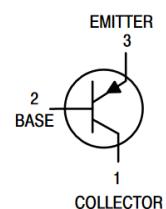


Abb. 11: Pinbelegung von BC337 und BC327

2.1.7 Reflexlichtschranken

Reflexlichtschranken (Abb. 12 und Abb. 13) können die Unterbrechung eines Lichtstrahls erkennen und die resultierende Stärke der Reflexion in ein elektrische Signal umwandeln. Sie bestehen aus einer Infrarot-Leuchtdiode und einem Phototransistor in einem gemeinsamen Gehäuse. Die Leuchtdiode sendet Strahlung im Infrarot-Bereich aus, die an Objekten, die sich im Strahlengang befinden, reflektiert wird. Je nach Helligkeitswert des Objektes wird ein bestimmter Anteil der Strahlung reflektiert, der dann einen Strom entsprechender Stromstärke im Phototransistor erzeugt.

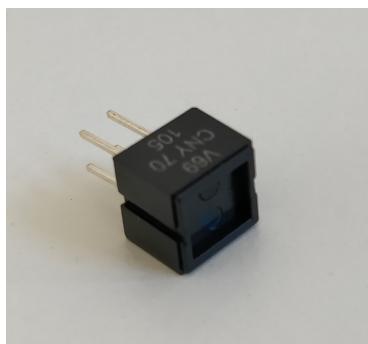


Abb. 12: Reflexlichtschranke (CNY70)

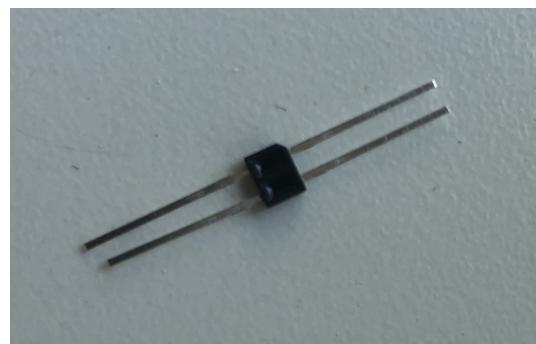


Abb. 13: Reflexlichtschranke (ITR8307)

2.1.8 Spannungsregler

Ein Spannungsregler (Abb. 14) stabilisiert eine variable Eingangsspannung und gibt eine feste Ausgangsspannung aus. Der Regler im *Armuro* ist ein sogenannter Linearregler, bei dem die Eingangsspannung größer sein muss als die Ausgangsspannung. Die Verlustleistung wird dann in Abhängigkeit der Spannungsdifferenz zwischen Ein- und Ausgang in Wärme umgewandelt.

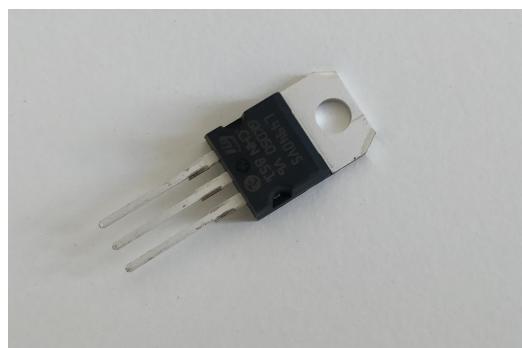


Abb. 14: Spannungsregler

2.1.9 Integrierte Schaltungen

Integrierte Schaltungen (ICs) sind Bauteile, die komplexe Schaltungen auf Halbleiterbasis realisieren. Dazu zählen beispielsweise Mikrocontroller und das Vierfach-Buffergate (Abb. 15), das im *Armuro* verwendet wird.

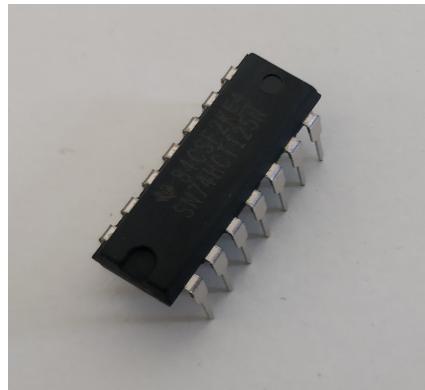


Abb. 15: Vierfach-Buffergate

2.1.10 Nucleo-Board

Das Herzstück des *Armuros* ist das Nucleo-Board (Abb. 16). Es wird auf zwei Buchsenleisten (Abb. 17) aufgesteckt, die an der Platine angebracht sind. Auf ihm befindet sich zur Steuerung des Roboters ein Mikrocontroller des Herstellers STMicroelectronics vom Typ STM32L432KC. Außerdem ist auf dem Board ein Programmieradapter enthalten, über den mit USB Programme vom Computer auf den Mikrocontroller aufgespielt und debuggt werden können.

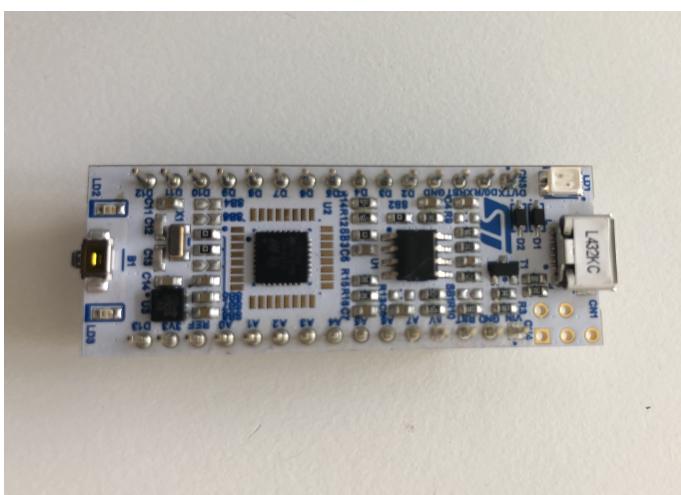


Abb. 16: Nucleo-Board

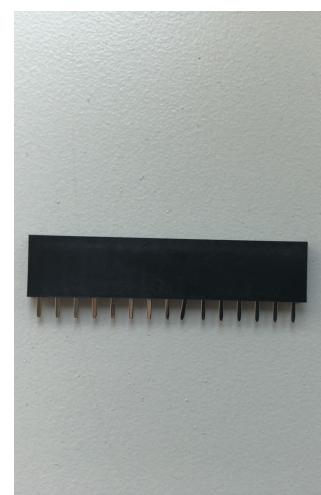


Abb. 17: Buchsenleiste

Als Taktgeber für den Controller wird ein 24 MHz Schwingquarz verwendet. Die meisten Pins des Microcontrollers sind mit den Pins des Boards verbunden und können per Software gesteuert werden und verschiedene Funktionen übernehmen, wie zum Beispiel die Ausgabe von PWM-Signalen oder das Einlesen von Analogwerten. Das Manual mit Schaltplan des Boards befindet sich unter:

https://www.st.com/resource/en/user_manual/um1956-stm32-nucleo32-boards-mb1180-stmicroelectronics.pdf

2.1.11 Räder

Zur Fortbewegung verfügt der *Armuro* über zwei Räder, die mit einem 3D-Drucker hergestellt wurden. Die vorgefertigten Räder bestehen aus durchsichtigem Plastik und haben Rillen auf ihrer Innenseite. Diese Rillen sind für das linke und rechte Rad spiegelverkehrt zueinander, daher sind die Außenseiten der Räder mit „links“ und „rechts“ gekennzeichnet.

