

Start mit ChatGPT + Arduino

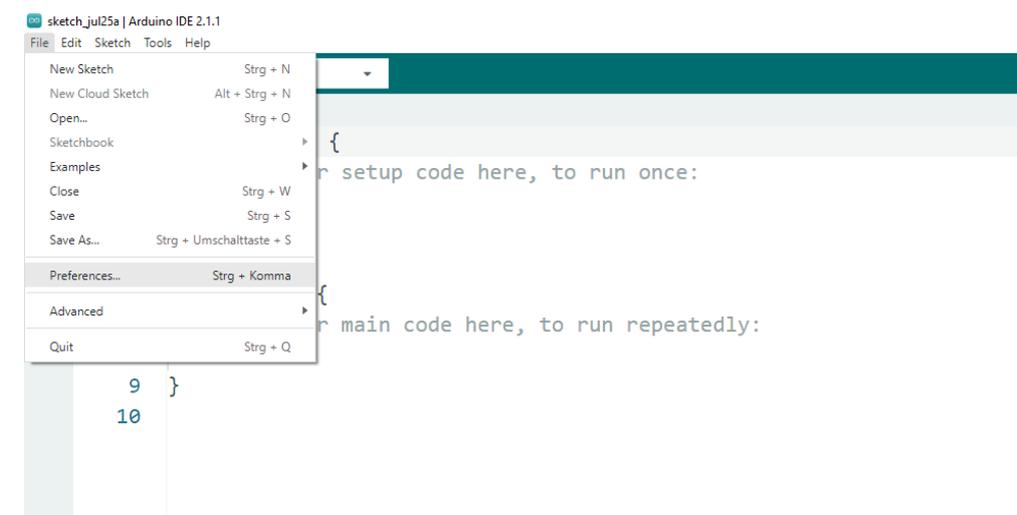


Das Board Arduino Uno und ChatGPT

Bevor mit Projekten gestartet wird sollte die nötige Hardware wie Board, LEDs und Widerstände beschafft werden. Erhältlich sind Originalboards oder kompatiblen Bords die Preise variieren zwischen 2.- bis 30.- CHF. Im Onlinehandel sind viele Anbieter zu finden. LEDs und Widerstände gehören zur Grundausrüstung einer LP21 konformen Werkstatt und sind über die bekannten Anbieter zu finden.

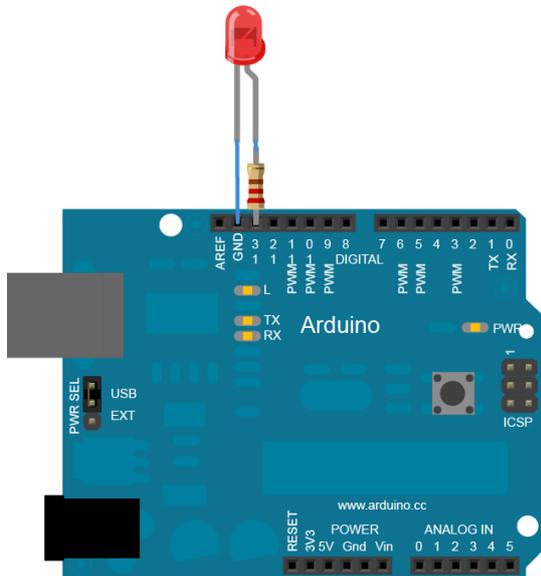
Um die Programmcodes auf das Arduino Bord zu laden, wird ein USB-Druckerkabel benötigt. Die Software „Arduino IDE 2.1.1“ kann aus dem [www](https://www.arduino.cc/en/software) downgeloadet werden (<https://www.arduino.cc/en/software>). Für die bekannten Betriebssysteme sind verschiedene Optionen erhältlich. Die Installation verläuft unter Windows problemlos, im Verlauf der Installation müssen lediglich USB-Treiber Erweiterungen zugelassen werden.

Nach erfolgreicher Installation kann in der Menüleiste unter File > Preferences die Sprache und die Darstellung angepasst werden.

