

Arduino-Mikrocontroller-Projekte mit KI-Unterstützung Vorbereitung

Jonathan Herter

June 5, 2025



1 Ablauf des Praktikums

Innerhalb dieses Versuchs sollen Sie mithilfe von Künstlicher Intelligenz (KI oder im Englischen AI) einen Mikrocontroller programmieren. Mit einem Mikrocontroller kann man unterschiedliche Sensoren ansteuern, um Daten zu erfassen oder Komponenten zu steuern. So können z.B. Wetterstationen oder Roboter gebaut werden. Im Praktikum sollen Sie lernen, wie Sie kleine Projekte selbst realisieren können. Dazu wird ein Arduino Uno verwendet. Als KI kommt 'Copilot'¹ zum Einsatz. Zu diesem haben alle KIT-Studierenden Vollzugang. So werden neben dem Wissen zum Mikrocontroller auch Kompetenzen zur Verwendung von KI, insbesondere großen Sprachmodellen wie ChatGPT oder Copilot, vermittelt.

Im Folgenden wird zuerst kurz erklärt, was ein ‚Large Language Model‘ (LLM) ist und wie es grob funktioniert. Anschließend werden vier Patterns vorgestellt. Ein Pattern ist dabei ein Kommunikationsmuster, wie man Anfragen oder Chats mit einer KI führen kann, was zu besseren Antworten der KI führt. In diesen Patterns werden sogenannte Prompts vorgestellt – also Benutzereingaben, mit denen man die KI steuert. Sie helfen dabei, den Chat produktiver und zielgerichteter zu gestalten.

Für diesen Versuch wird empfohlen, Copilot zu verwenden. Die vermittelten Kompetenzen lassen sich auch auf andere KIs wie ChatGPT anwenden. Da es aber Unterschiede gibt, wie gut die einzelnen LLM in unterschiedlichen Anwendungsbereichen sind, empfiehlt es sich, die LLM zu verwenden, mit welcher dieser Versuch erstellt wurde.

Zur Vorbereitung gibt es dann noch ein Arbeitsblatt, welches Sie bearbeiten sollen. Vor Ort stehen Ihnen viele unterschiedliche Sensoren und weitere Bauteile zur Verfügung, um Projekte abzuschließen. So können Sie vor Ort unterschiedliche Projekte bearbeiten und Ihre eigenen Ideen umsetzen.

2 Was sind Large Language Models?

Nicht jede KI ist gleich KI. Künstliche Intelligenz ist dabei nur ein Überbegriff. Eine KI wie ChatGPT oder Copilot ist anders als etwa die bildgenerierenden KIs. Hier verwenden wir eine Untergruppe, die Large Language Models (LLMs) genannt wird. Ihre Funktionsweise ist, dass man mit ihnen chatten kann wie mit einer normalen Person. Die LLMs sind dann darauf trainiert, möglichst zufriedenstellende Antworten zu generieren. Hierbei bekommt die KI kein tieferes Verständnis über den Inhalt oder die Richtigkeit der produzierten Inhalte. Die KI generiert dabei iterativ ein Wort nach dem anderen. Für jedes neue Wort wird eine Wahrscheinlichkeitsverteilung über alle möglichen Wörter erstellt, und dann wird eines der Wörter mit der höchsten Wahrscheinlichkeit ausgewählt. Sobald die KI mit einem Wort fertig ist, generiert sie das nächste Wort. Dabei können inhaltliche Fehler passieren und infolgedessen falsche Antworten generiert

¹Copilot ist für KIT-Angehörige gratis. Alle Informationen zur Freischaltung und Nutzung finden sie auf der SCC Seite. <https://www.scc.kit.edu/dienste/copilot.php>

werden. So muss jede Antwort vom Nutzer auf Richtigkeit überprüft werden. Ein Beispiel hierfür ist, wenn man die Frage stellt, wie häufig der Buchstabe ‚r‘ in dem Wort „Strawberry“ vorkommt. Zu sehen ist das zum Beispiel unten im Bild (1). Es ist hier aber zu sagen, dass, wenn solche Fehler passieren, diese in das Trainingsset aufgenommen werden, um die nächste Generation an LLMs besser zu machen. So taucht der Fehler bei neueren LLMs nicht mehr auf, ist aber noch bei ChatGPT 3.5 zu beobachten.