Um die Räder in vollem Umfang nutzen zu können, müssen sie zunächst richtig vorbereitet werden. Die Reflexlichtschranken werden später direkt vor den Rädern angebracht. Daher müssen die Räder abwechselnd schwarz und weiß angemalt werden, sodass die unterschiedlichen Helligkeitswerte unterschiedliche Ströme in den Lichtschranken auslösen können. **Dabei ist unbedingt darauf zu achten, dass die Täler der Räder weiß und die ebenen Flächen schwarz angemalt werden.** Andersherum wird die Funktionalität stark beeinträchtigt. In Abbildung 18 sieht man ein Beispiel für die fertig angemalten Räder. Bei Bedarf kann die Mitte noch verzerrt werden, es ist aber für die Funktionalität der Räder nicht notwendig.



Abb. 18: Richtig angemalte Räder

2.1.12 Motoren

Damit sich die Räder drehen können, werden auf der Platine kleine Motoren (Abb. 19) angebracht, auf die die Räder aufgesteckt werden können.

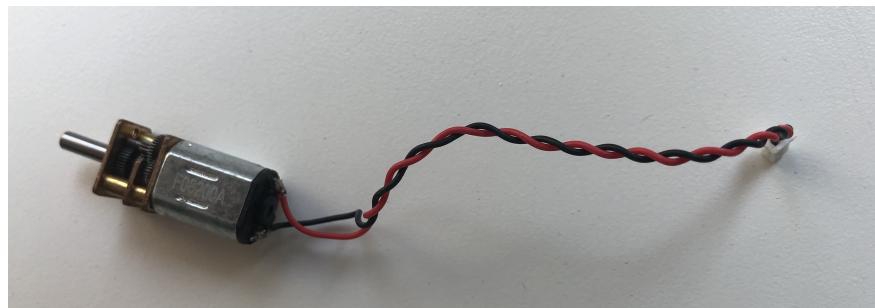


Abb. 19: Motor

2.1.13 Batterie

Die Stromversorgung des *Armuro* erfolgt über USB-Verbindung vom Computer. Außerdem kann der Roboter mittels Batterien versorgt werden. Diese sind in einem Batteriehalter (Abb. 21) mit zugehörigem Batterieclip (Abb. 20) an der Platine befestigt.



Abb. 20: Batterieclip



Abb. 21: Batteriehalter

2.2 Lötvorgang

Für den Aufbau des *Armuro* werden die einzelnen Bauteile auf die Platine gelötet. Daher wird in diesem Abschnitt kurz in die Technik des Löten eingeführt. Eine ausführliche Anleitung findet sich im ILIAS-Kurs unter „ERSA_Loetfibel“.



Abb. 22: Lötkolben, Metallwolle und weiteres Zubehör



Abb. 23: Besondere Vorsicht ist bei der Spitze des Lötkolbens geboten!

Beim Löten werden zwei leitende Materialien stoffschlüssig zusammengefügt mit dem Ziel, elektrische Leitfähigkeit und mechanische Stabilität zwischen den beiden Leitern herzustellen. Für das Löten wird ein Lötkolben (Abb. 22) verwendet, dessen Spitze auf bis zu 450°C erwärmt werden kann (Abb. 23).

Als Bindemittel wird eine bei niedrigen Temperaturen schmelzbare Metalllegierung, das sogenannte Lot, verwendet. Mit der Spitze des Lötkolbens werden dann die zu lötenden Teilstücke und das Lot erwärmt, bis das flüssige Lot in den Lötpalt fließt und dort fest wird. Dieser Vorgang ist schematisch in Abbildung 24 dargestellt. Durch das erstarrte Lot werden die Teilstücke zusammengehalten.



Abb. 24: Ablauf des Lötvorgangs

Das Material des Lots, auch als Lötzinn bezeichnet, ist eine Mischung aus Zinn, Silber und Kupfer. Im Kern des Lötdrahts befindet sich zusätzlich ein Flussmittel, das dafür sorgt, dass die Lötstelle homogener wird und anschließend verdampft. An der Lötpitze bildet sich durch verbranntes Flussmittel mit der Zeit eine schwarze Kruste, die an einem Schwamm oder der Metallwolle (Abb. 22) abgestreift werden sollte. Daher und um die Wärmeverluste der Bauteile zu reduzieren, sollte möglichst zügig mit maximal zehn Sekunden Dauer gelötet werden. **Die Flussmitteldämpfe sind reizend und sollten daher nicht eingetauscht werden. Außerdem müssen nach dem Löten unbedingt die Hände gewaschen werden.**

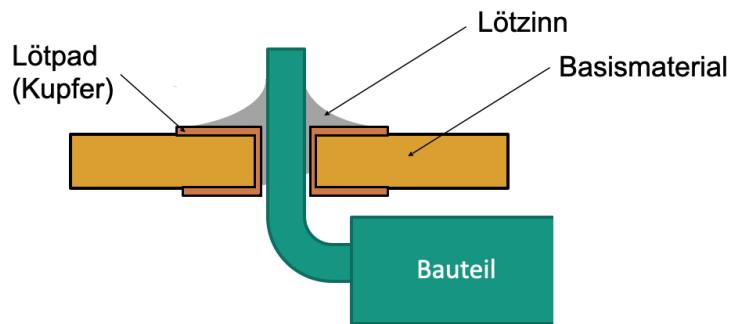


Abb. 25: Anordnung der Komponenten beim Löten

Für den erfolgreichen Lötvorgang werden zunächst Bauteil und LötPad mit der Lötspitze berührt. Anschließend wird der Lötdraht hinzugegeben bis der Lötzinn schmilzt und zwischen LötPad und Bauteil fließt. Danach werden Lötspitze und Lötdraht entfernt. Nach etwa drei Sekunden hat sich die Lötstelle hinreichend abgekühlt. Ein Schema dieser Anordnung befindet sich in Abbildung 25.

