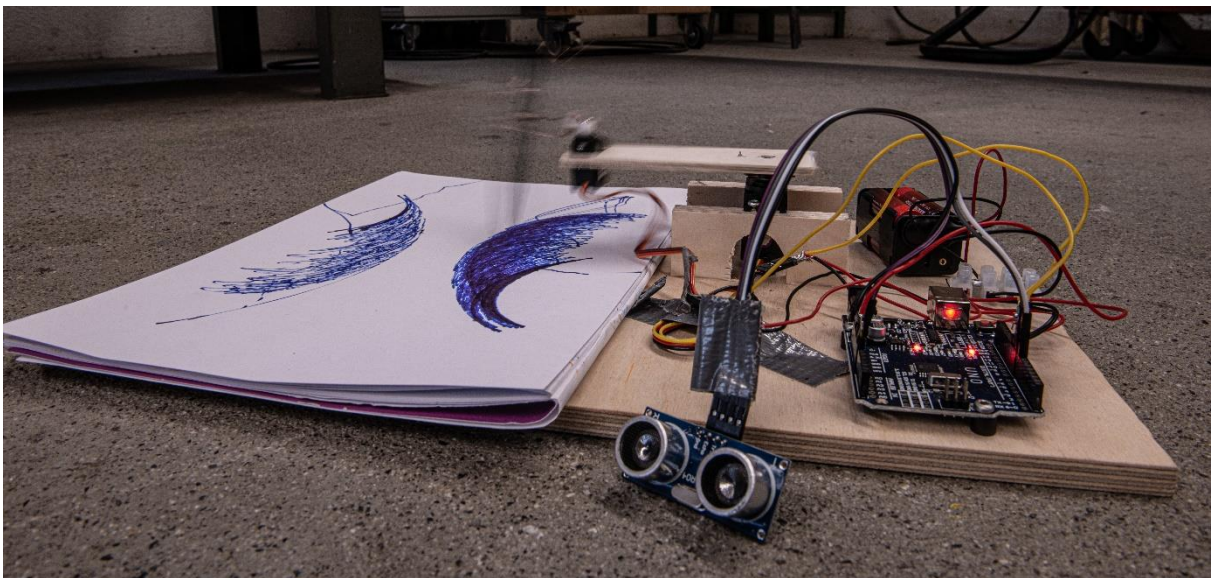


Zeichenmaschine



Jean_Tinquely, Méta-Matic No. 10 <https://www.myswitzerland.com/de-ch/erlebnisse/staedte-kultur/kunst-kultur/kunst/metamatic-no-10/>



Zeichenmaschine von SuS der SEKI

Zeichenmaschine

Aufgabe

Erfinde und baue eine Zeichenmaschine. Setze dabei Sensoren und Aktuatoren in ein optimales Zusammenspiel. Gefragt sind Bewegungsmöglichkeiten von Modellbauservos und Hebeln, die auf Sensorwerte des Ultraschallsensors reagieren.

Material

Arduino Uno Board, Ultraschallsensorsensor HC-SR045, Modellbauservos analog, Schaltdraht oder Jumper-Kabel, nützlich sind auch Leiterplattensteckverbinder, Holzplatten, Drähte, Kleber, Heissleim usw.



Verfahren

Holz sägen, Draht trennen und biegen, Schrauben, Weichlöten, Abisolieren, Kleben

Vorgehen

Ideensammlung: Halte sechs Ideen in Skizzen fest.

Mit Karton und Rundkopfklemmen machst du ein Modell deiner Hebel/Zeichenarme. Im Original sind die Drehpunkte die Servos. Experimentiere mit den Bewegungsmöglichkeiten deiner Hebelkombination

Bespreche die Ideen mit deiner LP. Kläre dabei die Grösse und Umsetzbarkeit.

Studiere die Arbeitsschritte, organisiere dein Material und starte mit dem Bau des Objekts.

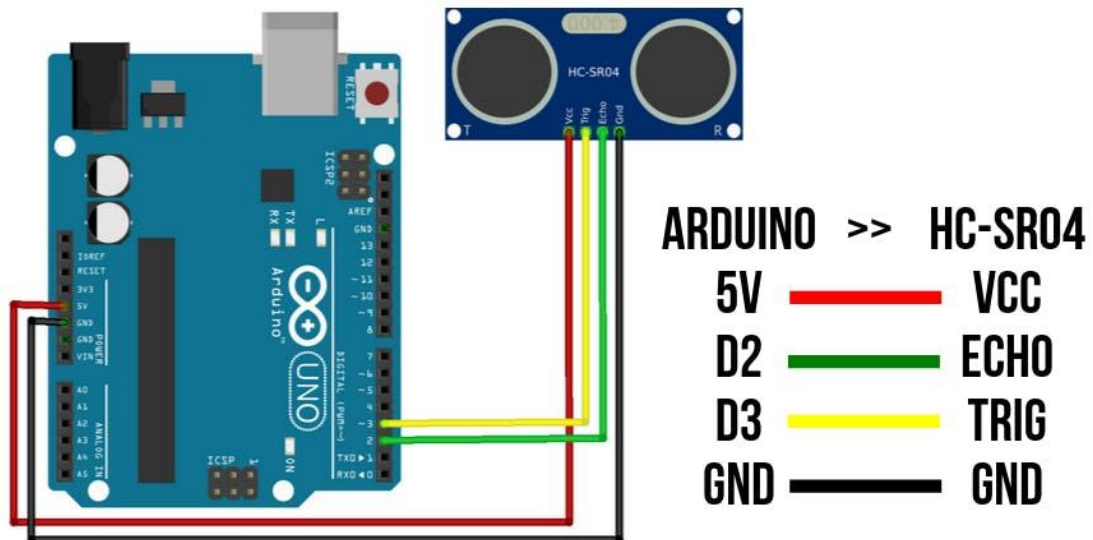
Starte nun deine ChatGPT/Arduino Umgebung. Nun gibst du den ersten Prompt in ChatGPT ein.

Wichtig ist es die Anschlüsse am Board zu erfragen. Danach beschreibst/erfragst du erste Bewegungen deiner Servos in Abhängigkeit des Sensorwertes des Distanzsensors (Ultraschallsensors).

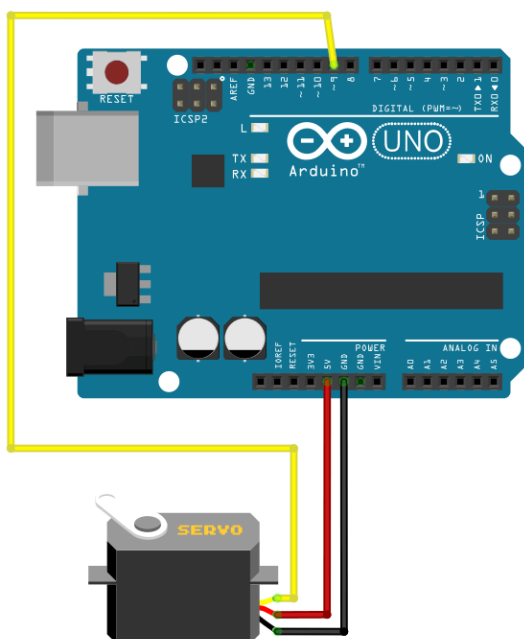
Servos können eine Teilkreisbewegung von 0-180 Grad ausführen.

