

ChatGPT

Was ist ChatGPT?

ChatGPT ist ein KI-gestützter Chatbot, der auf der GPT-3.5-Architektur von OpenAI basiert. GPT steht für "Generative Pre-trained Transformer". Es handelt sich um ein fortschrittliches neuronales Netzwerkmodell, das auf umfangreichen Textdaten trainiert wurde und in der Lage ist, menschenähnlichen Text zu generieren und auf Anfragen zu antworten.

ChatGPT wurde speziell entwickelt, um auf Chat-Konversationen zu reagieren und sinnvolle und kontextbezogene Antworten zu liefern. Es kann Fragen beantworten, Informationen bereitstellen, Ratschläge geben, Anleitungen geben und vieles mehr.

Das Modell lernt aus großen Mengen an Textdaten und erfasst dabei komplexe Zusammenhänge und Muster in der Sprache. Es kann auf eine Vielzahl von Themen und Fragen eingehen, aber es ist wichtig zu beachten, dass ChatGPT auf vorheriges Wissen und den Text, mit dem es trainiert wurde, beschränkt ist. Es kann nicht auf aktuelle Ereignisse nach dem Trainingszeitpunkt antworten und kann auch keine persönlichen Erfahrungen haben.

ChatGPT hat das Potenzial, in verschiedenen Anwendungen eingesetzt zu werden, wie zum Beispiel beim Kundenservice, der Unterstützung bei der Problemlösung, bei der Wissensvermittlung, bei der Sprachinteraktion und mehr. Es ist jedoch wichtig zu beachten, dass ChatGPT ein computergenerierter Text ist und seine Antworten sorgfältig überprüft werden sollten, insbesondere wenn es um sicherheitsrelevante oder vertrauliche Informationen geht.

Wie funktioniert ChatGPT?

ChatGPT basiert auf dem Konzept des maschinellen Lernens und verwendet eine spezielle Form des neuronalen Netzwerks, die als Transformer bezeichnet wird. Hier ist eine vereinfachte Erklärung, wie ChatGPT funktioniert:

1. **Training:** ChatGPT wird mit großen Mengen an Textdaten trainiert. Diese Daten können aus Büchern, Artikeln, Websites, Chatprotokollen und vielen anderen Quellen stammen. Das Modell lernt, die statistischen Muster und Strukturen in den Daten zu erfassen.
2. **Architektur:** ChatGPT basiert auf der GPT-3.5-Architektur (Generative Pre-trained Transformer 3.5). Transformer sind neuronale Netzwerkmodelle, die speziell für die Verarbeitung von Text entwickelt wurden. Sie bestehen aus mehreren Schichten von Aufmerksamkeitsmechanismen und Feedforward-Netzwerken.

3. Eingabe und Codierung: Wenn eine Eingabe an ChatGPT gegeben wird, wird der Text in eine numerische Darstellung umgewandelt, die das Modell verarbeiten kann. Das Modell codiert die Eingabe und berechnet eine interne Darstellung, die den Kontext und die Bedeutung des Textes erfasst.
4. Verarbeitung und Generierung: Das Modell verwendet die interne Darstellung, um eine Vorhersage für die nächste Antwort zu treffen. Es generiert Text basierend auf den gelernten Mustern und dem Kontext der Eingabe. Dabei werden sowohl die unmittelbare Eingabe als auch frühere Interaktionen berücksichtigt, um eine kontextbezogene Antwort zu erzeugen.
5. Ausgabe: Die generierte Antwort wird in Textform zurückgegeben und dem Benutzer präsentiert.
6. Feinabstimmung: Nach dem Training kann das Modell weiter feinabgestimmt werden, um es an spezifische Anwendungsfälle oder Domänen anzupassen. Dies kann bedeuten, dass das Modell auf spezifische Datensätze oder Anforderungen hin optimiert wird, um bessere und spezifischere Antworten zu generieren.

Es ist wichtig zu beachten, dass ChatGPT ein statistisches Modell ist und seine Antworten auf Wahrscheinlichkeiten basieren. Die Qualität der Antworten hängt stark von der Qualität der Trainingsdaten ab. Obwohl ChatGPT bemerkenswerte Fähigkeiten beim Verständnis und Generieren von Text hat, kann es gelegentlich auch ungenaue oder unplausible Antworten liefern. Daher ist es wichtig, die Antworten des Modells kritisch zu prüfen und bei Bedarf weitere Überprüfungen oder Recherchen durchzuführen.

Quelle: Der Text "Was ist ChatGPT?" und "Wie funktioniert ChatGPT?" wurde von ChatGPT erstellt.

Umsetzung eines konkreten Projektes mit ChatGPT

Machen wir einen Versuch und lassen ChatGPT ein Programm für einen Arduino UNO schreiben. Das Programm soll mit einem Poti einen Servo-Motor ansteuern.

Zunächst müssen wir auf der Seite <https://chat.openai.com> einen Account erstellen. Dazu reicht eine Email-Adresse. Die Nutzung von ChatGPT ist kostenlos.