2.3 Aufbau

In diesem Abschnitt wird der Aufbau des *Armuro* Schritt für Schritt beschrieben. Die hier aufgeführte Reihenfolge sollte unbedingt eingehalten werden, damit der Fortschritt beim Zusammenbau thematisch mit den Kolloquien und Übungsblättern übereinstimmt. Viele Bauteile haben Besonderheiten, die im Laufe des Praktikums genauer behandelt werden.

In den Abbildungen 26 und 27 sind die Layouts der Ober- und Unterseite der Platine aus Abbildung 1 dargestellt. Fast alle elektrischen Bauteile werden von oben eingesteckt und an der Unterseite angelötet. Ausnahmen werden beim jeweiligen Schritt vermerkt. Positionen und Bezeichnungen (z.B. R17, C3, etc.) der Bauteile sind auf der Platine mit weißem Bestückungsdruck markiert.

Beim Verlöten der Bauteile ragen die Beinchen der Bauteile in der Regel noch über die Platine hinaus. Daher werden sie direkt über der Lötstelle abgeknipst. **Die Lötstellen an den einzelnen Beinchen der Bauelemente dürfen sich nicht berühren, sonst kann es zu Kurzschläüssen kommen.**

2.3.1 Aufbauanleitung

Es folgt nun eine Schritt für Schritt Anleitung zum Aufbau des *Armuro*:

1. Die 10 kΩ Widerstände (Abb. 2) an den Stellen R9, R10, R11, R12, R22, R23, R24, R25 und R27 einzulöten. Dabei müssen zuerst die Beinchen im rechten Winkel abgeknickt werden, um durch die Löcher in der Platine zu passen.

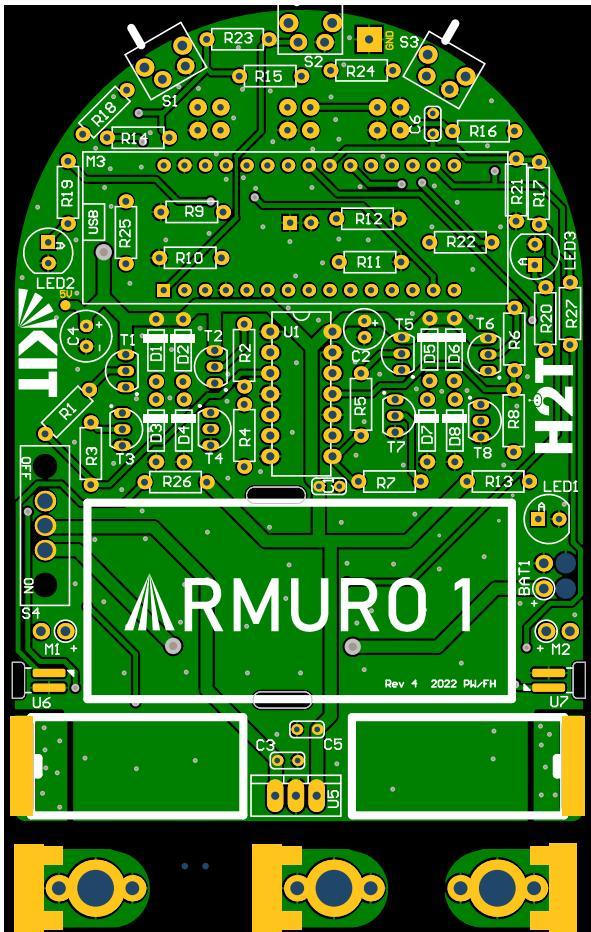


Abb. 26: Layout der Platinenoberseite

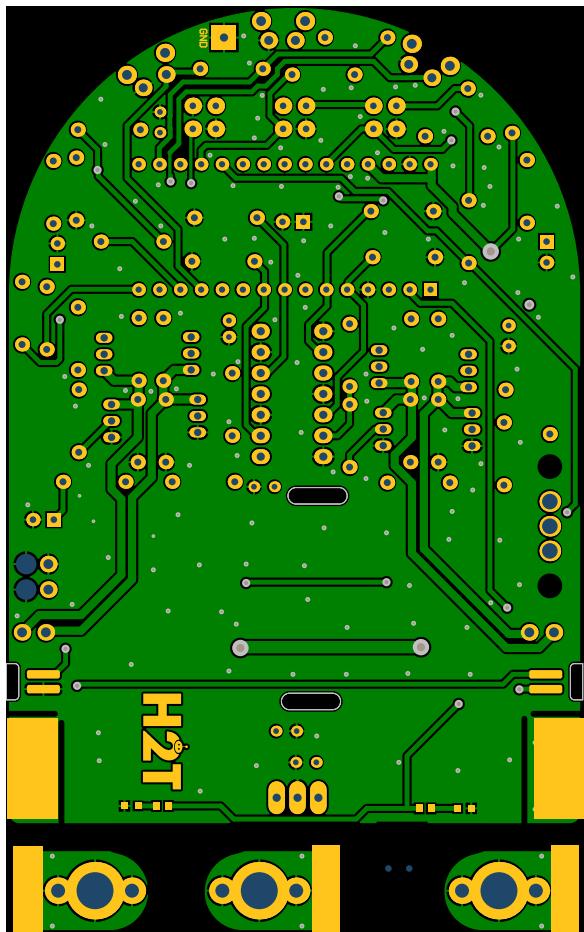


Abb. 27: Layout der Platinenunterseite

2. Die $1\text{ k}\Omega$ Widerstände an den Stellen R1, R2, R3, R4, R5, R6, R7, R8, R13, R17 und R21 einlöten.
 3. Die $4.7\text{ k}\Omega$ Widerstände an den Stellen R14, R15 und R16 einlöten.
 4. Den 47Ω Widerstand an der Stelle R18 einlöten.
 5. Die 120Ω Widerstände an den Stellen R19, R20 und R26 einlöten.
 6. Die Dioden 1N4148 (Abb. 6) an den Stellen D1, D2, D3, D4, D5, D6, D7 und D8 einlöten. Dabei die **Polarität** beachten. Der Streifen auf der Diode muss in Richtung des Streifens auf der Platine zeigen.
 7. Das Vierfach-Buffergate 74HCT125 (Abb. 15) an der Stelle U1 einlöten. Dabei die **Polarität** beachten. Der Halbkreis am Gehäuse zeigt in die gleiche Richtung wie der Halbkreis auf der Platine.
 8. Die 100 nF Keramikkondensatoren (Abb. 4) an den Stellen C1, C3, C5 und C6 einlöten.