Anschluss HC-SR04

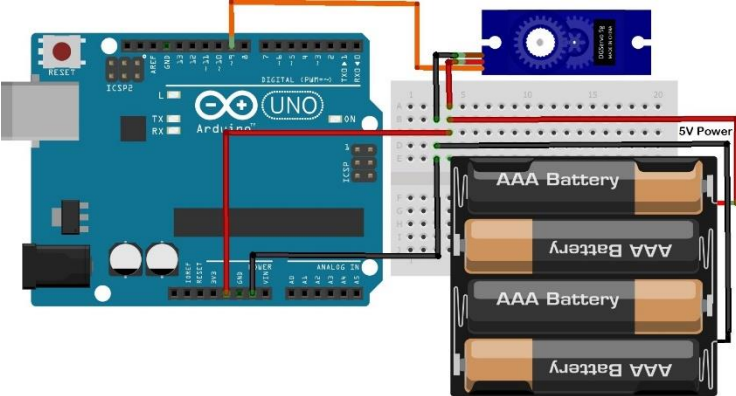
Der Distanzsensor (Ultraschallsensor HC-SR04) muss mit Strom versorgt werden. Dazu verbindest du den Pluspin des Sensors mit dem 5V Anschluss des Boards, den Minuspin verbindest du mit einem GND Anschluss. Weiter wird der Trig-Kontakt auf einen digitalen Ein/Ausgang des Uno Boards gesteckt, ebenso der Echo-Kontakt. Erfrage dazu ChatGPT oder bestimme die Arduino Pins selbst und erwähne sie in deinem Prompt.



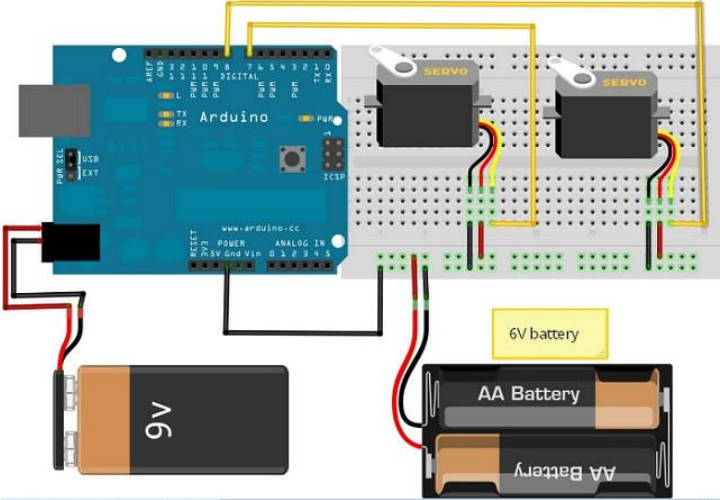
Anschluss Servo



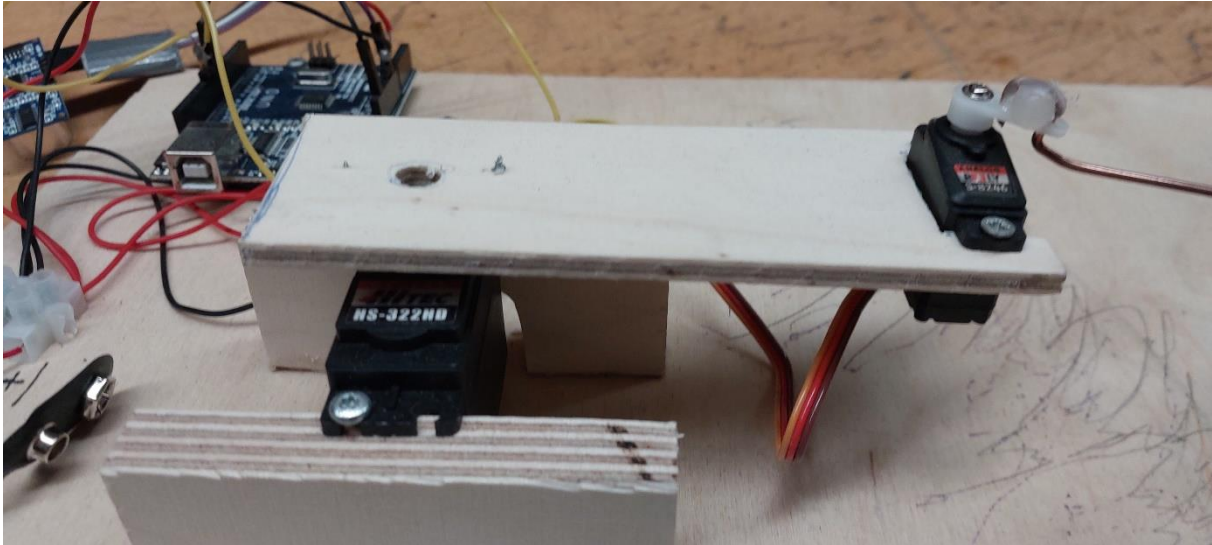
Ein Servo kann möglicherweise direkt an das Board angeschlossen werden. Servos benötigen in der Regel viel Strom und lassen dadurch die Spannung am Board abfallen. Dies führt zu Fehlfunktionen. Anschluss: Plus/ Minus an 5V und GND, Signalleitung z. B. an Ausgang 9.



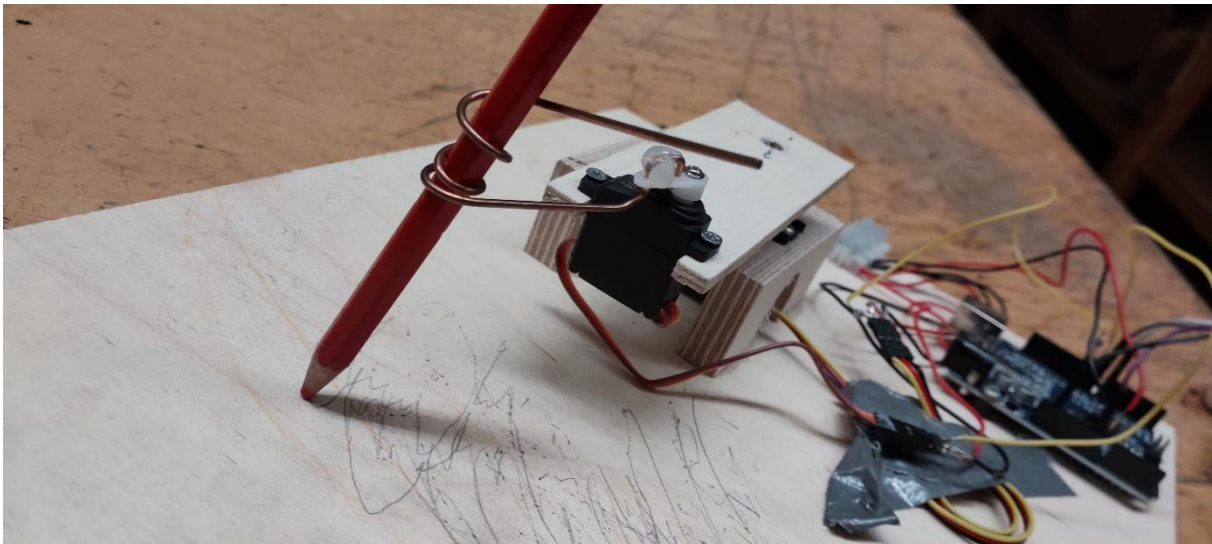
Besser ist es, den Servo direkt mit einer Batterie zu versorgen, weiter kann eine zweite Batterie direkt an das Board angeschlossen werden. In diesem Falle müssen die Minuspole der Batterien verbunden werden.