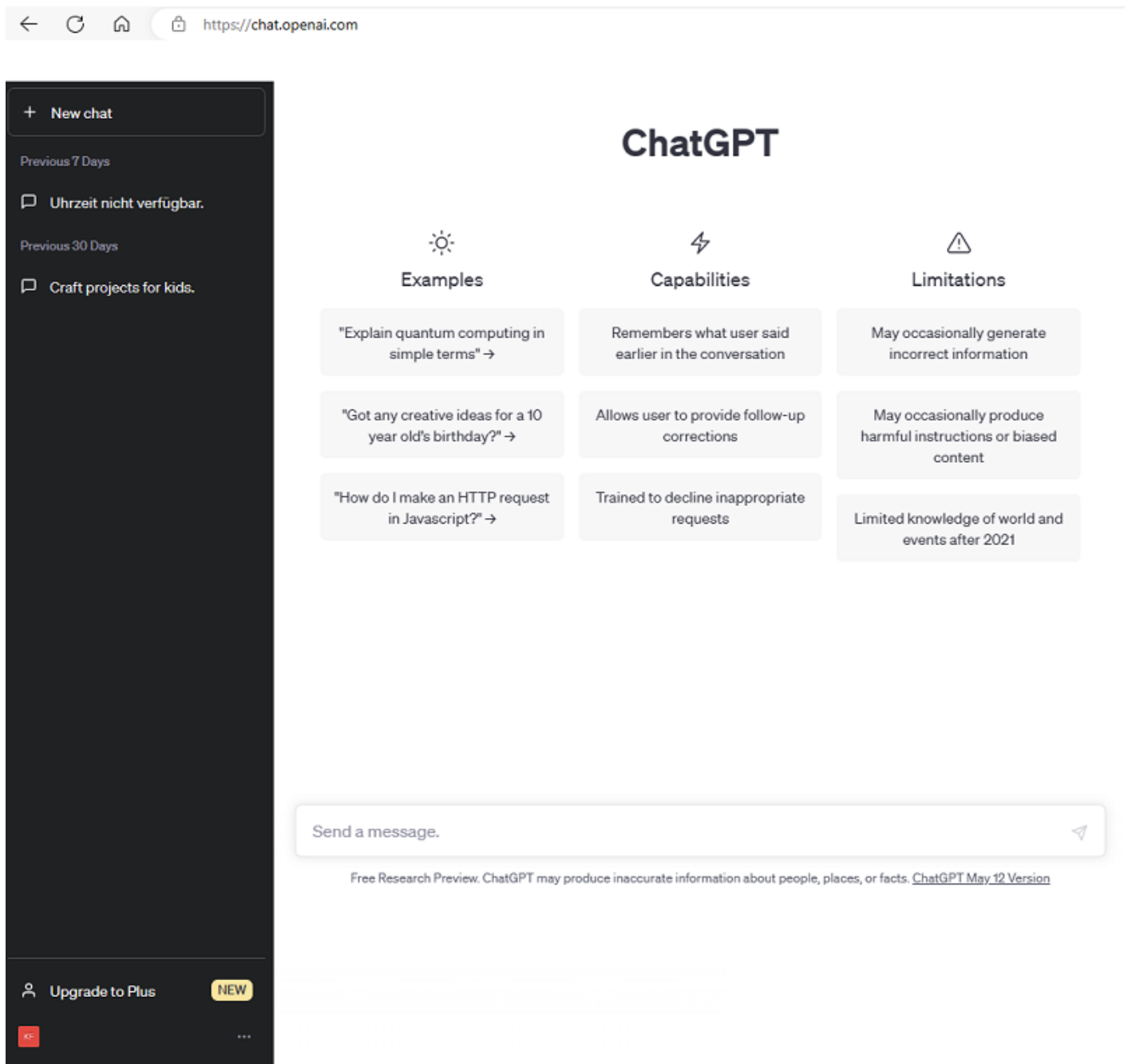
Welcome to ChatGPT

Log in with your OpenAI account to continue

Log in

Sign up

Nach erfolgreichem Login öffnet sich das Chat-Fenster.



Nun stellen wir ChatGPT die Aufgabe ein Programm zu schreiben.
Die Aufgabe: Erstelle ein Programm für einen Arduino UNO. Das Programm soll einen Servo-Motor über ein Poti steuern.

New chat

Today

Poti-Servo-Steuerung

Calliope Mini Motorsteuerung

Previous 7 Days

Uhrzeit nicht verfügbar.

Previous 30 Days

Craft projects for kids.

Upgrade to Plus

NEW

KF

...

Erstelle ein Programm für einen Arduino UNO. Das Programm soll einen Servo-Motor über ein Poti steuern.

Natürlich! Hier ist ein Beispielprogramm, das einen Servomotor über ein Potentiometer (Poti) steuert. Das Programm wurde für den Arduino UNO entwickelt und verwendet die Servo-Bibliothek, die im Arduino-Entwicklungsumfeld verfügbar ist. Stelle sicher, dass du die Servo-Bibliothek installiert hast, bevor du das Programm ausführst.

cpp

Copy code

```
#include <Servo.h>

// Definiere die Pins für den Servo-Motor und das Pot
const int servoPin = 9;
const int potPin = A0;

Servo servo;

void setup() {
  // Initialisiere den Servo und das Potentiometer
  servo.attach(servoPin);
}

void loop() {
  // Lese den Wert des Potentiometers
  int potValue = analogRead(potPin);

  // Skaliere den Potentiometerwert auf den Servo-Winkel
  int angle = map(potValue, 0, 1023, 0, 180);

  // Bewege den Servo zum berechneten Winkel
  servo.write(angle);
}
```

Regenerate response

Send a message.

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. [ChatGPT May 12 Version](#)

Nach kurzer Wartezeit erscheint das Ergebnis.

```

cpp
Copy code

#include <Servo.h>

// Verbinde das Potentiometer mit dem Pin A0
#define POTENTIOMETER_PIN A0

// Erstelle ein Servo-Objekt
Servo servoMotor;

// Definiere die minimalen und maximalen Werte für den Servowinkel
const int SERVO_MIN_ANGLE = 0;
const int SERVO_MAX_ANGLE = 180;

// Variablen zur Speicherung des aktuellen Werts des Potentiometers
int potentiometerValue = 0;
int servoAngle = 0;

void setup() {
    // Initialisiere das Servo-Objekt
    servoMotor.attach(9); // Verbinde den Servo mit Pin 9
}

void loop() {
    // Lies den Wert des Potentiometers
    potentiometerValue = analogRead(POTENTIOMETER_PIN);

    // Konvertiere den Potentiometerwert in einen Servowinkel
    servoAngle = map(potentiometerValue, 0, 1023, SERVO_MIN_ANGLE, SERVO_MAX_ANGLE);

    // Stelle den Servowinkel ein
    servoMotor.write(servoAngle);

    // Eine kurze Verzögerung, um ein Flackern zu verhindern
    delay(15);
}

```

Regenerate response

Send a message.