9. Die Taster (Abb. 9) an den Stellen S1, S2 und S3 einlöten. **Dabei dürfen die Plastikteile des Schalters und insbesondere der Stößel nicht mit dem Lötkolben berührt werden.**
10. Die PNP-Transistoren (Abb. 11, BC327) an den Stellen T1, T2, T5 und T6 einlöten. Dabei die **Polarität** beachten. Die gerade Seite der Transistoren ist auf der Platine markiert. Eventuell müssen die Beinchen etwas zurechtgebogen werden.
11. Die NPN-Transistoren (Abb. 11, BC337) an den Stellen T3, T4, T7 und T8 einlöten. Dabei die **Polarität** beachten. Eventuell müssen die Beinchen etwas zurechtgebogen werden.
12. Die grünen LEDs (Abb. 7) LED1, LED2 und LED3 einlöten. Dabei die **Polarität** beachten. Die abgeflachten Seiten der LEDs sind auf der Platine markiert.
13. Die beiden Buchsenleisten (Abb. 17) für das Nucleo-Board an der markierten Stelle in Abbildung 28 einlöten. Dabei sollten die Leisten nicht verkippen, sondern senkrecht auf der Platine stehen. Dafür sollten zuerst zwei einzelne Pins eingelötet und anschließend die Ausrichtung nochmals geprüft werden. Wenn die Ausrichtung stimmt, kann der Rest der Pins angelötet werden.

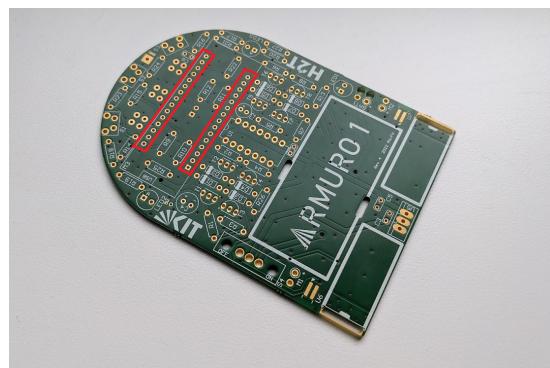


Abb. 28: Position der Buchsenleisten

14. Den 10 μF Elektrolytkondensator (Abb. 5) an der Stelle C2 einlöten. Dabei die **Polarität** beachten. Der Kondensator hat eine Markierung für den Minus-Pol. Außerdem ist das Beinchen am Minus-Pol kürzer. Auf der Platine ist der Plus-Pol markiert.
15. Den 470 μF Elektrolytkondensator an der Stelle C4 einlöten. Dabei die **Polarität** beachten.
16. Den Schiebeschalter (Abb. 8) an der Stelle S4 mit der Schalterposition auf **off** einlöten.
17. Den Spannungsregler (Abb. 14, L4940V5) an der Stelle U5 einlöten. Dabei die **Polarität** beachten. Der metallische Kühlkörper zeigt nach vorne.

18. Die beiden Reflexlichtschranken (Abb. 13, ITR8307) an den Stellen U6 und U7 einlöten. Dabei die **Polarität** beachten. Die Lichtschranke hat eine abgeschrägte Ecke, die auch auf der Platinen-Oberseite eingezeichnet ist. Dieses Bauteil wird seitlich an die Platine montiert. Dazu werden die zwei Beinchen horizontal auf der Oberseite aufgelötet und die unteren beiden Beinchen auf der Unterseite gelötet, wie in Abbildung 29 zu sehen ist. Zudem sollte die Vorderseite der Lichtschranke möglichst orthogonal zur Platine sitzen, sodass die Encoderscheiben auf den Rädern zuverlässig ausgelesen werden können.

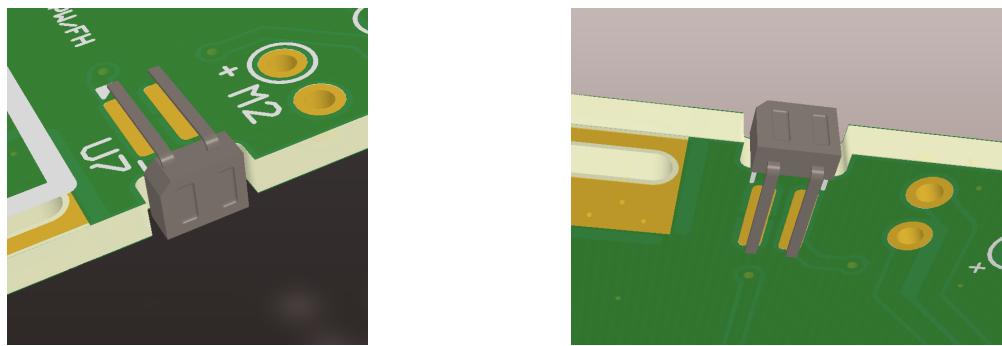


Abb. 29: Angebrachte Reflexlichtschranke von oben (links) und unten (rechts)

19. Die drei Reflexlichtschranken (Abb. 12, CNY70) auf der **Unterseite der Platine** an den Stellen U2, U3 und U4 einlöten. Dabei die **Polarität** beachten. Die Seite der Bauteile mit der Beschriftung liegt an der Seite mit der eingezeichneten dicken Linie auf der Platine auf.
20. Die kleine runde Teilplatine an der Rückseite der Platine abbrechen. Die kleine Teilplatine auf die Beinchen der grünen LED aufstecken und mit diesem Abstandshalter bei LED4 auf der **Unterseite** einlöten.
21. Die drei hinteren Teilplatten für die Motorhalterung (Abb. 30) vorsichtig vom PCB abbrechen.
(Es werden nur zwei benötigt, eine ist Reserve)

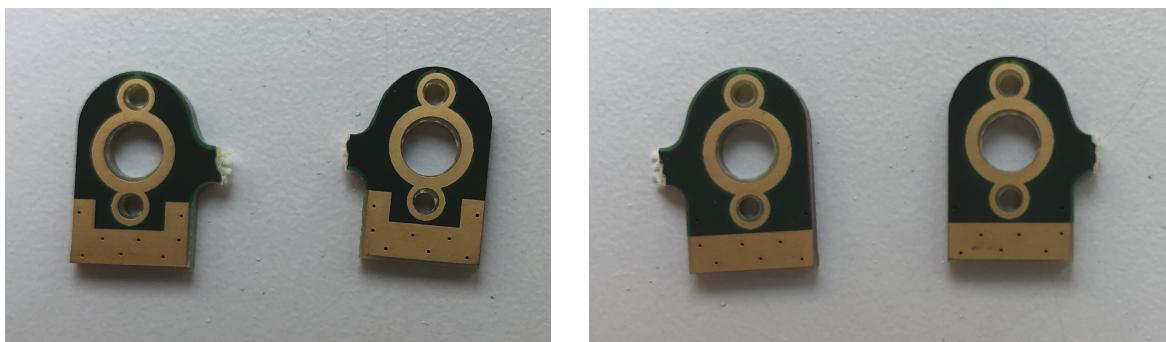


Abb. 30: Außen- (links) und Innenseite (rechts) der kleinen Teilplatten für die Motorhalterung

22. An jede der beiden kleinen Platten auf der Innenseite (**Seite mit dem goldenen Strich**, nicht

mit der goldenen „eckige Klammer“ einen Motor anschrauben. Die richtige Anbringung ist in Abbildung 31 dargestellt.