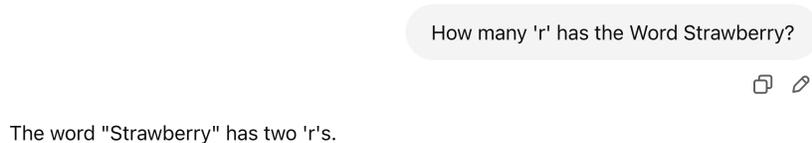


Abbildung 1: ChatGPT 4o mini Dialog vom 22.4.25

3 Patterns

Wir wollen uns jetzt genauer anschauen, wie wir sinnvoll mit einer KI arbeiten können. Intuitiv geht das eigentlich relativ leicht. Man kann sich das vorstellen wie ein Gespräch mit einem anderen Menschen. So bekommt man bereits gute Antworten, wenn wir einfach Fragen an die KI stellen. So wird in vielen Fällen die KI verwendet. Es kann nun aber mit Patterns gearbeitet werden, welche den Input (was wir eingeben) oder Output (was die KI uns antwortet) verbessern. Hier werden jetzt vier davon vorgestellt.

Zu allen vorgestellten Patterns sind Beispiele im Anhang zu finden. Diese sollen auch für die Bearbeitung der Aufgaben genutzt werden.

Diese Patterns sind aus einer wissenschaftlichen Publikation, welche mehrere Patterns zusammen gefasst hat. Gerne können Sie die Publikation bei Interesse lesen.

Titel: A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT
Quelle: <https://www.dre.vanderbilt.edu/~schmidt/PDF/prompt-patterns.pdf>

3.1 The Flipped Interaction Pattern

Bei einer normalen Interaktion mit einer KI steht normalerweise der Nutzer als treibendes Konversationsglied im Fokus. Das heißt, der Nutzer stellt die Fragen und die KI reagiert darauf mit einer Antwort. Die Idee in dem Flipped Interaction Pattern ist, das umzudrehen. So wird die KI aufgefordert, den Nutzer so lange (oder eine begrenzte Anzahl an) Fragen zu stellen, um ein Ziel zu erreichen. Die Motivation hierbei ist, dass die KI weniger 'educated guesses' machen muss und einfach den Nutzer fragen kann. Das kann zum Beispiel bei einer Programmieraufgabe gemacht werden. So kann sie z.B. nach der Programmiersprache oder weiteren Details fragen. Durch das Nachfragen kann die KI weitere

Informationen erhalten, die der Nutzer vergessen hat zu geben. So kann sich die KI ein vollständigeres Bild davon machen, was sie tun soll, und ein besseres Ergebnis liefern.

Aufbau

Sprachlicher Aufbau
Ich möchte, dass du mir Fragen stellst, um X zu erzielen.
Frage mich so lange Fragen, bis du ausreichend Informationen hast, um das Ziel zu erreichen.
(Optional) Frage immer eine (zwei, drei, ...) Fragen auf einmal.

Der erste Teil ist dafür da, um das Ziel an die KI zu übermitteln. Das ist wichtig, damit die Fragen in die richtige Richtung gehen und die KI auch Fragen weglässt, welche sie als unnötig sieht.

Der zweite Teil ist dafür da, um zu steuern, dass die KI mit den Fragen auch irgendwann aufhört und das Ziel erstellt. Wenn das fehlt, kann es dazu kommen, dass die KI immer mehr und mehr Fragen stellen wird. Das kann aber auch abgebrochen werden durch den Nutzer mit der Eingabe "Höre auf, weiter Fragen zu stellen."

Die dritte Eingabe ist optional. Sie dient lediglich der Art und Weise, wie die Fragen gestellt werden. Dabei ist es persönliche Präferenz, ob man alle Fragen auf einmal bekommen möchte zum Beantworten oder immer nur ein paar. Wenn man diesen Teil weglässt, entscheidet sich die KI zufällig für eine Variante.