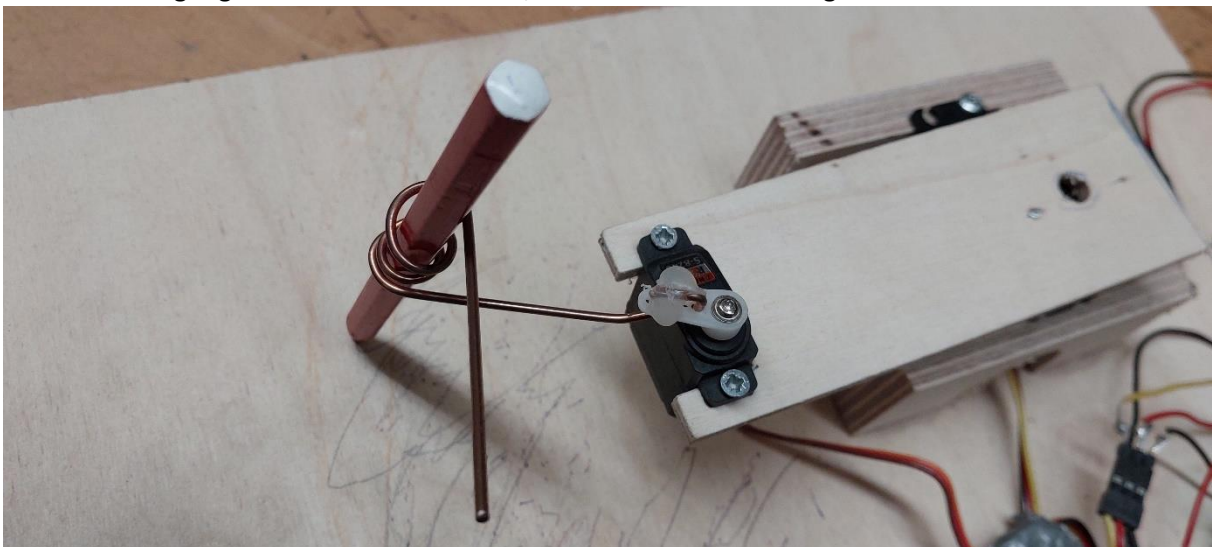
Zeichenmaschine Tipps



Möglichkeit der Befestigungen des primären und sekundären Servos



Für die Befestigung des Stiftes ist es sinnvoll, eine wechselbare Lösung zu finden.



Die Hebelkombination entspricht einer Basic-Variante. Erfinde neue Hebelkombinationen die ein grösseres Zeichenspektrum ermöglichen. Experimentiere auch mit weiteren Materialien

Folgend findest du die ganze Konversation mit ChatGPT und den immer wieder angepassten Arduino Codes

ich habe ein arduino uno board, zwei analog servos und einen hc sr04 ultraschall abstands sensor. wie muss ich die Bauteile anschliessen? In abhängigkeit der distanz des Sensors, im Bereich von 10-40cm, sollen sich die Servos unterschiedlich bewegen. Kannst du einen code schreiben?

ChatGPT

Um den HC-SR04 Ultraschallsensor, zwei analoge Servos und ein Arduino Uno Board miteinander zu verbinden und einen Code zu schreiben, der die Servos in Abhängigkeit von der gemessenen Entfernung des Sensors steuert, folge diesen Schritten:

Schaltungsaufbau:

1. Schließe den HC-SR04 Ultraschallsensor wie folgt an:
 - VCC (Versorgungsspannung) des Sensors an 5V des Arduino.
 - GND (Ground) des Sensors an GND des Arduino.
 - Trig-Pin des Sensors an Pin 7 des Arduino.
 - Echo-Pin des Sensors an Pin 8 des Arduino.
2. Schließe die beiden analogen Servos an den Arduino an:
 - Verbinde den Pluspol (rot) der Servos mit 5V des Arduino.
 - Verbinde den Minuspol (braun) der Servos mit GND des Arduino.
 - Verbinde das Steuerkabel (orange) eines Servos mit Pin 9 des Arduino.
 - Verbinde das Steuerkabel (orange) des anderen Servos mit Pin 10 des Arduino.

Jetzt kannst du den folgenden Arduino-Code verwenden, um die Servos entsprechend der gemessenen Entfernung des Ultraschallsensors zu steuern. Der Code verwendet die Servo-Bibliothek und die NewPing-Bibliothek (für den Ultraschallsensor). Du kannst die NewPing-Bibliothek hier herunterladen: <https://bitbucket.org/teckel12/arduino-new-ping/downloads/>

```
cpp
#include <Servo.h>
#include <NewPing.h>

#define TRIG_PIN 7
#define ECHO_PIN 8
#define SERVO_PIN_1 9
#define SERVO_PIN_2 10

Servo servo1;
Servo servo2;
NewPing sonar(TRIG_PIN, ECHO_PIN, 10); // 10 cm Maximalentfernung

void setup() {
  servo1.attach(SERVO_PIN_1);
  servo2.attach(SERVO_PIN_2);
  Serial.begin(9600);
}

void loop() {
  delay(50);
  unsigned int distance = sonar.ping_cm();
  Serial.print("Entfernung: ");
  Serial.print(distance);
  Serial.println(" cm");
}
```

```
if (distance >= 10 && distance <= 40) {
  // Kartesische Koordinaten in Polarkoordinaten umrechnen
  int angle = map(distance, 10, 40, 0, 180);

  // Servos bewegen
  servo1.write(angle);
  servo2.write(180 - angle);

  delay(500); // Verzögerung für die Servobewegung
}
}
```

Dieser Code verwendet die gemessene Entfernung, um die Servos in Abhängigkeit von der Entfernung zu bewegen. Beachte, dass die Servos aufgrund ihrer mechanischen Beschaffenheit möglicherweise nicht exakt auf die gewünschten Winkelpositionen eingestellt werden können, aber dieser Code sollte dir einen guten Ausgangspunkt bieten. Du kannst die Werte in der `map`-Funktion anpassen, um die Servobewegung feiner abzustimmen.

es funktioniert nicht richtig. kannst du beide servos analog zur gemessenen distanz bewegen lassen?

ChatGPT

Entschuldigung für das Missverständnis. Wenn du beide Servos analog zur gemessenen Entfernung bewegen möchtest, kannst du den folgenden Arduino-Code verwenden:

```
cpp
#include <Servo.h>
#include <NewPing.h>

#define TRIG_PIN 7
#define ECHO_PIN 8
#define SERVO_PIN_1 9
#define SERVO_PIN_2 10

Servo servo1;
Servo servo2;
NewPing sonar(TRIG_PIN, ECHO_PIN, 10); // 10 cm Maximalentfernung

void setup() {
  servo1.attach(SERVO_PIN_1);
  servo2.attach(SERVO_PIN_2);
  Serial.begin(9600);
}

void loop() {
  delay(50);
  unsigned int distance = sonar.ping_cm();
  Serial.print("Entfernung: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (distance >= 10 && distance <= 40) {
    // Kartesische Koordinaten in Polarkoordinaten umrechnen
    int servoAngle = map(distance, 10, 40, 0, 180);

    // Servos bewegen analog zur Entfernung
    servo1.write(servoAngle);
    servo2.write(servoAngle);
  }
}
```

```
    delay(500); // Verzögerung für die Servobewegung
  }
}
```

Mit diesem Code werden beide Servos gleichzeitig analog zur gemessenen Entfernung bewegt, wobei der Servowinkel proportional zur Entfernung von 10 cm bis 40 cm skaliert wird. Bitte stelle sicher, dass die Servos korrekt angeschlossen sind und die Mechanik der Servos keine physischen Einschränkungen aufweist, die die Bewegung blockieren könnten.

die beiden servos bewegen sich nur zweimal. sie sollten sich permanent und linear zur gemessenen entfernung bewegen

ChatGPT

Entschuldigung für die vorherigen Missverständnisse. Wenn du möchtest, dass beide Servos sich permanent und linear zur gemessenen Entfernung bewegen, kannst du den folgenden Code verwenden:

```
cpp
#include <Servo.h>
#include <NewPing.h>

#define TRIG_PIN 7
#define ECHO_PIN 8
#define SERVO_PIN_1 9
#define SERVO_PIN_2 10

Servo servo1;
Servo servo2;
NewPing sonar(TRIG_PIN, ECHO_PIN, 10); // 10 cm Maximalentfernung

void setup() {
  servo1.attach(SERVO_PIN_1);
  servo2.attach(SERVO_PIN_2);
  Serial.begin(9600);
}

void loop() {
  delay(50);
  unsigned int distance = sonar.ping_cm();
  Serial.print("Entfernung: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (distance >= 10 && distance <= 40) {
    // Servos bewegen linear zur Entfernung
    int servoAngle = map(distance, 10, 40, 0, 180);
    servo1.write(servoAngle);
    servo2.write(servoAngle);
  } else {
    // Wenn die Entfernung außerhalb des Bereichs liegt, Servos anhalten
    servo1.write(90); // 90 Grad ist die Mitte
    servo2.write(90); // 90 Grad ist die Mitte
  }
}
```

Mit diesem Code bewegen sich die Servos linear zur gemessenen Entfernung im Bereich von 10 cm bis 40 cm. Wenn die Entfernung außerhalb dieses Bereichs liegt, werden die Servos auf

die Mitte (90 Grad) zurückgesetzt. Dadurch sollten die Servos kontinuierlich und linear zur Entfernung reagieren.

kannst du die servos langsamer und einen im anderen Drehsinn bewegen. Der sensor wert wird nicht zuverlässig eingelesen, kannst du das verbessern?