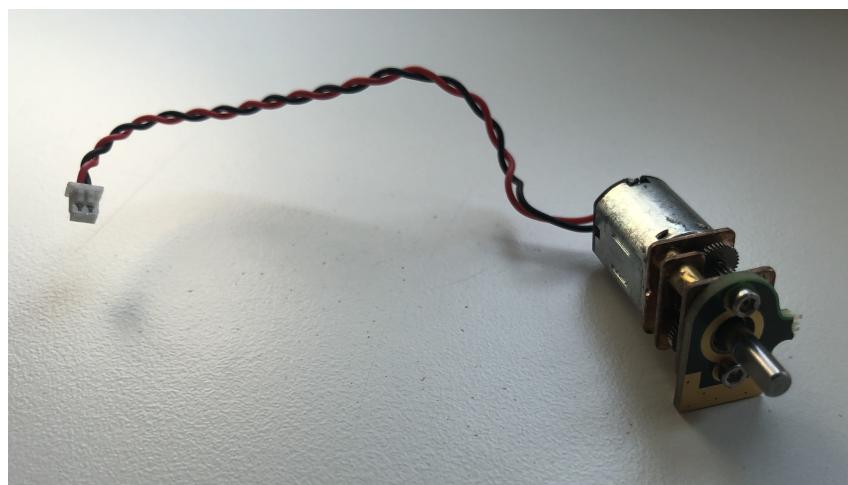


Abb. 31: Richtige Anbringung des Motors an der Platine

23. Ein rotes und schwarzes ca 7cm langes Kabel an die Kontakte der Motoren löten. Das rote Kabel wird an den mit „+“ markierten Kontakt gelötet. Vorsicht, die Kunststoffrückseite des Motors ist hitzeempfindlich. An den Enden der Kabel werden dazu 2 mm der Isolierung entfernt, die einzelnen Kupfer-Adern verdrillt und verzinnt. Danach die Kabel an den Motorkontakt anlöten. Zum Abisolieren kann die bereitgestellte Abisolierzange, die Fingernägel oder ein Messer verwendet werden.
24. Die Teilplatinen mit den Motoren in die dafür vorgesehenen Schlitze am hinteren Ende der Hauptplatine einstecken. Die Motoren sollten auf der Platine aufliegen.
25. Die beiden Platinen auf der Unterseite am Schlitz miteinander verlöten.
26. Die Kabel der Motoren an den Stellen M1 (linker Motor) und M2 (rechter Motor) auf der Platine einlöten. Das jeweils rote Kabel wird in das mit „+“ markierte Pad eingelötet, das schwarze Kabel in das Pad daneben.
27. Beide Kabel des Batterieclips durch die zwei Löcher bei BAT1 von oben durchstecken. Danach von der Platinenunterseite das rote Kabel des Batterieclips in das mit „+“ beschriftete Pad an der Stelle BAT1 einlöten, das schwarze Kabel in das Pad daneben. Der Batteriehalter kann nun am Batterieclip befestigt werden.
28. Das Nucleo-Board auf die Buchsenleisten stecken. Dabei die **Polarität** beachten. Der USB-Port ist auf der Platine markiert.
29. Zu den Betreuern des Praktikums gehen, um den *Armuro* in Betrieb zu nehmen. Bitte **noch nicht einschalten**, solange kein Betreuer darüber geschaut hat!
30. Batterien eingelegen und den Batteriehalter mit einem Klettband am *Armuro* fixieren.
31. Räder anmalen, O-Ringe auf die angemalten Räder aufziehen und bis zum Anschlag auf die Achse stecken. Auf die Beschriftung achten, es gibt ein rechtes und linkes Rad. Die angemalte Seite muss nach innen zeigen.
32. Der *Armuro* ist fertig und kann nun programmiert werden!

Optional: Heutige Elektronik wird nur noch sehr selten mit Bauteilen zum Durchstecken entworfen. Stattdessen werden die Bauteile direkt auf der Oberfläche der Platine angebracht. Dies wird SMD-Technik (Surface Mounted Devices) genannt. Üblicherweise werden SMD-Bauteile maschinell bestückt, allerdings ist es durchaus möglich sie auch mit der Hand einzulöten. Um ein Gefühl für die heutige Platinenfertigung zu bekommen, ist auf der Unterseite eine SMD-Challenge eingebaut. Wenn der Roboter bereits komplett aufgebaut und getestet wurde, können auf der Unterseite ganz hinten noch die Stellen R28, R29, LED5 und LED6 verlötet werden. Bei den LEDs ist weiterhin die Polarität zu beachten. Der kleine Pfeil auf der Unterseite der LED muss mit dem auf der Platine übereinstimmen. Bei den Widerständen liegt die

schwarze Seite oben.

2.3.2 Schaltplan

Der Roboter ist nun fertig aufgebaut und kann in Betrieb genommen werden. Der Schaltplan auf Seite 19 zeigt an, wie die einzelnen Komponenten des Roboters miteinander verschaltet sind.

2.4 Inbetriebnahme

Um zu überprüfen, ob alles richtig zusammengebaut wurde, wird von den Tuto ren ein Testprogramm, das alle Funktionen des Roboters ansteuert, auf den Roboter aufgespielt. Dabei werden die einzelnen Schritte des Programms im Computer ausgegeben. Der *Armuro* muss daher während des Testlaufs über das Kabel verbunden bleiben.

Um die über USB gesendeten Daten außerdem am PCB empfangen zu können, wird ein Terminalprogramm benötigt. Dafür kann z.B. *HTerm* (downloadbar unter <https://www.der-hammer.info/pages/terminal.html>), verwendet werden. Für die Verbindung mit dem Programmer des Nucleo-Boards muss im Dropdown-Feld „Port“ des Programmes der richtige COM-Port ausgewählt. Dazu können alle möglichen Optionen durchprobiert werden. Außerdem muss die Baudrate im Dropdown-Feld „Baud“ auf 115200 eingestellt werden.