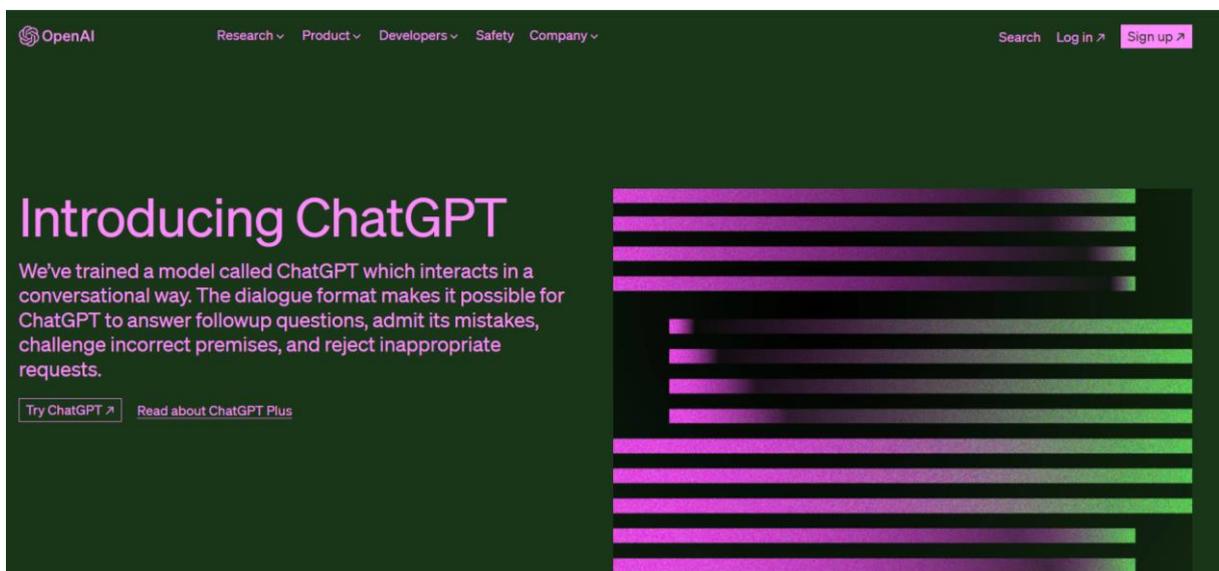


Das Arduino UNO Bord kann jetzt mit dem PC verbunden werden. Die Arduino IDE erkennt nun das Board und den Port, das wird im Textfeld in der grünen Leiste angezeigt. Durch Anwählen des angezeigten Bords inklusive des Ports ist nun alles betriebsbereit.

Für Einsteigende ist eine erste Schaltung mit einer blinkenden LED sinnvoll. An das längere Beinchen (Pluspol) der LED wird ein 220 Ω Widerstand angelötet. Das Minusbeinchen wird beim GND Pin und das Plusbeinchen im Pin 13 eingesteckt. In der Regel ist ein «Blinkcode» mit aktiviertem Ausgangspin 13 bereits aufs Bord geladen. Für Projekte mit wenigen LEDs ist die Stromversorgung des Bords über das USB Kabel ausreichend.



Um den Arduino-ChatGPT Spielplatz zu vervollständigen ist jetzt eine Anmeldung bei ChatGPT notwendig. Über die Internetseite von OpenAI (<https://openai.com/blog/chatgpt>) wird durch Klicken des Sign up Buttons die Anmeldung durchgeführt.



Nun kann eine erste Anfrage an ChatGPT gestellt werden: «Ich möchte eine LED mit dem Arduino Uno Board blinken lassen. Was brauche ich alles, wie wird die LED angeschlossen und wie lautet der Code?». Es ist erstaunlich wie präzise und gut verständlich ChatGPT Antworten liefert. Der gleich mitgelieferte Code kann nun kopiert und direkt in die Arduino IDE eingefügt werden. Dazu stellt ChatGPT über den Copy code Button oben rechts im Codefeld die Kopierfunktion gleich zur Verfügung.

Schritt 2: Code hochladen

Öffne die Arduino-Entwicklungsumgebung (IDE) und erstelle ein neues Sketch (Programm) mit dem folgenden Code:

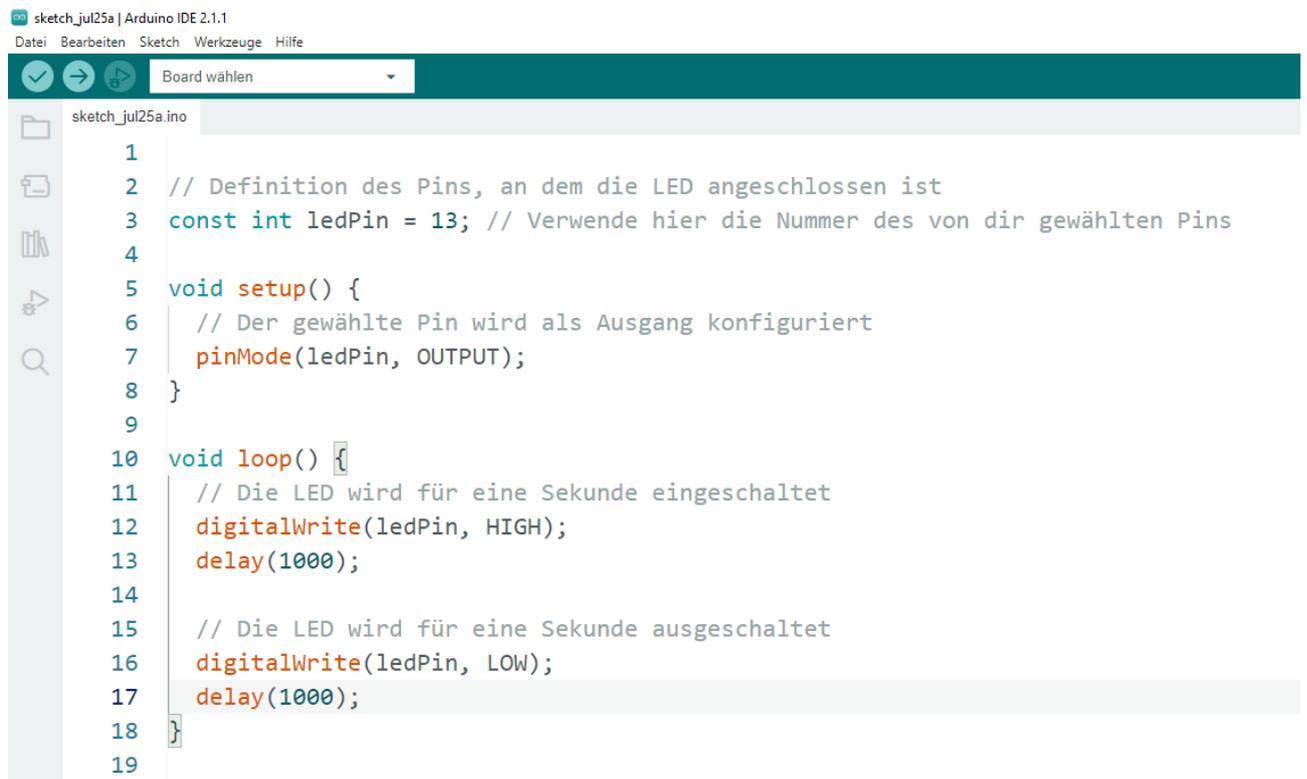
```
arduino Copy code

// Definition des Pins, an dem die LED angeschlossen ist
const int ledPin = 13; // Verwende hier die Nummer des von dir gewählten Pins

void setup() {
  // Der gewählte Pin wird als Ausgang konfiguriert
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // Die LED wird für eine Sekunde eingeschaltet
  digitalWrite(ledPin, HIGH);
  delay(1000);
```