ChatGPT Servo Programm:

```

// C++ code
//
#include <Servo.h>

```

```

// Verbinde das Potentiometer mit dem Pin A0
#define POTENTIOMETER_PIN A0

// Erstelle ein Servo-Objekt
Servo servoMotor;

// Definiere die minimalen und maximalen Werte für den Servo
const int SERVO_MIN_ANGLE = 0;
const int SERVO_MAX_ANGLE = 180;

// Variablen zur Speicherung des aktuellen Werts des Potentiometers und des Servowinkels
int potentiometerValue = 0;
int servoAngle = 0;

void setup() {
  // Initialisiere das Servo-Objekt
  servoMotor.attach(9); // Verbinde den Servo mit Pin 9
}

void loop() {
  // Lies den Wert des Potentiometers
  potentiometerValue = analogRead(POTENTIOMETER_PIN);

  // Konvertiere den Potentiometerwert in einen Servowinkel
  servoAngle = map(potentiometerValue, 0, 1023, SERVO_MIN_ANGLE, SERVO_MAX_ANGLE);

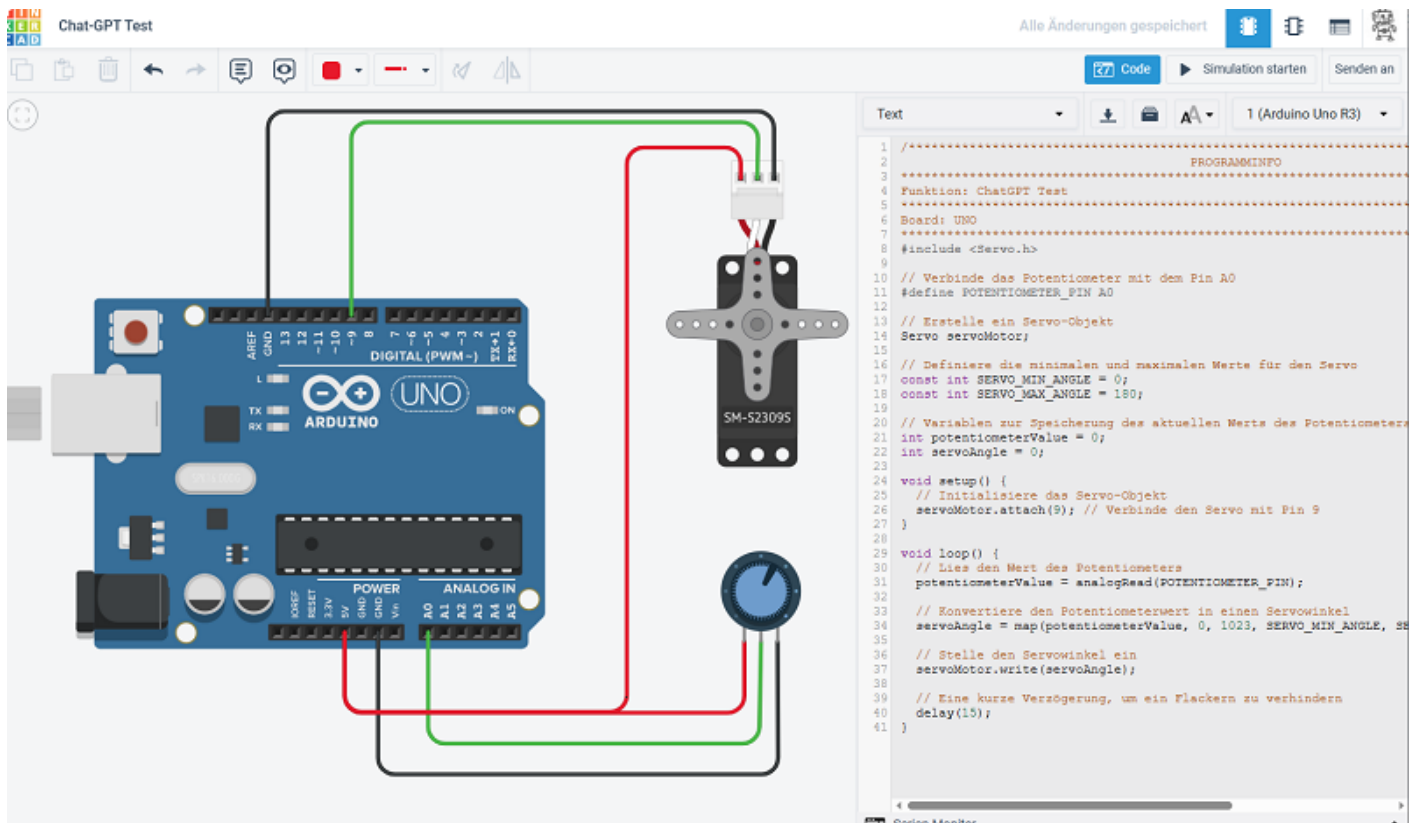
  // Stelle den Servowinkel ein
  servoMotor.write(servoAngle);

  // Eine kurze Verzögerung, um ein Flackern zu verhindern
  delay(15);
}

```

01 Den erzeugten Programmcode können wir auch in Tinkercad testen und

simulieren.



02 Teste jetzt ChatGPT. ChatGPT soll dir das Programm um ein LCD-Display erweitern und den Winkel des Servo-Motors anzeigen.

Simuliere es dann in Tinkercad.