Im Folgenden werden die einzelnen Schritte des Testprogramms beschrieben. Die richtige Ausgabe im Terminalprogramm ist in Abbildung 32 dargestellt.

1. Begrüßung mit „Hello World“.
2. Ausgabe der Batteriespannung in mV. Um hier ein sinnvolles Ergebnis zu erhalten, muss der Schalter des Roboters auf **on** gestellt sein.
3. Überprüfung der LEDs: Beide vorderen LEDs leuchten nacheinander für etwa 4 Sekunden auf, zuerst links, dann rechts.
4. Überprüfung der Tastsensoren: Die Sensoren müssen von links nach rechts nacheinander gedrückt werden, damit das Programm weiterläuft.
5. Überprüfung der unteren Lichtsensoren: Die Lichtsensoren an der Unterseite des Roboters werden von links nach rechts für jeweils 12 Sekunden ausgelesen. Wird der Roboter in diesem Zeitraum auf den Tisch gestellt oder der jeweilige Sensor mit dem Finger zugehalten, leuchtet eine LED auf.
6. Überprüfung der Lichtsensoren neben den Rädern: Die Räder drehen sich nacheinander (zuerst links, dann rechts, das andere Rad bleibt still) für 6 Sekunden, sodass die Sensoren abwechselnd weiße und schwarze Bereiche messen. Wird der weiße Bereich detektiert, so leuchtet die LED auf der entsprechenden Seite auf. Wird der schwarze Bereich detektiert, geht die LED wieder aus.

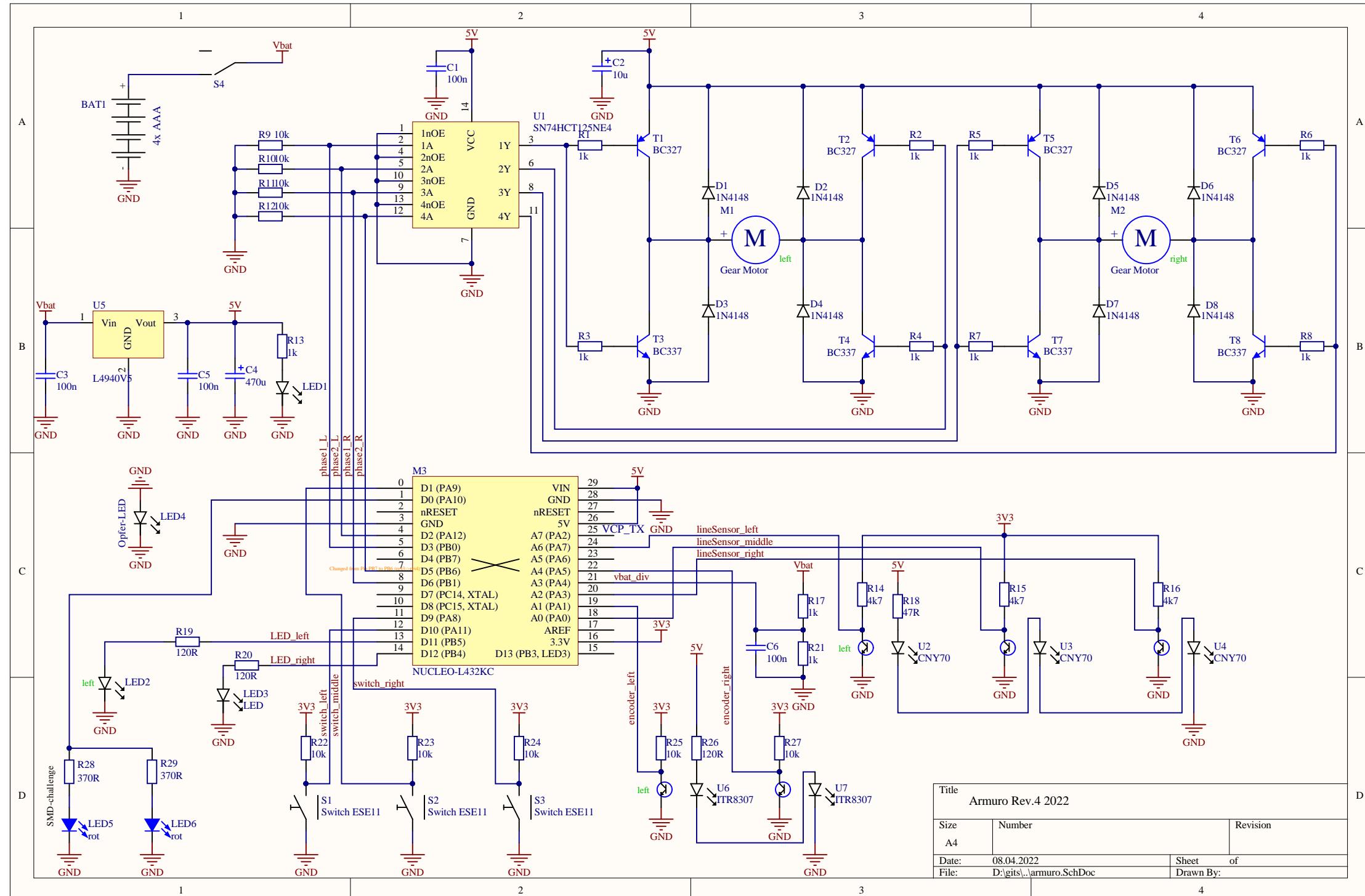
7. Überprüfung der Radmotoren I: Der Roboter fährt jeweils 4 Sekunden zweimal vorwärts und rückwärts, erst langsam, dann mit voller Geschwindigkeit.
8. Überprüfung der Radmotoren II: Der Roboter dreht sich jeweils 4 Sekunden erst nach links, dann nach rechts. Dabei fährt jeweils ein Rad vorwärts und ein Rad rückwärts und die LED auf Innenseite leuchtet auf.
9. Der Testdurchlauf wurde erfolgreich abgeschlossen!

```

1 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 9
hello world
.
.
.
battery: 4597 mV
.
.
.
left LED on
.
right LED on
.
touch sensor left at state 0
please press the touch sensor on the left
touch sensor left at state 1
.
touch sensor middle at state 0
please press the touch sensor in the middle
touch sensor middle at state 1
.
touch sensor right at state 0
please press the touch sensor on the right
touch sensor right at state 1
.
left LED on for touching the line sensor on the left or bringing the robot to the ground
LEDs on for touching the line sensor in the middle or bringing the robot to the ground
right LED on for touching the line sensor on the right or bringing the robot to the ground
.
left LED on for white wheel marks, LEDs out for black wheel marks
right LED on for white wheel marks, LEDs out for black wheel marks
.
drive forward
drive backward
drive forward
drive backward
.
turn left
.
turn right
.
test run done

```

Abb. 32: Ausgabe des Testprogramms zur Überprüfung der Funktionalität im Programm HTerm



3 Programmierumgebung

Im Folgenden wird die Programmierumgebung *STM32CubeIDE* vorgestellt, mit deren Hilfe der *Armuro* programmiert werden kann.