ChatGPT

Um die Servos langsamer zu bewegen und einen im Uhrzeigersinn und den anderen gegen den Uhrzeigersinn zu drehen, und gleichzeitig die Zuverlässigkeit des Sensorwerts zu verbessern, kannst du den folgenden überarbeiteten Code verwenden:

```
cpp
#include <Servo.h>
#include <NewPing.h>

#define TRIG_PIN 7
#define ECHO_PIN 8
#define SERVO_PIN_1 9
#define SERVO_PIN_2 10

Servo servo1;
Servo servo2;
NewPing sonar(TRIG_PIN, ECHO_PIN, 200); // 200 ms Maximalentfernung
(verlangsamt die Abfrage)

void setup() {
  servo1.attach(SERVO_PIN_1);
  servo2.attach(SERVO_PIN_2);
  Serial.begin(9600);
}

void loop() {
  delay(50); // Kleine Verzögerung zwischen den Schleifen
  unsigned int distance = sonar.ping_cm();
  Serial.print("Entfernung: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (distance >= 10 && distance <= 40) {
    // Servos bewegen langsamer und entgegengesetzt
    int servoAngle1 = map(distance, 10, 40, 0, 90); // Im Uhrzeigersinn
    int servoAngle2 = map(distance, 10, 40, 180, 90); // Gegen den
    Uhrzeigersinn

    servo1.write(servoAngle1);
    servo2.write(servoAngle2);

    delay(500); // Verzögerung für die Servobewegung
  } else {
    // Wenn die Entfernung außerhalb des Bereichs liegt, Servos anhalten
    servo1.write(90); // 90 Grad ist die Mitte
    servo2.write(90); // 90 Grad ist die Mitte
  }
}
```

In diesem Code werden die Servos langsamer bewegt (der Bereich wurde von 0 bis 90 Grad angepasst), und einer dreht sich im Uhrzeigersinn, während der andere sich gegen den Uhrzeigersinn dreht. Die NewPing-Abfrage wird alle 200 ms durchgeführt, um die

Zuverlässigkeit des Sensorwerts zu verbessern. Beachte, dass du die Verzögerungszeit für die Servobewegung nach Bedarf anpassen kannst, um die Geschwindigkeit weiter zu reduzieren.

der Serevo an pin 9 bewegt sich in einem kleinen winkel kannst du den vergrößern. kannst du den sensor genauer einlesen und die servos langsam bewegen lassen?

ChatGPT

Um den Servo an Pin 9 in einem größeren Winkelbereich zu bewegen und den Sensor genauer auszulesen, kannst du den Code wie folgt anpassen:

```

cpp
#include <Servo.h>
#include <NewPing.h>

#define TRIG_PIN 7
#define ECHO_PIN 8
#define SERVO_PIN_1 9
#define SERVO_PIN_2 10

Servo servo1;
Servo servo2;
NewPing sonar(TRIG_PIN, ECHO_PIN, 50); // 50 ms Maximalentfernung
(häufigere Messungen für Genauigkeit)

void setup() {
  servo1.attach(SERVO_PIN_1);
  servo2.attach(SERVO_PIN_2);
  Serial.begin(9600);
}

void loop() {
  delay(50); // Kleine Verzögerung zwischen den Schleifen
  unsigned int distance = sonar.ping_cm();
  Serial.print("Entfernung: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (distance >= 10 && distance <= 40) {
    // Servos bewegen langsamer und mit größerem Winkelbereich
    int servoAngle1 = map(distance, 10, 40, 0, 180); // Größerer
Winkelbereich
    int servoAngle2 = 180 - servoAngle1; // Gegenüberliegender Winkel

    servo1.write(servoAngle1);
    servo2.write(servoAngle2);

    delay(1000); // Verzögerung für die Servobewegung (langsamer)
  } else {
    // Wenn die Entfernung außerhalb des Bereichs liegt, Servos anhalten
    servo1.write(90); // 90 Grad ist die Mitte
    servo2.write(90); // 90 Grad ist die Mitte
  }
}

```

In diesem aktualisierten Code habe ich den Winkelbereich des Servos an Pin 9 auf 0 bis 180 Grad erweitert und die Verzögerung für die Servobewegung verlängert, um sie langsamer zu machen. Außerdem wird der Sensor alle 50 ms abgefragt, um genauere Messungen zu ermöglichen.

Damit solltest du eine größere und genauere Bewegung des Servos an Pin 9 sehen, die sich in Abhängigkeit von der gemessenen Entfernung verändert.

kannst du den servo an pin10 auf eine winkelbewegung von 60 grad reduzieren. Die servos bewegen sich immer noch viel zu schnell. kannst du sie viel langsamer bewegen.

ChatGPT

Um den Servo an Pin 10 auf eine Winkelbewegung von 60 Grad zu begrenzen und die Servos langsamer zu bewegen, kannst du den Code wie folgt anpassen:

```
cpp
#include <Servo.h>
#include <NewPing.h>

#define TRIG_PIN 7
#define ECHO_PIN 8
#define SERVO_PIN_1 9
#define SERVO_PIN_2 10

Servo servo1;
Servo servo2;
NewPing sonar(TRIG_PIN, ECHO_PIN, 50); // 50 ms Maximalentfernung
(häufigere Messungen für Genauigkeit)

void setup() {
  servo1.attach(SERVO_PIN_1);
  servo2.attach(SERVO_PIN_2);
  Serial.begin(9600);
}

void loop() {
  delay(50); // Kleine Verzögerung zwischen den Schleifen
  unsigned int distance = sonar.ping_cm();
  Serial.print("Entfernung: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (distance >= 10 && distance <= 40) {
    // Servos bewegen langsamer und auf begrenzte Winkel
    int servoAngle1 = map(distance, 10, 40, 0, 120); // Begrenzter
Winkelbereich (0-120 Grad)
    int servoAngle2 = 60; //Begrenzter Winkel auf 60 Grad für Servo an Pin
10

    servo1.write(servoAngle1);
    servo2.write(servoAngle2);

    delay(2000); // Verzögerung für die Servobewegung (langsamer)
  } else {
    // Wenn die Entfernung außerhalb des Bereichs liegt, Servos anhalten
    servo1.write(60); // 60 Grad ist die Mitte für Servo an Pin 9
    servo2.write(60); // 60 Grad ist die Mitte für Servo an Pin 10
  }
}
```

In diesem Code wurde der Winkel des Servos an Pin 10 auf 60 Grad begrenzt, und die Servos bewegen sich langsamer, da die Verzögerung für die Servobewegung auf 2 Sekunden erhöht

wurde. Dies sollte die Servos viel langsamer und mit den gewünschten Begrenzungen bewegen.

es bewegt sich nur noch servo 9. servo 10 muss auch bewegen?