ChatGPT Servo Programm mit LCD-Display:

```
// C++ code
//

#include <Servo.h>
#include <Adafruit_LiquidCrystal.h>

const int servoPin = 9;
const int potPin = A0;
```



```
Servo servo;

Adafruit_LiquidCrystal lcd_1(0);

void setup()
{
  lcd_1.begin(16, 2);
  servo.attach(servoPin);
  lcd_1.print("Winkel: ");
}

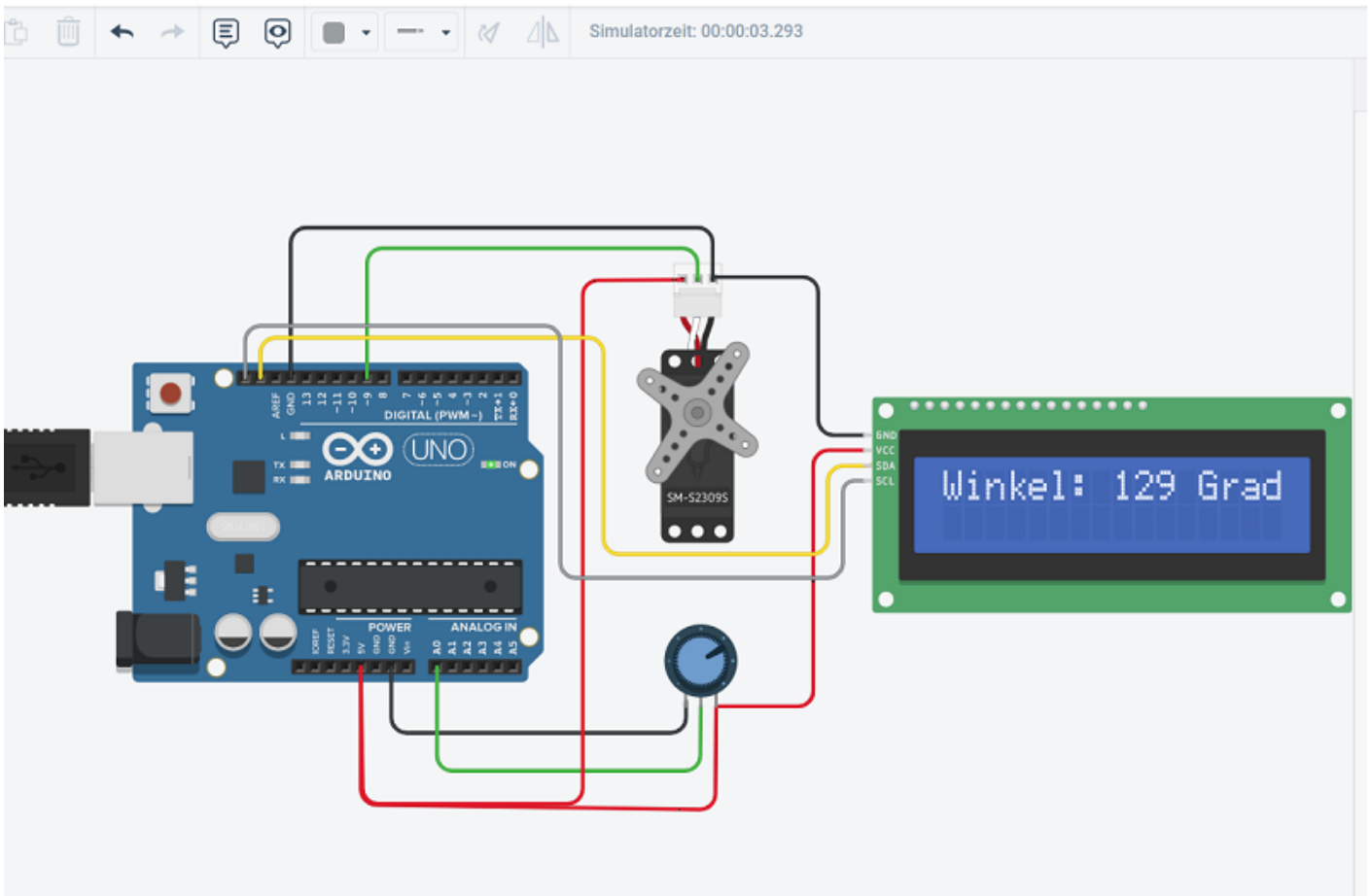
void loop(){
  // Lese den Wert des Potentiometers
  int potValue = analogRead(potPin);

  // Skaliere den Potentiometerwert auf den Servo-Winkelbereich (0-180 Grad)
  int angle = map(potValue, 0, 1023, 0, 180);

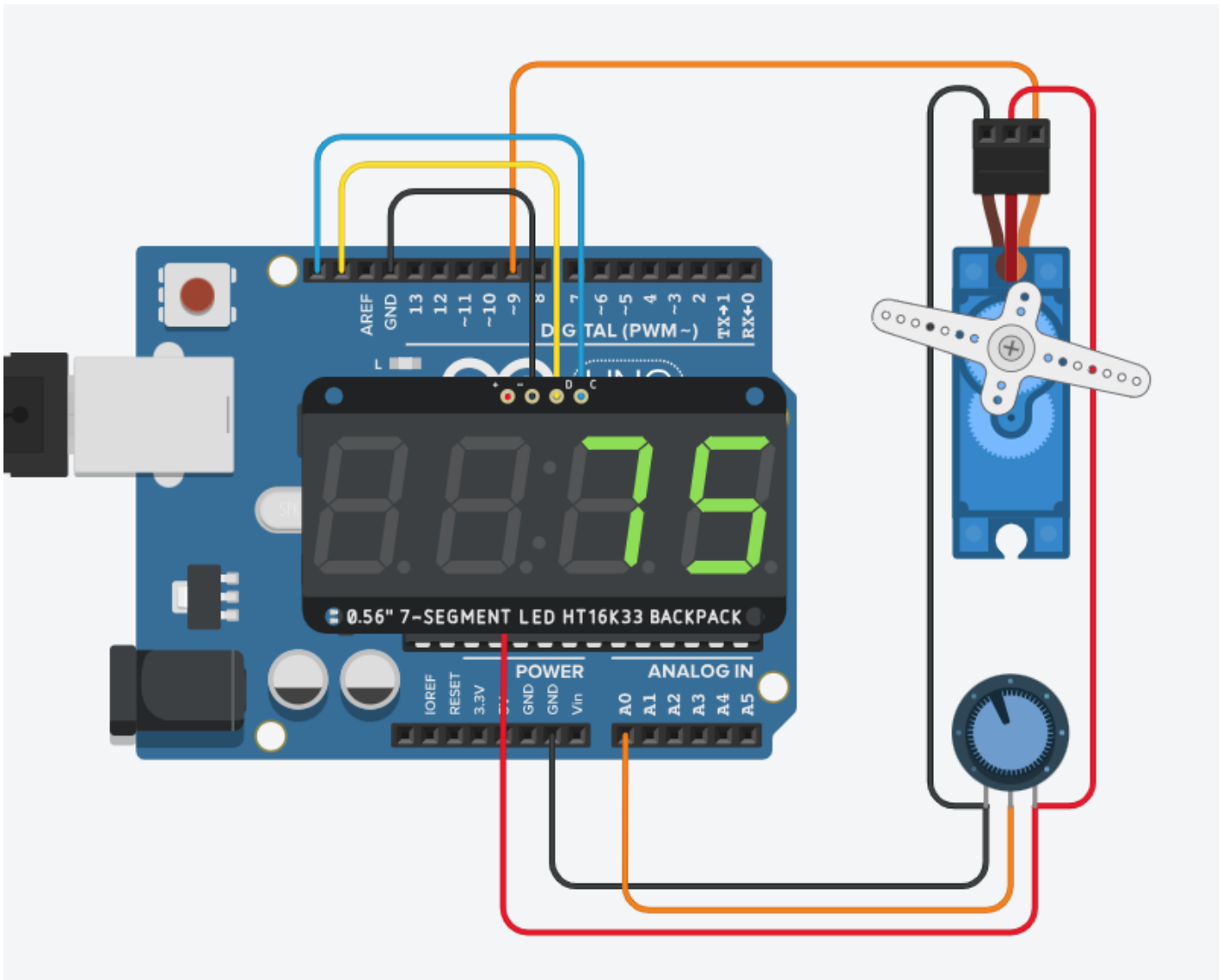
  // Bewege den Servo zum berechneten Winkel
  servo.write(angle);

  // Aktualisiere den Winkel auf dem LCD-Display
  lcd_1.setCursor(8, 0);
  lcd_1.print(angle);
  lcd_1.print(" Grad");

  lcd_1.print("  "); // Leere Stellen löschen
}
```



Alternativ zum LCD-Display kannst du auch eine 7-Segment-Anzeige verwenden.