3.1 STM32Cube IDE installieren

Zur Installation können die erforderlichen Dateien auf der Entwicklerseite heruntergeladen werden:

<https://www.st.com/en/development-tools/stm32cubeide.html>

Danach wird den Anweisungen im Installationsfenster gefolgt. In Abbildung 33 sieht man dies für Windows (links) und macOS (rechts).



Abb. 33: Installierung

3.2 Ein Projekt erstellen

Nach der erfolgreichen Installation kann nun ein Projekt erstellt werden. Dafür wurde bereits eine ioc.-Datei mit allen nötigen Einstellungen des Mikrocontrollers vorbereitet und im ILIAS-Kurs hinterlegt. Die Datei wird nun heruntergeladen und erstmal im Downloads Ordner gespeichert. Die ioc.-Datei wird verwendet um basierend darauf ein Projekt mit der nötigen Konfiguration des Mikrocontrollers und den Hardware-Treibern zu erstellen. Zur Erstellung wird in *STM32CubeIDE* unter „File“ „New“ „STM32 Project from an existing STM32CubeMX Configuration File (.ioc)“ aufgerufen (Abb. 34).

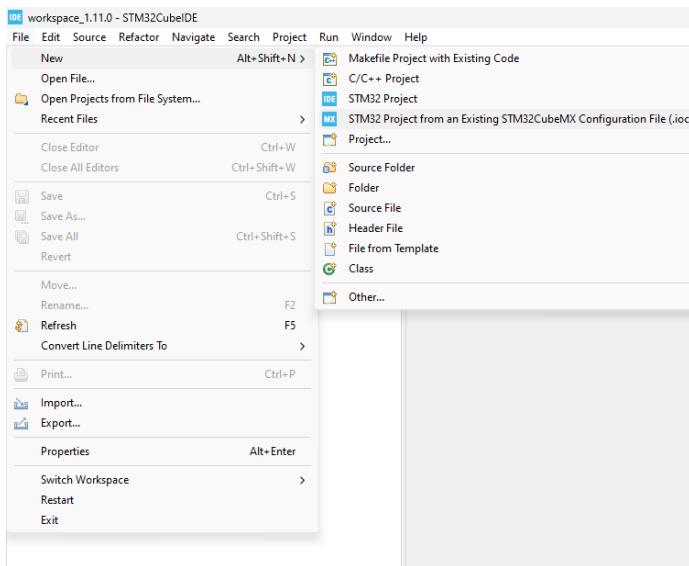


Abb. 34: Erstellen eines Projektes aus ioc.-Datei

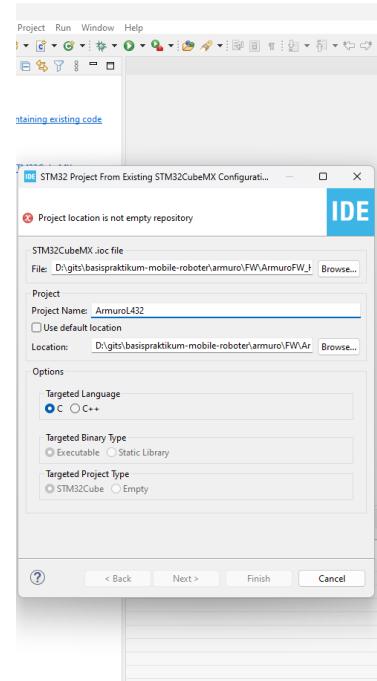
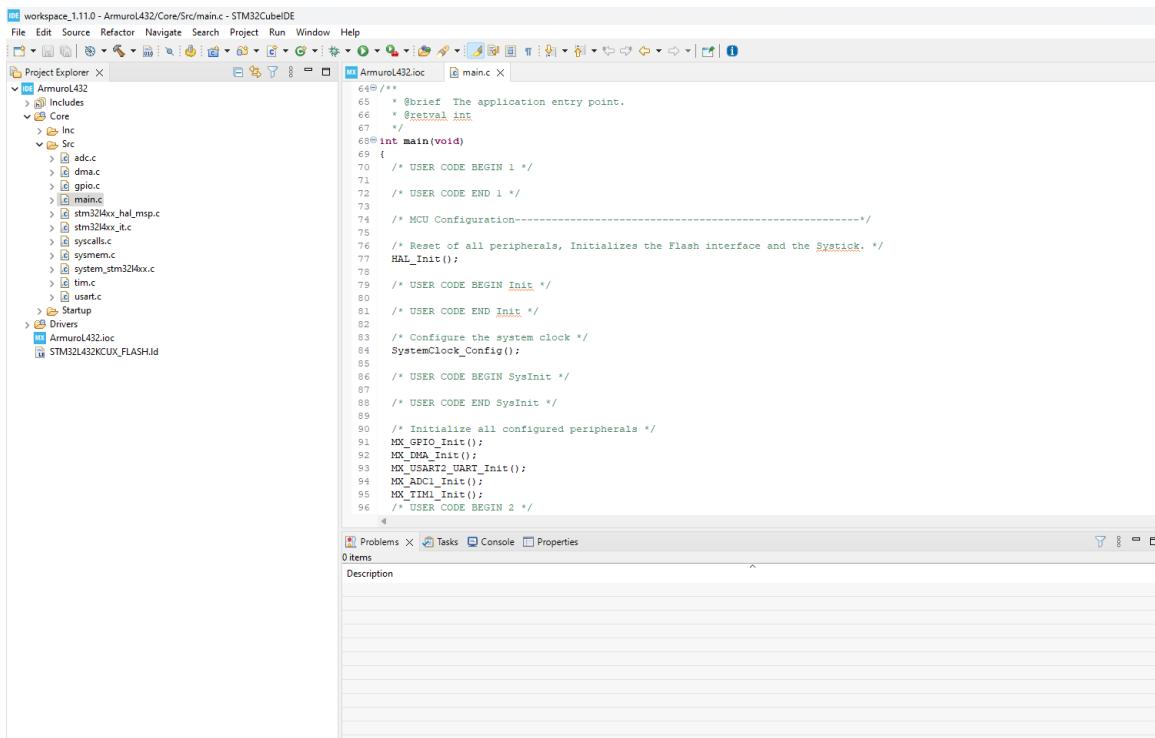


Abb. 35: Konfiguration

Im Konfigurations-Fenster (Abb. 35), welches sich nun öffnet, muss unter „File“ die ioc.-Daten angegeben werden. Der Haken bei Default-Location wird entfernt und ein geeigneter Dateipfad für das neue Projekt angegeben. Mit „Finish“ wird das Projekt dann erstellt.