ChatGPT

Entschuldigung für das Missverständnis. Du hast recht, in meiner vorherigen Antwort habe ich vergessen, die Bewegung für Servo 10 zu implementieren. Hier ist der korrigierte Code:

```
cpp
#include <Servo.h>
#include <NewPing.h>

#define TRIG_PIN 7
#define ECHO_PIN 8
#define SERVO_PIN_1 9
#define SERVO_PIN_2 10

Servo servo1;
Servo servo2;
NewPing sonar(TRIG_PIN, ECHO_PIN, 50); // 50 ms Maximalentfernung
(häufigere Messungen für Genauigkeit)

void setup() {
  servo1.attach(SERVO_PIN_1);
  servo2.attach(SERVO_PIN_2);
  Serial.begin(9600);
}

void loop() {
  delay(50); // Kleine Verzögerung zwischen den Schleifen
  unsigned int distance = sonar.ping_cm();
  Serial.print("Entfernung: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (distance >= 10 && distance <= 40) {
    // Servos bewegen langsamer und auf begrenzte Winkel
    int servoAngle1 = map(distance, 10, 40, 0, 120); // Begrenzter
Winkelbereich (0-120 Grad)
    int servoAngle2 = map(distance, 10, 40, 0, 60); // Begrenzter
Winkelbereich (0-60 Grad)

    servo1.write(servoAngle1);
    servo2.write(servoAngle2);

    delay(2000); // Verzögerung für die Servobewegung (langsamer)
  } else {
    // Wenn die Entfernung außerhalb des Bereichs liegt, Servos anhalten
    servo1.write(60); // 60 Grad ist die Mitte für Servo an Pin 9
    servo2.write(30); // 30 Grad ist die Mitte für Servo an Pin 10
  }
}
```

Jetzt bewegen sich beide Servos entsprechend der gemessenen Entfernung. Servo 1 bewegt sich im Bereich von 0-120 Grad, und Servo 2 bewegt sich im Bereich von 0-60 Grad. Die Verzögerung für die Servobewegung wurde beibehalten, um die Bewegungen langsamer zu machen.

bewege die servos zufälliger sobald der sensor werte einliest

ChatGPT

Um die Servos zufälliger zu bewegen, sobald der Sensor Werte einliest, kannst du eine zufällige Zielposition für die Servos generieren. Hier ist der aktualisierte Code:

```

cpp
#include <Servo.h>
#include <NewPing.h>

#define TRIG_PIN 7
#define ECHO_PIN 8
#define SERVO_PIN_1 9
#define SERVO_PIN_2 10

Servo servo1;
Servo servo2;
NewPing sonar(TRIG_PIN, ECHO_PIN, 50); // 50 ms Maximalentfernung
(häufigere Messungen für Genauigkeit)

void setup() {
  servo1.attach(SERVO_PIN_1);
  servo2.attach(SERVO_PIN_2);
  Serial.begin(9600);
  randomSeed(analogRead(0)); // Zufallsgenerator initialisieren
}

void loop() {
  delay(50); // Kleine Verzögerung zwischen den Schleifen
  unsigned int distance = sonar.ping_cm();
  Serial.print("Entfernung: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (distance >= 10 && distance <= 40) {
    // Zufällige Servo-Positionen generieren
    int servoAngle1 = random(0, 121); // Zufälliger Wert zwischen 0 und 120
    Grad
    int servoAngle2 = random(0, 61); // Zufälliger Wert zwischen 0 und 60
    Grad

    servo1.write(servoAngle1);
    servo2.write(servoAngle2);

    delay(2000); // Verzögerung für die Servobewegung (langsamer)
  } else {
    // Wenn die Entfernung außerhalb des Bereichs liegt, Servos anhalten
    servo1.write(60); // 60 Grad ist die Mitte für Servo an Pin 9
    servo2.write(30); // 30 Grad ist die Mitte für Servo an Pin 10
  }
}

```

In diesem Code wird die `randomSeed`-Funktion verwendet, um den Zufallsgenerator mit einem zufälligen Wert zu initialisieren. Dann werden zufällige Servo-Positionen für beide Servos generiert, wenn der Sensor Werte innerhalb des gewünschten Bereichs liest. Dies sollte die Servos in zufällige Positionen bewegen, sobald der Sensor eine Entfernung misst, die im Bereich von 10 cm bis 40 cm liegt.

kannst du beim einlesen eines sensorwertes eine zufällige bewegungsabfolge von zwei sekunden machen

ChatGPT

Um eine zufällige Bewegungsabfolge der Servos über einen Zeitraum von zwei Sekunden zu erstellen, wenn der Sensor einen Wert einliest, kannst du die `millis()`-Funktion verwenden, um die Zeit zu überwachen und die Servos entsprechend zu steuern. Hier ist der überarbeitete Code:

```

cpp
#include <Servo.h>
#include <NewPing.h>

#define TRIG_PIN 7
#define ECHO_PIN 8
#define SERVO_PIN_1 9
#define SERVO_PIN_2 10

Servo servo1;
Servo servo2;
NewPing sonar(TRIG_PIN, ECHO_PIN, 50); // 50 ms Maximalentfernung
(häufigere Messungen für Genauigkeit)

unsigned long previousMillis = 0;
unsigned long interval = 2000; // 2 Sekunden

void setup() {
  servo1.attach(SERVO_PIN_1);
  servo2.attach(SERVO_PIN_2);
  Serial.begin(9600);
  randomSeed(analogRead(0)); // Zufallsgenerator initialisieren
}

void loop() {
  unsigned long currentMillis = millis();
  unsigned int distance = sonar.ping_cm();
  Serial.print("Entfernung: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (currentMillis - previousMillis >= interval) {
    // Zeitablauf von 2 Sekunden erreicht
    previousMillis = currentMillis;

    if (distance >= 10 && distance <= 40) {
      // Zufällige Servo-Positionen generieren
      int servoAngle1 = random(0, 121); // Zufälliger Wert zwischen 0 und
120 Grad
      int servoAngle2 = random(0, 61); // Zufälliger Wert zwischen 0 und
60 Grad

      servo1.write(servoAngle1);
      servo2.write(servoAngle2);
    } else {
      // Wenn die Entfernung außerhalb des Bereichs liegt, Servos anhalten
      servo1.write(60); // 60 Grad ist die Mitte für Servo an Pin 9
      servo2.write(30); // 30 Grad ist die Mitte für Servo an Pin 10
    }
  }
}

```

```
}

```

In diesem Code wird die Zeit zwischen dem Einlesen des Sensorwerts und der Aktualisierung der Servo-Positionen überwacht. Wenn zwei Sekunden verstrichen sind, werden zufällige Servo-Positionen generiert und die Servos bewegen sich entsprechend. Dies wiederholt sich dann alle zwei Sekunden, solange der Sensorwerte innerhalb des gewünschten Bereichs (10 cm bis 40 cm) liefert.

wenn du einen sensorwert einliest sollen sich die servos zwei sekunden lang zufällig bewegen, danach warten bis erneut ein wert eingelesen wird und danach die servos wieder bewegen als loop

ChatGPT

Ich verstehe, dass du die Servos zwei Sekunden lang zufällig bewegen möchtest, nachdem ein Sensorwert eingelesen wurde, und dann in einer Schleife darauf wartest, bis erneut ein Wert eingelesen wird, bevor die Servos erneut bewegt werden. Hier ist der angepasste Code:

```
cpp
#include <Servo.h>
#include <NewPing.h>

#define TRIG_PIN 7
#define ECHO_PIN 8
#define SERVO_PIN_1 9
#define SERVO_PIN_2 10

Servo servo1;
Servo servo2;
NewPing sonar(TRIG_PIN, ECHO_PIN, 50); // 50 ms Maximalentfernung
(häufigere Messungen für Genauigkeit)

unsigned long previousMillis = 0;
unsigned long interval = 2000; // 2 Sekunden

void setup() {
  servo1.attach(SERVO_PIN_1);
  servo2.attach(SERVO_PIN_2);
  Serial.begin(9600);
  randomSeed(analogRead(0)); // Zufallsgenerator initialisieren
}

void loop() {
  unsigned long currentMillis = millis();
  unsigned int distance = sonar.ping_cm();
  Serial.print("Entfernung: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (distance >= 10 && distance <= 40) {
    if (currentMillis - previousMillis >= interval) {
      // Zeitablauf von 2 Sekunden erreicht
      previousMillis = currentMillis;

      // Zufällige Servo-Positionen generieren
      int servoAngle1 = random(0, 121); // Zufälliger Wert zwischen 0 und
120 Grad
      int servoAngle2 = random(0, 61); // Zufälliger Wert zwischen 0 und
60 Grad

```

```

        servo1.write(servoAngle1);
        servo2.write(servoAngle2);
    }
} else {
    // Wenn die Entfernung außerhalb des Bereichs liegt, Servos anhalten
    servo1.write(60); // 60 Grad ist die Mitte für Servo an Pin 9
    servo2.write(30); // 30 Grad ist die Mitte für Servo an Pin 10
}
}
}