ChatGPT Servo Programm mit 7-Segment-Anzeige:

```
// C++ code
//
#include "Adafruit_LEDBackpack.h"

#include <Servo.h>

int Potiwert = 0;

int Winkel = 0;

Adafruit_7segment led_display1 = Adafruit_7segment();

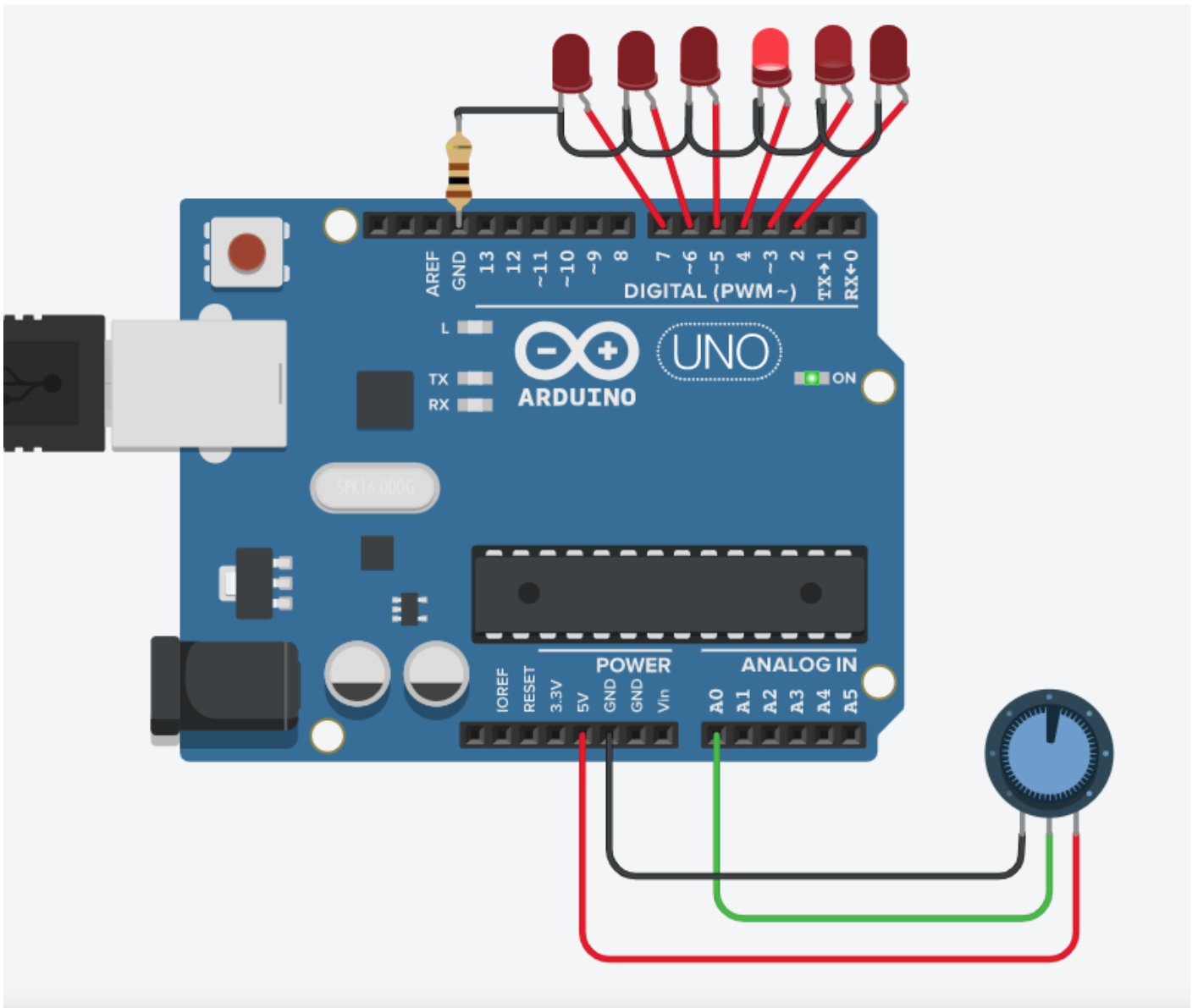
Servo servo_9;
```

```
void setup()
{
  led_display1.begin(112);
  servo_9.attach(9, 500, 2500);
  pinMode(A0, INPUT);

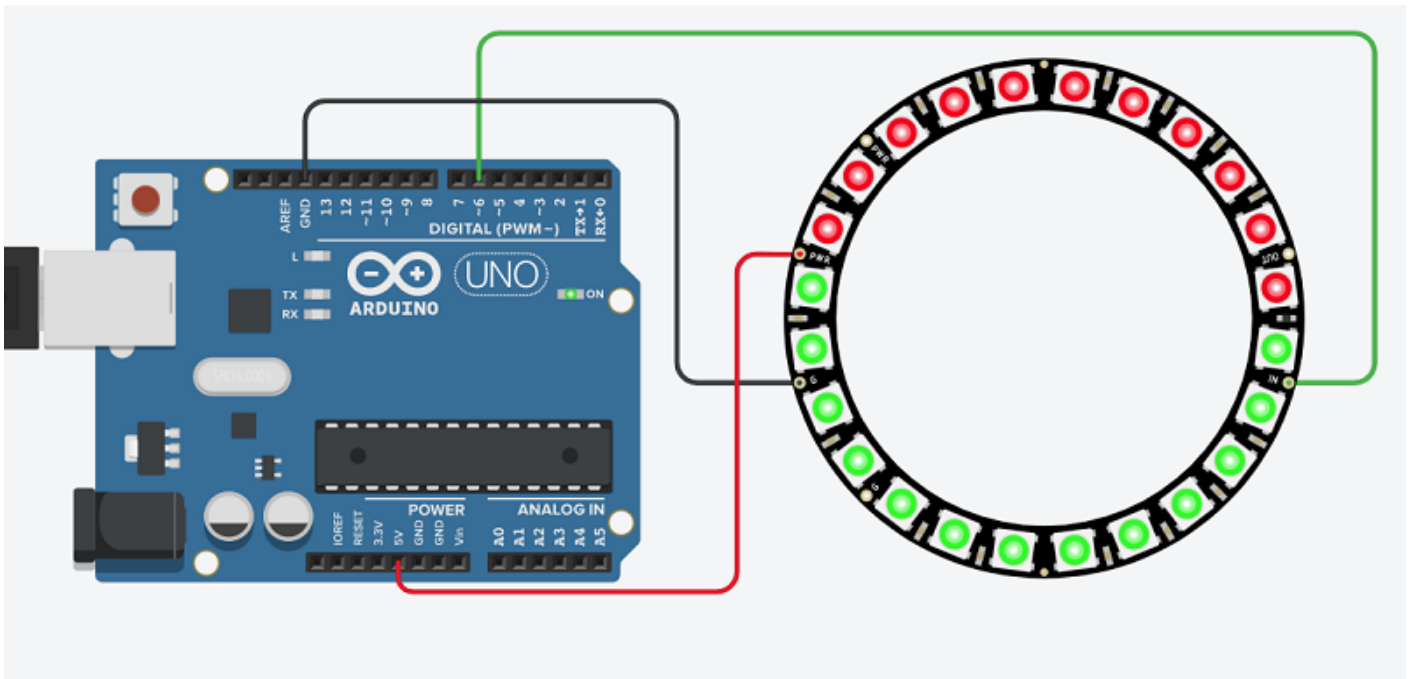
  servo_9.write(0);
}

void loop()
{
  Potiwert = analogRead(A0);
  Winkel = map(Potiwert, 0, 1023, 0, 180);
  led_display1.println(Winkel);
  led_display1.writeDisplay();
  servo_9.write(Winkel);
  delay(100); // Wait for 100 millisecond(s)
}
```

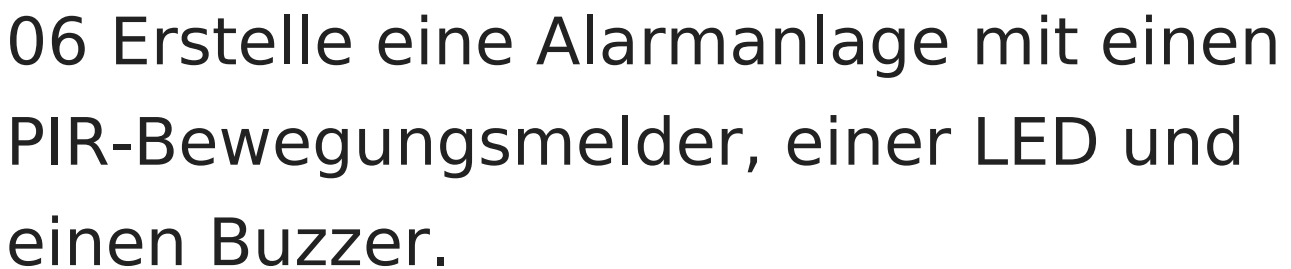
03 Lasse ChatGPT ein Lauflicht-Programm für 6 LEDs schreiben. Die Lauflicht-Geschwindigkeit soll mit einem Poti einstellbar sein.