3.3 Projektstruktur und Ausführung

In Abbildung 36 sieht man die Gesamtansicht der Benutzeroberfläche der *STM32CubeIDE*. In der oberen Leiste befinden sich die Werkzeuge zur Ausführung des Codes. Mit dem Symbol der Diskette kann das Projekt abgespeichert werden. Mit dem Hammer wird es gebaut, ohne dass der *Armuro* über das Kabel verbunden sein muss. So können Fehler im Code schnell bemerkt werden. Mit dem weißen Pfeil auf grünem Grund schließlich kann das Programm gebaut und auf den Roboter überspielt werden. Dazu muss der Roboter über das USB-Kabel verbunden sein. Mit dem grünen Käfer gelangt man in den Debug-Modus.



The screenshot shows the STM32CubeIDE interface. The left pane is the Project Explorer, displaying the project structure for 'ArmuroL432'. It includes folders for Includes, Core, Inc, and Src, containing files like adc.c, dma.c, gpio.c, main.c, and system_stm32l4xx.c. The right pane shows the code editor for 'main.c'. The code is as follows:

```
64/* brief The application entry point.
65 * @retval int
66 */
67
68int main(void)
69{
70    /* USER CODE BEGIN 1 */
71
72    /* USER CODE END 1 */
73
74    /* MCU Configuration-----*/
75
76    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
77    HAL_Init();
78
79    /* USER CODE BEGIN Init */
80
81    /* USER CODE END Init */
82
83    /* Configure the system clock */
84    SystemClock_Config();
85
86    /* USER CODE BEGIN SysInit */
87
88    /* USER CODE END SysInit */
89
90    /* Initialize all configured peripherals */
91    MX_GPIO_Init();
92    MX_DMA_Init();
93    MX_USART2_UART_Init();
94    MX_ADC1_Init();
95    MX_TIM1_Init();
96    /* USER CODE BEGIN 2 */
97}
```

The bottom pane is the 'Problems' window, which is currently empty.

Abb. 36: main-Datei

Im linken Fenster der *STM32CubeIDE* ist die Projektstruktur des Projekts aufgeführt. In Abbildung 36 wurde die „main.c“-Datei ausgewählt. In dieser Datei wird der Hauptteil der Programmierarbeit stattfinden. Im unteren Fenster werden Fehler im Programmcode ausgegeben. Auch eine Kopie der ioc.-Datei ist im Projekt enthalten und kann eingesehen werden (Abb. 37).

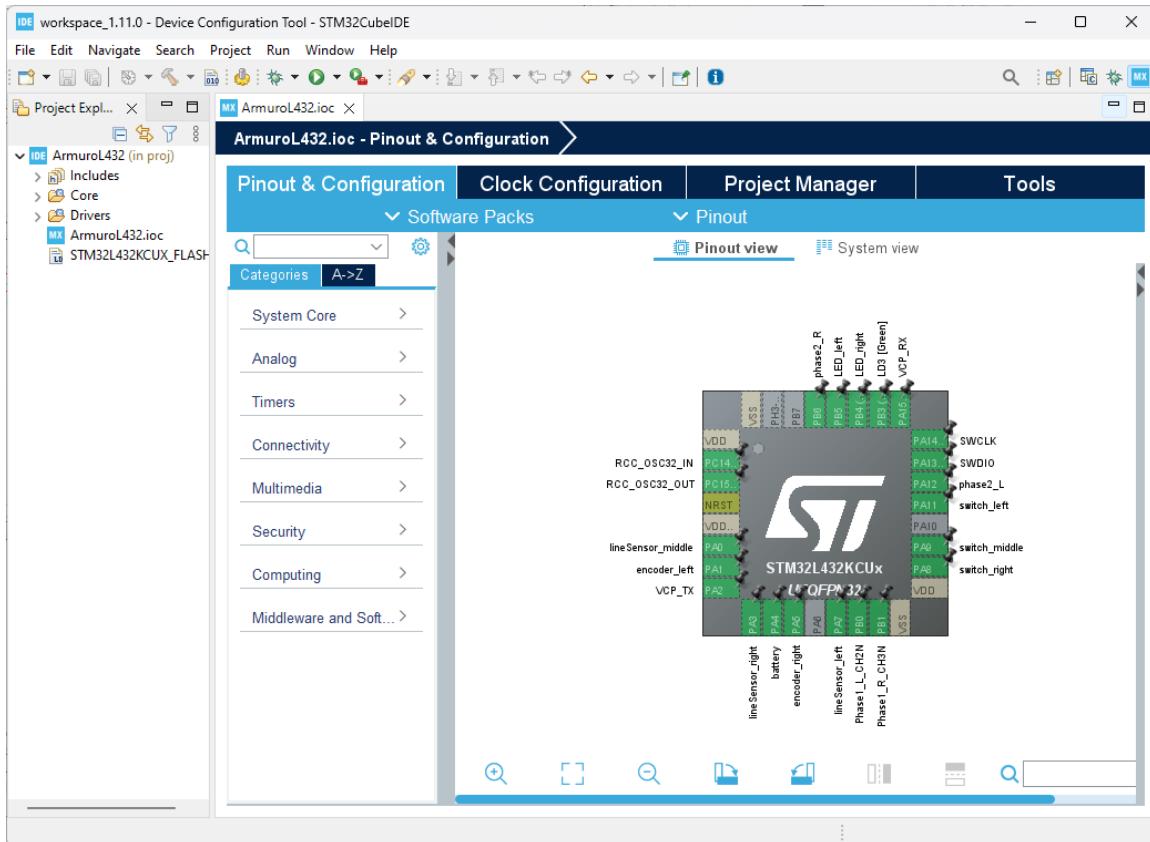


Abb. 37: ioc-Datei

3.4 Debug-Modus

Mithilfe des Debug-Modus können Fehler im Code leichter gefunden werden. Dazu werden sogenannte „breakpoints“ an den Stellen im Code gesetzt, die man genauer untersuchen möchte. Mit dem grünen Pfeil kann man dann das Projekt solange durchlaufen lassen, bis man an einen breakpoint gelangt. Mit den drei gelben Pfeilen „step in“, „step over“ und „step return“ (Die Namen werden angezeigt, sobald man mit der Maus darüber fährt.) kann das Programm dann schrittweise durchlaufen werden.



Abb. 38: Debug-Modus