```

Mit diesem Code werden die Servos zwei Sekunden lang zufällig bewegt, nachdem ein gültiger Sensorwert eingelesen wurde. Dann wird in einer Schleife gewartet, bis ein neuer Sensorwert eingelesen wird, bevor die Servos erneut bewegt werden. Der Code überprüft währenddessen kontinuierlich den Sensorwert und führt die zufällige Bewegung aus, wenn der Sensorwert gültig ist.

kannst du den analog Servo an pin10 in 4 sekunden von 0 -60 Grad und wieder zurückbewegen

ChatGPT

Um den analogen Servo an Pin 10 innerhalb von 4 Sekunden von 0 auf 60 Grad und dann wieder zurück zu bewegen, kannst du die `millis()`-Funktion verwenden, um die Zeit zu überwachen und die Servo-Position entsprechend zu steuern. Hier ist der Code:

```

cpp
#include <Servo.h>
#include <NewPing.h>

#define TRIG_PIN 7
#define ECHO_PIN 8
#define SERVO_PIN_1 9
#define SERVO_PIN_2 10

Servo servo1;
Servo servo2;
NewPing sonar(TRIG_PIN, ECHO_PIN, 50); // 50 ms Maximalentfernung
(häufigere Messungen für Genauigkeit)

unsigned long previousMillis = 0;
unsigned long interval = 4000; // 4 Sekunden
int servoAngle2 = 0; // Startposition

void setup() {
    servo1.attach(SERVO_PIN_1);
    servo2.attach(SERVO_PIN_2);
    Serial.begin(9600);
    randomSeed(analogRead(0)); // Zufallsgenerator initialisieren
}

void loop() {
    unsigned long currentMillis = millis();
    unsigned int distance = sonar.ping_cm();
    Serial.print("Entfernung: ");
    Serial.print(distance);
    Serial.println(" cm");

    if (distance >= 10 && distance <= 40) {
        if (currentMillis - previousMillis >= interval) {
            // Zeitablauf von 4 Sekunden erreicht

```

```

    previousMillis = currentMillis;

    // Zufällige Servo-Positionen für Servo an Pin 9 generieren
    int servoAngle1 = random(0, 121); // Zufälliger Wert zwischen 0 und
120 Grad

    servo1.write(servoAngle1);

    // Servo an Pin 10 von 0 auf 60 Grad und zurück bewegen
    for (int angle = 0; angle <= 60; angle++) {
        servoAngle2 = angle;
        servo2.write(servoAngle2);
        delay(40); // Kleine Verzögerung für sanfte Bewegung
    }

    for (int angle = 60; angle >= 0; angle--) {
        servoAngle2 = angle;
        servo2.write(servoAngle2);
        delay(40); // Kleine Verzögerung für sanfte Bewegung
    }
}
} else {
    // Wenn die Entfernung außerhalb des Bereichs liegt, Servos anhalten
    servo1.write(60); // 60 Grad ist die Mitte für Servo an Pin 9
    servo2.write(servoAngle2); // Servo an Pin 10 auf der zuletzt gesetzten
Position belassen
}
}
}

```

Dieser Code bewegt den analogen Servo an Pin 10 in einem Zeitraum von 4 Sekunden von 0 auf 60 Grad und dann wieder zurück, während der Servo an Pin 9 eine zufällige Position einnimmt, wenn ein gültiger Sensorwert vorliegt. Bitte beachte, dass die Verzögerungen in der Schleife die Bewegung des Servos verlangsamen, um eine sanfte Bewegung zu ermöglichen.

kannst du gleichzeitig analog servo an Pin 9 in zyklen von einer Sekunde von 0-60 grad bewegen, dass soll zufällig aber häufig passieren

ChatGPT

Um den analogen Servo an Pin 9 zufällig alle paar Sekunden von 0 auf 60 Grad und zurück zu bewegen, während der andere Servo (an Pin 10) auf seine letzte Position eingestellt bleibt, kannst du den folgenden Code verwenden:

```

cpp
#include <Servo.h>
#include <NewPing.h>

#define TRIG_PIN 7
#define ECHO_PIN 8
#define SERVO_PIN_1 9
#define SERVO_PIN_2 10

Servo servo1;
Servo servo2;
NewPing sonar(TRIG_PIN, ECHO_PIN, 50); // 50 ms Maximalentfernung
(häufigere Messungen für Genauigkeit)

unsigned long previousMillis = 0;

```

```

unsigned long interval = 2000; // Zeitintervall für Servo an Pin 9 (2
Sekunden)

void setup() {
  servo1.attach(SERVO_PIN_1);
  servo2.attach(SERVO_PIN_2);
  Serial.begin(9600);
  randomSeed(analogRead(0)); // Zufallsgenerator initialisieren
}

void loop() {
  unsigned long currentMillis = millis();
  unsigned int distance = sonar.ping_cm();
  Serial.print("Entfernung: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (currentMillis - previousMillis >= interval) {
    // Zeitablauf von 2 Sekunden für Servo an Pin 9 erreicht
    previousMillis = currentMillis;

    // Zufällige Servo-Positionen für Servo an Pin 9 generieren
    int servoAngle1 = random(0, 61); // Zufälliger Wert zwischen 0 und 60
Grad

    servo1.write(servoAngle1);
  }

  // Servo an Pin 10 auf der letzten Position belassen
  // (Servo an Pin 10 bewegt sich nicht)
  delay(50); // Kleine Verzögerung zwischen den Schleifen
}

```

In diesem Code wird der Servo an Pin 9 alle paar Sekunden zufällig zwischen 0 und 60 Grad bewegt, während der Servo an Pin 10 seine Position beibehält. Die Verzögerung von 50 ms zwischen den Schleifen sorgt dafür, dass der Servo an Pin 10 nicht bewegt wird. Du kannst das Zeitintervall, in dem der Servo an Pin 9 bewegt wird, durch Anpassen der `interval`-Variable ändern.

servo an pin 10 soll sich in 4 sekunden von 0 -60 grad bewegen und wieder zurück, gleichzeitig soll sich servo an pin 9 ebenfalls und zufällig bewegen

ChatGPT

Um den Servo an Pin 10 innerhalb von 4 Sekunden von 0 auf 60 Grad und zurück zu bewegen, während sich der Servo an Pin 9 zufällig bewegt, kannst du den folgenden Code verwenden:

```

cpp
#include <Servo.h>
#include <NewPing.h>

#define TRIG_PIN 7
#define ECHO_PIN 8
#define SERVO_PIN_1 9
#define SERVO_PIN_2 10

Servo servo1;
Servo servo2;

```


ChatGPT + Arduino Task 4 Zeichenmaschine

```
NewPing sonar(TRIG_PIN, ECHO_PIN, 50); // 50 ms Maximalentfernung
(häufigere Messungen für Genauigkeit)

unsigned long previousMillis = 0;
unsigned long interval = 4000; // 4 Sekunden für Servo an Pin 10
int servoAngle1 = 0; // Startposition für Servo an Pin 1
int servoAngle2 = 0; // Startposition für Servo an Pin 2

void setup() {
  servo1.attach(SERVO_PIN_1);
  servo2.attach(SERVO_PIN_2);
  Serial.begin(9600);
  randomSeed(analogRead(0)); // Zufallsgenerator initialisieren
}

void loop() {
  unsigned long currentMillis = millis();
  unsigned int distance = sonar.ping_cm();
  Serial.print("Entfernung: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (currentMillis - previousMillis >= interval) {
    // Zeitablauf von 4 Sekunden für Servo an Pin 10 erreicht
    previousMillis = currentMillis;

    // Servo an Pin 10 von 0 auf 60 Grad und zurück bewegen
    for (int angle = 0; angle <= 60; angle++) {
      servoAngle2 = angle;
      servo2.write(servoAngle2);
      delay(40); // Kleine Verzögerung für sanfte Bewegung
    }

    for (int angle = 60; angle >= 0; angle--) {
      servoAngle2 = angle;
      servo2.write(servoAngle2);
      delay(40); // Kleine Verzögerung für sanfte Bewegung
    }
  }

  // Zufällige Servo-Positionen für Servo an Pin 9 generieren
  servoAngle1 = random(0, 121); // Zufälliger Wert zwischen 0 und 120 Grad
  servo1.write(servoAngle1);

  // Eine kleine Verzögerung zwischen den Schleifen, um Servo an Pin 9
  beweglicher zu machen
  delay(1000);
}
```