3.2 Persona Pattern

Wenn man Fragen oder Probleme hat, wendet man sich oft an erfahrene Personen, die weiterhelfen können. Diese Personen haben meistens mehr Wissen oder Erfahrung, um einem weiterzuhelfen. Dies wird sich in diesem Pattern zunutze gemacht. Wir sagen der KI, dass sie sich wie eine bestimmte Person sieht, und aus dieser Sichtweise Antworten generieren soll. Mit dieser angenommenen Person kann die KI dann bessere Antworten generieren. So geht man davon aus, dass, wenn sie eine Lehrpersönlichkeit annimmt, Sachverhalte besser oder verständlicher erklärt oder als ein Experte neue Sichtweisen einbringt, die man selbst vielleicht vergessen hat zu bedenken. Dieses Prinzip kann weitergedacht werden – die KI könnte sich zum Beispiel auch als Baum oder Tier vorstellen, um kreative Perspektiven zu eröffnen.

Aufbau

Sprachlicher Aufbau
Agiere als Person X
Erstelle Antworten, welche ein X geben würde.

Persona X oder nur das X kann hierbei z. B. durch einen Beruf ersetzt werden. Die Person sollte eine Reihe von konkreten Eigenschaften aufweisen, die

mit einer bekannten Berufsbezeichnung, einem bekannten Personentyp usw. verbunden sind. Bei unklaren Personas, z.B. Schwester, ist es nicht eindeutig, wie die KI auf Fragen antworten soll.

3.3 The Refine Question Pattern

The Refine Question Pattern ist nützlich, wenn man sich nicht sicher ist, wie eine gute Frage formuliert sein könnte. So gibt man der KI die Aufgabe, zuerst eine bessere Frage zu formulieren. Dabei gibt die KI im Allgemeinen bessere und auch spezifischere Fragen zurück, aus welchen der Nutzer dann wählen kann. Dies hat am Ende den Effekt, dass der Nutzer mit weniger Fragen zum Ziel kommen kann. Gerade wenn der Nutzer kein Experte in dem Bereich ist, in welchem er Fragen stellt, kann die KI weitere Fachbegriffe benutzen, um eine bessere Version der Frage zu erstellen. An dieser Stelle kann die KI auch nach mehr Informationen fragen (The Flipped Interaction Pattern).

Aufbau

Sprachlicher Aufbau
In dem Anwendungsbereich X, schlage mir eine bessere Fragestellung vor, die ich verwenden kann.
(Optional) Frage mich, ob ich die bessere Fragestellung verwenden soll.

Im ersten Teil wird der KI gesagt, wie sie sich mit Fragen zu einem spezifischen Themenbereich verhalten soll. Der Zweite dient zur schnellen Weiterverwendung, wenn das gewünscht ist.

Dieses Pattern lässt sich auch hervorragend mit anderen Patterns verknüpfen. So kann auch das Flipped Interaction Pattern genutzt werden, um die KI aufzufordern, Fragen an den Nutzer zu stellen, um weitere Informationen zu erlangen, um eine bessere Version der Frage zu erstellen.

3.4 The Cognitive Verifier Pattern

Bei schwierigen Themen kann es sehr hilfreich sein, wenn auf eine Frage zuerst Rückfragen kommen, um zu schauen, auf welchem Wissensstand die fragende Person ist, um eine gute Antwort zu generieren. Genau dieses Vorgehen soll hier die KI übernehmen. So wird sie zunächst aufgefordert, weitere Fragen zu generieren, welche der Nutzer beantworten soll. Mit den Antworten soll dann die eigentliche Frage des Nutzers beantwortet werden. Dies ist besonders dann praktisch, wenn der Nutzer kein Experte auf dem Gebiet ist. So gibt der Nutzer seinen aktuellen Wissensstand an die KI weiter. Dies erleichtert auch das Formulieren der Frage. Diese könnte bereits mit allen relevanten Informationen gefüllt werden, was jedoch aufwendig ist. Hier kann auch schnell mal eine wichtige Information vergessen werden. Mit dem Cognitive Verifier Pattern fragt die KI selbst nach allen Informationen und dem Wissensstand nach und kann so eine besser verständliche Antwort generieren.