In der Arduino IDE Umgebung kann der Code überprüft und auf das Board geladen werden. Der Häkchenbutton in der grünen Leiste überprüft den Code und zeigt mögliche Fehler an. Mit dem Pfeilbutton wird der Code aufs Board geladen. Beim Hochladen blinken die board-eigenen SMD LEDs, das bestätigt den Transfer.



```
sketch_jul25a | Arduino IDE 2.1.1
Datei Bearbeiten Sketch Werkzeuge Hilfe

sketch_jul25a.ino
1
2 // Definition des Pins, an dem die LED angeschlossen ist
3 const int ledPin = 13; // Verwende hier die Nummer des von dir gewählten Pins
4
5 void setup() {
6   // Der gewählte Pin wird als Ausgang konfiguriert
7   pinMode(ledPin, OUTPUT);
8 }
9
10 void loop() {
11   // Die LED wird für eine Sekunde eingeschaltet
12   digitalWrite(ledPin, HIGH);
13   delay(1000);
14
15   // Die LED wird für eine Sekunde ausgeschaltet
16   digitalWrite(ledPin, LOW);
17   delay(1000);
18 }
19
```

Wie sieht die grundlegende Codestruktur eigentlich aus? Zuerst müssen die benötigten Pins benannt und zugeordnet werden. Das passiert über «const int» wie im Beispiel, ledPin =13. Mit dem Gleichzeichen und der Zahl wird der Name der entsprechenden Pinnummer zugeordnet. Im «void setup» wird dem Pin die eigentliche Funktion zugeteilt, «pinMode» (ledPin, OUTPUT);» bedeutet, dass der ledPin ein Ausgang ist, und 5V, 0V oder andere Signale ausgeben kann. «void loop ()» ist nun die eigentliche Programmschleife und muss zwingend wie auch das «void setup» in geschwungenen Klammern stehen. Der Code Zeile 10 bis 18 ist schon fast selbsterklärend, «digitalWrite(ledPin, HIGH);» heisst schalte die LED ein, bzw. setze den ledPin auf +5V. «delay (1000);» bedeutet eine Verzögerung von einer Sekunde, der Wert Tausend steht für Millisekunden. «digitalWrite (ledPin, LOW) bedeutet folglich, schalte den «ledpin» auf 0V. Die Programmschleife (void loop) wird solange das Bord mit Strom versorgt wird ausgeführt. Anmerkung: Hinter einem Doppelslash können Erklärungen eingefügt werden, alles was dahinter steht wird nicht als Code gelesen. Am Schluss jeder Programmzeile muss zwingend ein Semikolon stehen, damit der jeweilige Befehl geschlossen ist.

Aufgabe 1

Wenn dein Arduino-ChatGPT Spielplatz schon mal funktioniert, darfst du dir auf die Schulter klopfen!

Um das Arduino Coding etwas besser zu verstehen greifst du nun direkt in der Arduino IDE ein und veränderst den Code.

- 1) Verändere den Delaywert in beiden Zeilen. Überprüfe den Code und lade ihn aufs Board. Was stellst du fest. Beschreibe die Ergebnisse im Kasten unten. Schreibe ebenfalls die veränderte Programmzeile hin.

- 2) Du kannst Codezeilen auch kopieren. Versuche die LED zweimal kurz und zweimal lang blinken zu lassen. Wenn das funktioniert kannst du einen SOS Code Programieren.



Halte dein Vorgehen und deine Problemlösungen unten fest.