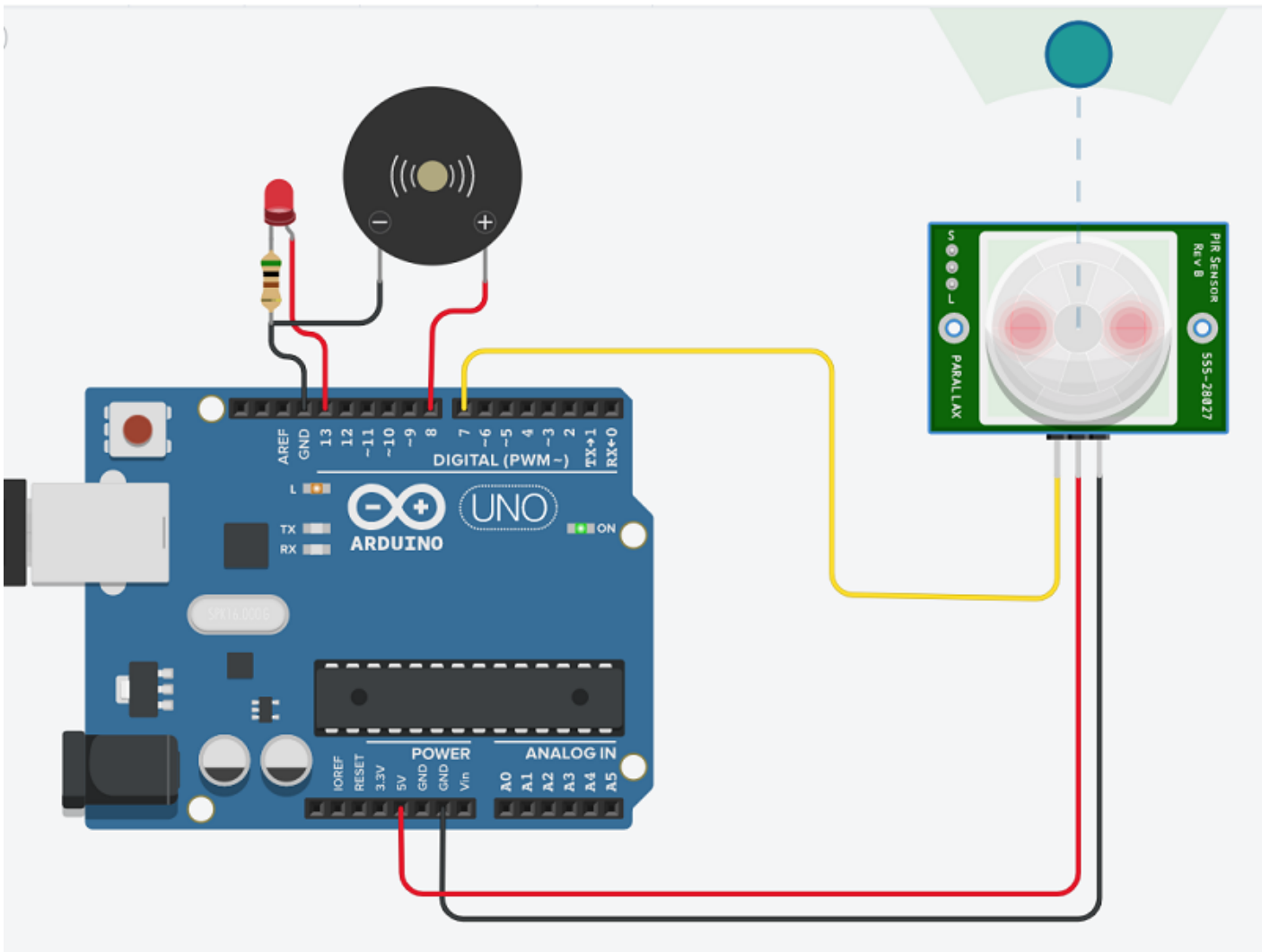


04 Lasse ChatGPT ein Programm für einen NEO-Pixel-Ring 24 mit verschiedenen Farbkombinationen schreiben.

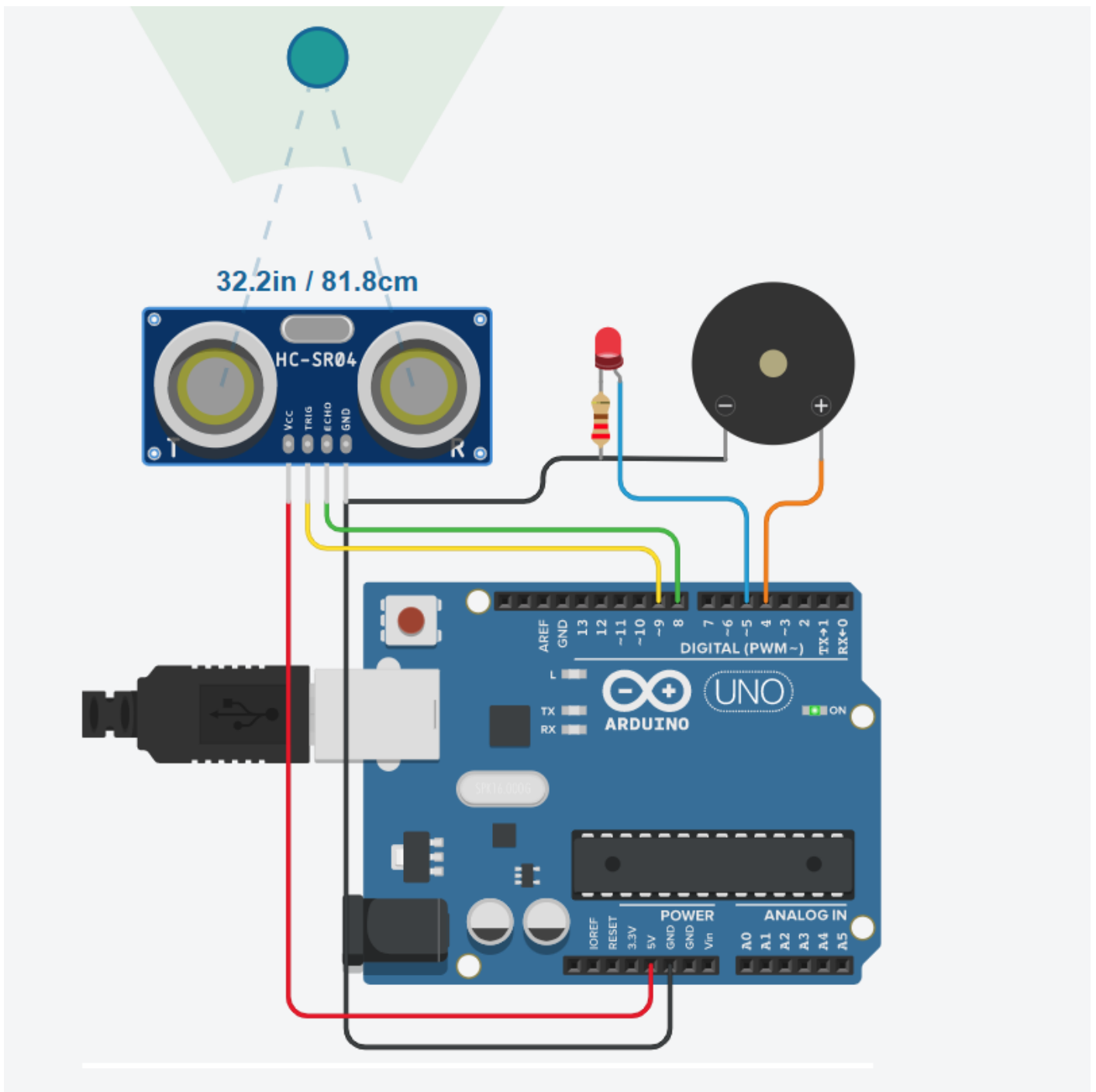


05 Erweitere jetzt das erste Programm mit dem Poti und den Servo-Motor zur Greifarmsteuerung mit 5 Potis und 5 Servos.

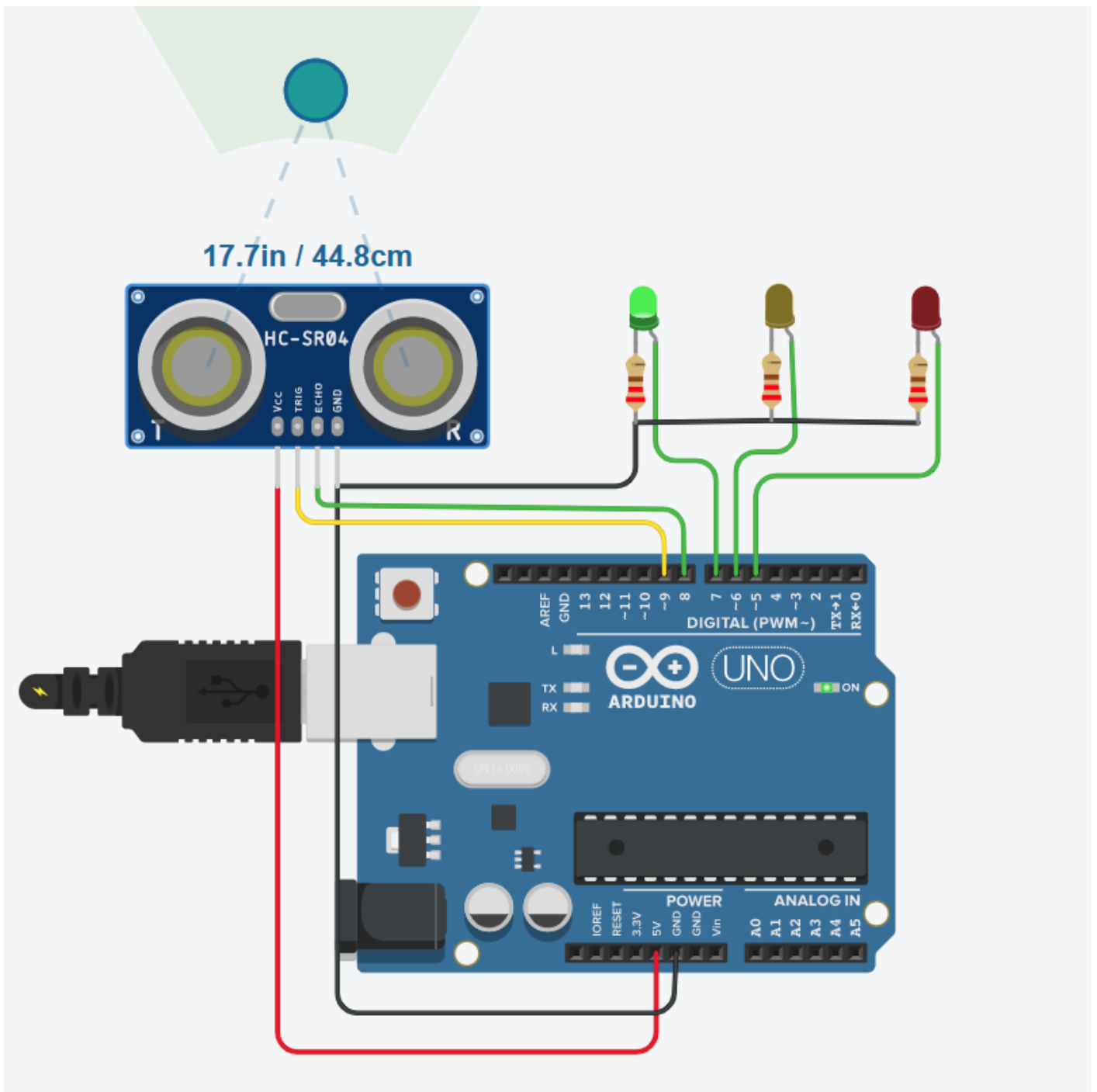