Aufbau

Sprachlicher Aufbau
Wenn dir eine Frage gestellt wird, halte dich an die folgenden Regeln.
Generiere zusätzliche Fragen, welche ich dir beantworte und dir helfen, die Hauptfrage besser zu beantworten.
Fasse die einzelnen Antworten zusammen, um eine abschließende Antwort auf die Hauptfrage zu erstellen.

Im ersten Teil wird die KI darauf hingewiesen, dass sie eine Regel hat, an die sie sich stets halten soll. Dieser Abschnitt kann auch modifiziert werden, sodass die Regel nur für ein bestimmtes Thema gilt. Die zwei weiteren Teile legen dann die genaue Regel fest. Zunächst wird die KI dazu aufgefordert, weitere Fragen an den Nutzer zu stellen, um mehr Informationen zu erhalten, die ihr bei der Beantwortung der ursprünglichen Frage helfen. Im letzten Teil soll sie basierend auf den gesammelten Informationen eine Antwort auf die ursprüngliche Frage formulieren.

4 Aufgaben

Um die folgenden Fragen zu beantworten, nutze bitte den Prompt, der den Embedded-Systems-Ingenieur beschreibt. Dieser ist im Anhang enthalten. Notiere dir außerdem, was besonders gut funktioniert hat, was weniger gut war und wo die KI möglicherweise Fehler gemacht hat.

1. Hardwareaufbau:
 - a) Wodurch zeichnet sich ein Mikrocontroller aus?
 - b) Wodurch zeichnet sich die Hardware eines Arduino aus?
 - c) Was sind die Hauptunterschiede zu einem Desktop-PC? In welchen Projekten verwendet man lieber einen Arduino bzw. Mikrocontroller?
2. Software:
 - a) Wie ist der grundlegende Aufbau eines Arduino-Programms?
 - b) Wofür werden die zwei Methoden benötigt, und was ist ihre Funktion?
3. Kontrollstrukturen:
 - a) Welche Funktionen haben die Kontrollflussstrukturen if, while, switch?
 - b) Wie wird eine Variable auf dem Arduino angelegt?
 - c) Was sind Methoden? Wie werden sie im Programm definiert?

4. Finden Sie mithilfe des Beispiel-Prompts vom Cognitive Verifier Pattern heraus, was der folgende Code macht und erklären Sie ihn kurz.

```
1 const int potPin = A0;
2 const int ledPin = 9;
3
4 void setup() {
5     pinMode(potPin, INPUT);
6     pinMode(ledPin, OUTPUT);
7 }
8
9 void loop() {
10    int sensorValue = analogRead(potPin);
11    int brightness = sensorValue / 4;
12    analogWrite(ledPin, brightness);
13    delay(10);
14 }
```

5 Anhang

5.1 Beispiel Prompts

Einen guten Prompt zu schreiben, ist gar nicht so einfach. Zum Glück haben wir aber eine KI, welche uns bereits beim Erstellen des Prompts helfen kann. Die folgenden Prompts sind zum großen Teil mit der Hilfe der KI erstellt worden. Sie müssen danach aber vom Nutzer überprüft werden, ob sie das erfüllen, was man möchte.

5.1.1 The Flipped Interaction Pattern

Ein kleines Beispiel, wie man mit einer KI eine Unterrichtsstunde planen kann. Am Ende muss aber eine Einschätzung der Lehrkraft erfolgen und die Stunde ggf. auf die Klasse angepasst werden.

```
Ich möchte, dass du mir Fragen stellst, um eine
Unterrichtsstunde zum Thema ‚Erneuerbare Energien‘ zu
entwerfen. Stelle mir so lange Fragen, bis du ausreichend
Informationen hast, um eine vollständige Unterrichtseinheit
mit Zielen, Ablauf und Methoden vorzuschlagen. Frage immer
eine Frage nach der anderen.
```