In diesem Code bewegt sich der Servo an Pin 10 in einem Zeitraum von 4 Sekunden von 0 auf 60 Grad und wieder zurück. Gleichzeitig wird der Servo an Pin 9 zufällig bewegt. Beachte, dass eine kleine Verzögerung von 1 Sekunde zwischen den Schleifen vorhanden ist, um die Bewegung von Servo an Pin 9 sichtbarer zu machen. Du kannst die Verzögerungswerte je nach Bedarf anpassen.

im Bewegungszeitraum von servo an pin10 soll sich servo 9 5mal zufällig bewegen

ChatGPT

Verstanden, du möchtest, dass sich der Servo an Pin 9 während des Bewegungszeitraums des Servos an Pin 10 fünfmal zufällig bewegt. Hier ist der angepasste Code:

```

cpp
#include <Servo.h>
#include <NewPing.h>

#define TRIG_PIN 7
#define ECHO_PIN 8
#define SERVO_PIN_1 9
#define SERVO_PIN_2 10

Servo servo1;
Servo servo2;
NewPing sonar(TRIG_PIN, ECHO_PIN, 50); // 50 ms Maximalentfernung
(häufigere Messungen für Genauigkeit)

unsigned long previousMillis = 0;
unsigned long interval = 4000; // 4 Sekunden für Servo an Pin 10
int servoAngle1 = 0; // Startposition für Servo an Pin 1
int servoAngle2 = 0; // Startposition für Servo an Pin 2

void setup() {
  servo1.attach(SERVO_PIN_1);
  servo2.attach(SERVO_PIN_2);
  Serial.begin(9600);
  randomSeed(analogRead(0)); // Zufallsgenerator initialisieren
}

void loop() {
  unsigned long currentMillis = millis();
  unsigned int distance = sonar.ping_cm();
  Serial.print("Entfernung: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (currentMillis - previousMillis >= interval) {
    // Zeitablauf von 4 Sekunden für Servo an Pin 10 erreicht
    previousMillis = currentMillis;

    // Servo an Pin 10 von 0 auf 60 Grad und zurück bewegen
    for (int angle = 0; angle <= 60; angle++) {
      servoAngle2 = angle;
      servo2.write(servoAngle2);
      delay(40); // Kleine Verzögerung für sanfte Bewegung
    }

    for (int angle = 60; angle >= 0; angle--) {
      servoAngle2 = angle;
      servo2.write(servoAngle2);
      delay(40); // Kleine Verzögerung für sanfte Bewegung
    }

    // Servo an Pin 9 soll sich 5-mal zufällig bewegen
    for (int i = 0; i < 5; i++) {
      servoAngle1 = random(0, 121); // Zufälliger Wert zwischen 0 und 120
      Grad
      servo1.write(servoAngle1);
      delay(1000); // Verzögerung von 1 Sekunde zwischen den Bewegungen
    }
  }
}

```

In diesem aktualisierten Code wird der Servo an Pin 9 während des Bewegungszeitraums des Servos an Pin 10 fünfmal zufällig bewegt. Es gibt eine Verzögerung von 1 Sekunde zwischen den einzelnen Servo-Bewegungen von Servo 9, um die Zufälligkeit und das Timing sicherzustellen.

während des bewegungszyklus von Servo an pin 10, soll sich gleichzeitig servo an pin 9 mindestens 5 mal zufällig bewegen

ChatGPT

Ich verstehe dein Anliegen. Du möchtest, dass sich der Servo an Pin 9 mindestens 5 Mal zufällig bewegt, während der Servo an Pin 10 sich über einen Zeitraum von 4 Sekunden von 0 auf 60 Grad und zurück bewegt. Hier ist der überarbeitete Code:

```

cpp
#include <Servo.h>
#include <NewPing.h>

#define TRIG_PIN 7
#define ECHO_PIN 8
#define SERVO_PIN_1 9
#define SERVO_PIN_2 10

Servo servo1;
Servo servo2;
NewPing sonar(TRIG_PIN, ECHO_PIN, 50); // 50 ms Maximalentfernung
(häufigere Messungen für Genauigkeit)

unsigned long previousMillis = 0;
unsigned long interval = 4000; // 4 Sekunden für Servo an Pin 10
int servoAngle1 = 0; // Startposition für Servo an Pin 1
int servoAngle2 = 0; // Startposition für Servo an Pin 2
int servoMovementCount = 0; // Zählt die Bewegungen von Servo an Pin 9

void setup() {
  servo1.attach(SERVO_PIN_1);
  servo2.attach(SERVO_PIN_2);
  Serial.begin(9600);
  randomSeed(analogRead(0)); // Zufallsgenerator initialisieren
}

void loop() {
  unsigned long currentMillis = millis();
  unsigned int distance = sonar.ping_cm();
  Serial.print("Entfernung: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (currentMillis - previousMillis >= interval) {
    // Zeitablauf von 4 Sekunden für Servo an Pin 10 erreicht
    previousMillis = currentMillis;

    // Servo an Pin 10 von 0 auf 60 Grad und zurück bewegen
    for (int angle = 0; angle <= 60; angle++) {
      servoAngle2 = angle;
      servo2.write(servoAngle2);
      delay(40); // Kleine Verzögerung für sanfte Bewegung
    }

    for (int angle = 60; angle >= 0; angle--) {

```

```

servoAngle2 = angle;
servo2.write(servoAngle2);
delay(40); // Kleine Verzögerung für sanfte Bewegung
}

// Servo an Pin 9 soll sich mindestens 5 Mal zufällig bewegen
while (servoMovementCount < 5) {
  servoAngle1 = random(0, 121); // Zufälliger Wert zwischen 0 und 120
Grad
  servo1.write(servoAngle1);
  delay(1000); // Verzögerung von 1 Sekunde zwischen den Bewegungen
  servoMovementCount++;
}

servoMovementCount = 0; // Zurücksetzen des Bewegungszählers
}
}

```

In diesem Code bewegt sich der Servo an Pin 9 mindestens 5 Mal zufällig, während der Servo an Pin 10 seinen Bewegungszyklus von 0 auf 60 Grad und zurück durchführt. Der Bewegungszähler `servoMovementCount` wird verwendet, um sicherzustellen, dass mindestens 5 Bewegungen von Servo 9 ausgeführt werden. Nachdem diese Bedingung erfüllt ist, wird der Zähler zurückgesetzt.

beide ssvos müssen sich gleichzeitig bewegen

ChatGPT

Um beide Servos gleichzeitig zu bewegen, während der Servo an Pin 10 sich über einen Zeitraum von 4 Sekunden von 0 auf 60 Grad und zurück bewegt und der Servo an Pin 9 mindestens 5 Mal zufällig positioniert wird, kannst du den folgenden Code verwenden:

```

cpp
#include <Servo.h>
#include <NewPing.h>

#define TRIG_PIN 7
#define ECHO_PIN 8
#define SERVO_PIN_1 9
#define SERVO_PIN_2 10

Servo servo1;
Servo servo2;
NewPing sonar(TRIG_PIN, ECHO_PIN, 50); // 50 ms Maximalentfernung
(häufigere Messungen für Genauigkeit)

unsigned long previousMillis = 0;
unsigned long interval = 4000; // 4 Sekunden für Servo an Pin 10
int servoAngle1 = 0; // Startposition für Servo an Pin 1
int servoAngle2 = 0; // Startposition für Servo an Pin 2
int servoMovementCount = 0; // Zählt die Bewegungen von Servo an Pin 9

void setup() {
  servo1.attach(SERVO_PIN_1);
  servo2.attach(SERVO_PIN_2);
  Serial.begin(9600);
  randomSeed(analogRead(0)); // Zufallsgenerator initialisieren
}