07 Erstelle eine Einparkhilfe mit einem Ultraschallsensor, einer LED und einem Buzzer.

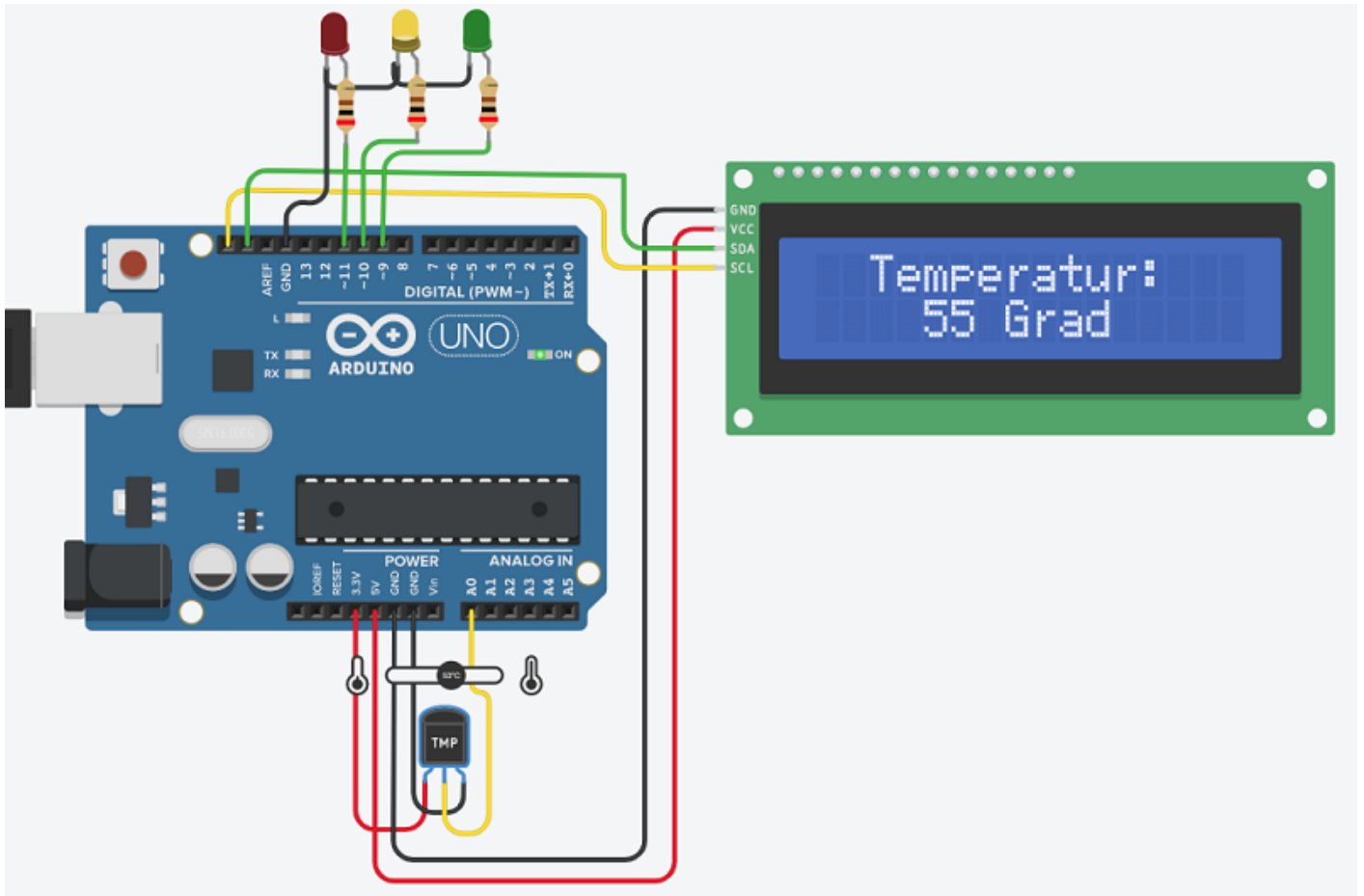


08 Erstelle eine Ampelschaltung. Die Grünphase soll durch einen Ultraschallsensor ausgelöst werden.

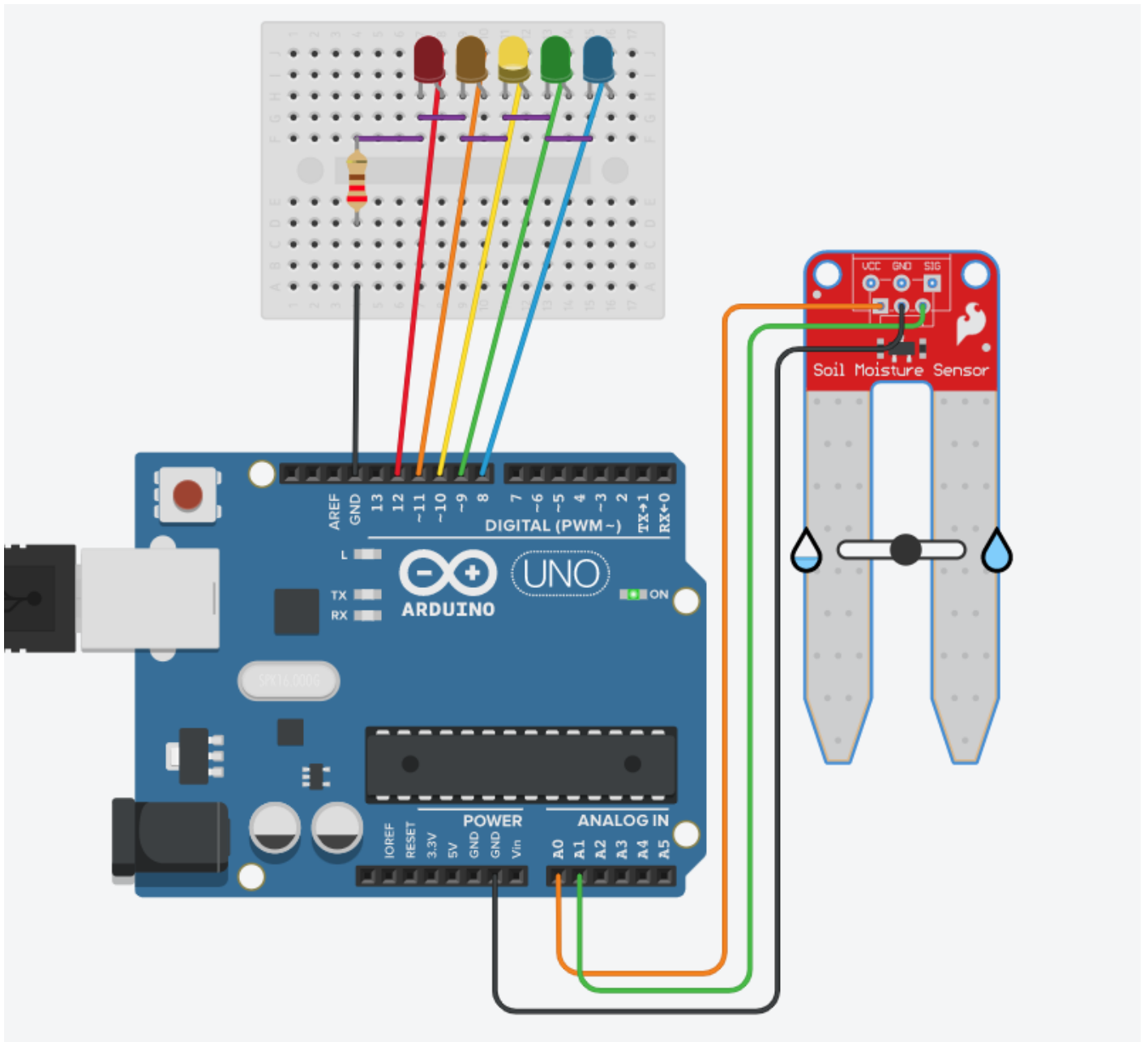


09 Erstelle eine Temperaturanzeige. Die grüne LED soll im Temperaturbereich 25°-39° leuchten, die gelbe LED im Temperaturbereich 40°-59° und die rote LED soll bei der Temperatur > 60°

leuchten.



10 Erstelle eine Feuchteanzeige mit 5 LEDs. Die LEDs sollen die Feuchte in der Abstufung 0-199 (rot), >200 (orange), >400 (gelb), >600 (grün) und >800 (blau) anzeigen.



Teste noch weitere Beispiele mit Hilfe von ChatGPT!

Version #3

Erstellt: 30 März 2025 21:43:08 von Joel Hatsch

Zuletzt aktualisiert: 30 März 2025 22:00:25 von Joel Hatsch