Hier ist noch ein kleiner Prompt, mit welchem man kleine Projekte mit einem Mikrocontroller realisieren kann:

```
Ich möchte, dass du mir Fragen stellst, um ein einfaches
Projekt mit einem Mikrocontroller zu planen. Frage so lange
weiter, bis du alle nötigen Informationen hast, um mir
einen Bauplan und Beispielcode für das Projekt zu liefern.
Stelle maximal drei Fragen gleichzeitig.
```

5.1.2 Persona Pattern

Embedded-Systems-Ingenieur, welcher Ihnen Fragen zum Mikrocontroller beantworten kann:

Du bist ein erfahrener Embedded-Systems-Ingenieur mit viel Lehrerfahrung. Du bist darauf spezialisiert, technische Zusammenhänge verständlich und anwendungsnah zu erklären - vom Einsteiger bis zum fortgeschrittenen Technikstudenten. Du wirst nun so antworten, wie du es mit einem echten Studierenden tun würdest: Du stellst zuerst Rückfragen, um herauszufinden, was ich schon weiß, was mich besonders interessiert und wie tief wir einsteigen sollen. Erst nach diesen Rückfragen beginnst du mit deinen Erklärungen - angepasst an mein Vorwissen und meine Ziele. Ich möchte mehr über Mikrocontroller erfahren - was sie sind, wie sie aufgebaut sind und wofür man sie verwendet. Bitte beginne also mit klärenden Fragen, bevor du Inhalte vermittelst.

Mit diesem Prompt verhält sich die KI wie ein Linux-Terminal und man kann ihr Befehle geben, wo sie die Ausführung simuliert:

Ich möchte, dass du dich wie ein Linux-Terminal verhältst. Meine Inputs sind dann die, welche in ein Terminal eingegeben werden, und deine Outputs sind nur die, welche man auf dem Terminal sehen würde.

5.1.3 The Refine Question Pattern

Hier ist ein Beispiel, wie man die Frage bei Experimenten verwenden kann.

Wenn ich eine Frage zu einem Experiment stelle, schlage mir bitte eine präzisere Variante vor, die potenzielle Fehlerquellen, Messgrößen oder Kontextbedingungen berücksichtigt. Bitte erkläre kurz, warum deine Variante besser ist.

So wurde die Frage: „Wie kann ich die Lichtbrechung beim Übergang von Luft in Wasser zeigen?“ umformuliert zu:

„Wie kann ich in einem kontrollierten Experiment die Lichtbrechung beim Übergang von Luft in Wasser sichtbar machen, wobei der Einfallswinkel, der Brechungswinkel und die Brechzahl gemessen und mögliche Fehlerquellen wie Streuung oder Reflexion minimiert werden?“

Ein weiteres Beispiel für eine Geschichtsfrage:

Wenn ich dir eine Frage zu einem historischen Thema stelle, formuliere sie so um, dass sie möglichst genau ist, zum Beispiel durch das Einfügen von Zeitangaben, geografischen Bezügen oder Akteuren. Bitte frage, ob du die verfeinerte Version verwenden sollst.

Hier wird die Frage: „Was kann man aus der Französischen Revolution lernen?“, umformuliert zu:

„Welche Lehren lassen sich aus der Französischen Revolution (1789–1799) in Frankreich im Hinblick auf politische Umbrüche, soziale Gerechtigkeit und die Entwicklung moderner Demokratien ziehen?“

5.1.4 The Cognitive Verifier Pattern

Das ist ein Beispiel, wie z.B. dieses Pattern bei Physikfragen eingesetzt werden kann:

Bitte erkläre mir den folgenden Code nicht sofort. Stelle mir zunächst gezielte Fragen zum Code, um herauszufinden, wie gut ich ihn bereits verstehe. Nutze meine Antworten, um mein Verständnis einzuschätzen und erst anschließend den gesamten Code vollständig und nachvollziehbar zu erklären. Achte dabei darauf, eventuelle Denkfehler aufzudecken und mir Verbesserungsmöglichkeiten aufzuzeigen.