```

```

void loop() {
  unsigned long currentMillis = millis();
  unsigned int distance = sonar.ping_cm();
  Serial.print("Entfernung: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (currentMillis - previousMillis >= interval) {
    // Zeitablauf von 4 Sekunden für Servo an Pin 10 erreicht
    previousMillis = currentMillis;

    // Servo an Pin 10 von 0 auf 60 Grad und zurück bewegen
    for (int angle = 0; angle <= 60; angle++) {
      servoAngle2 = angle;
      servo2.write(servoAngle2);

      // Zufällige Servo-Positionen für Servo an Pin 9 generieren
      servoAngle1 = random(0, 121); // Zufälliger Wert zwischen 0 und 120
Grad
      servo1.write(servoAngle1);

      delay(40); // Kleine Verzögerung für sanfte Bewegung
    }

    for (int angle = 60; angle >= 0; angle--) {
      servoAngle2 = angle;
      servo2.write(servoAngle2);

      // Zufällige Servo-Positionen für Servo an Pin 9 generieren
      servoAngle1 = random(0, 121); // Zufälliger Wert zwischen 0 und 120
Grad
      servo1.write(servoAngle1);

      delay(40); // Kleine Verzögerung für sanfte Bewegung
    }

    // Servo an Pin 9 soll sich mindestens 5 Mal zufällig bewegen
    while (servoMovementCount < 5) {
      servoAngle1 = random(0, 121); // Zufälliger Wert zwischen 0 und 120
Grad
      servo1.write(servoAngle1);
      delay(1000); // Verzögerung von 1 Sekunde zwischen den Bewegungen
      servoMovementCount++;
    }

    servoMovementCount = 0; // Zurücksetzen des Bewegungszählers
  }
}

```

In diesem Code werden beide Servos gleichzeitig bewegt. Der Servo an Pin 10 bewegt sich von 0 auf 60 Grad und zurück, während der Servo an Pin 9 mindestens 5 Mal zufällig positioniert wird. Dies wird durch die gleichzeitige Anpassung der Servo-Positionen in beiden Schleifen erreicht.

perfekt, starte die schlaufe aber erst wenn und jedesmal wenn ein sensorwert des Ultraschalsensors eingelesen wird

ChatGPT

Um die Schleife zu starten, wenn ein Sensorwert des Ultraschallsensors eingelesen wird, und dies jedes Mal zu wiederholen, kannst du den folgenden Code verwenden:

```

cpp
#include <Servo.h>
#include <NewPing.h>

#define TRIG_PIN 7
#define ECHO_PIN 8
#define SERVO_PIN_1 9
#define SERVO_PIN_2 10

Servo servo1;
Servo servo2;
NewPing sonar(TRIG_PIN, ECHO_PIN, 50); // 50 ms Maximalentfernung
(häufigere Messungen für Genauigkeit)

bool servoMovement = false; // Flag, um die Servo-Bewegung zu steuern
unsigned long interval = 4000; // 4 Sekunden für Servo an Pin 10
int servoAngle1 = 0; // Startposition für Servo an Pin 1
int servoAngle2 = 0; // Startposition für Servo an Pin 2
int servoMovementCount = 0; // Zählt die Bewegungen von Servo an Pin 9

void setup() {
  servo1.attach(SERVO_PIN_1);
  servo2.attach(SERVO_PIN_2);
  Serial.begin(9600);
  randomSeed(analogRead(0)); // Zufallsgenerator initialisieren
}

void loop() {
  unsigned int distance = sonar.ping_cm();
  Serial.print("Entfernung: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (distance >= 10 && distance <= 40) {
    // Sensorwert im gewünschten Bereich
    servoMovement = true;
  } else {
    // Sensorwert außerhalb des Bereichs
    servoMovement = false;
  }

  if (servoMovement) {
    // Servo-Bewegung aktivieren
    unsigned long currentMillis = millis();

    if (currentMillis - previousMillis >= interval) {
      // Zeitablauf von 4 Sekunden für Servo an Pin 10 erreicht
      previousMillis = currentMillis;

      // Servo an Pin 10 von 0 auf 60 Grad und zurück bewegen
      for (int angle = 0; angle <= 60; angle++) {
        servoAngle2 = angle;
        servo2.write(servoAngle2);

        // Zufällige Servo-Positionen für Servo an Pin 9 generieren
        servoAngle1 = random(0, 121); // Zufälliger Wert zwischen 0 und 120
        Grad
        servo1.write(servoAngle1);

```

```

    delay(40); // Kleine Verzögerung für sanfte Bewegung
  }

  for (int angle = 60; angle >= 0; angle--) {
    servoAngle2 = angle;
    servo2.write(servoAngle2);

    // Zufällige Servo-Positionen für Servo an Pin 9 generieren
    servoAngle1 = random(0, 121); // Zufälliger Wert zwischen 0 und 120
Grad
    servo1.write(servoAngle1);

    delay(40); // Kleine Verzögerung für sanfte Bewegung
  }

  // Servo an Pin 9 soll sich mindestens 5 Mal zufällig bewegen
  while (servoMovementCount < 5) {
    servoAngle1 = random(0, 121); // Zufälliger Wert zwischen 0 und 120
Grad
    servo1.write(servoAngle1);
    delay(1000); // Verzögerung von 1 Sekunde zwischen den Bewegungen
    servoMovementCount++;
  }

  servoMovementCount = 0; // Zurücksetzen des Bewegungszählers
}
}
}

```

In diesem Code wird die Schleife nur gestartet, wenn ein gültiger Sensorwert im Bereich von 10 cm bis 40 cm erkannt wird. Die Schleife wird jedes Mal ausgeführt, wenn dieser Bedingung entsprochen wird.

arduino meldet diesen Fehler: Compilation error: 'previousMillis' was not declared in this scope

ChatGPT

Es tut mir leid für das Missverständnis. Es scheint, dass ich eine wichtige Variable, `previousMillis`, versehentlich aus dem vorherigen Codeausschnitt entfernt habe. Du kannst dies beheben, indem du die `previousMillis`-Variable wie folgt deklarierst:

```

cpp
unsigned long previousMillis = 0;

```

Hier ist der vollständige Code mit dieser Ergänzung:

```

cpp
#include <Servo.h>
#include <NewPing.h>

#define TRIG_PIN 7
#define ECHO_PIN 8
#define SERVO_PIN_1 9
#define SERVO_PIN_2 10

Servo servo1;
Servo servo2;
NewPing sonar(TRIG_PIN, ECHO_PIN, 50); // 50 ms Maximalentfernung
(häufigere Messungen für Genauigkeit)

```

```

bool servoMovement = false; // Flag, um die Servo-Bewegung zu steuern
unsigned long interval = 4000; // 4 Sekunden für Servo an Pin 10
int servoAngle1 = 0; // Startposition für Servo an Pin 1
int servoAngle2 = 0; // Startposition für Servo an Pin 2
int servoMovementCount = 0; // Zählt die Bewegungen von Servo an Pin 9
unsigned long previousMillis = 0; // Zeitstempel für die Zeitmessung

void setup() {
  servo1.attach(SERVO_PIN_1);
  servo2.attach(SERVO_PIN_2);
  Serial.begin(9600);
  randomSeed(analogRead(0)); // Zufallsgenerator initialisieren
}

void loop() {
  unsigned int distance = sonar.ping_cm();
  Serial.print("Entfernung: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (distance >= 10 && distance <= 40) {
    // Sensorwert im gewünschten Bereich
    servoMovement = true;
  } else {
    // Sensorwert außerhalb des Bereichs
    servoMovement = false;
  }

  if (servoMovement) {
    // Servo-Bewegung aktivieren
    unsigned long currentMillis = millis();

    if (currentMillis - previousMillis >= interval) {
      // Zeitablauf von 4 Sekunden für Servo an Pin 10 erreicht
      previousMillis = currentMillis;

      // Servo an Pin 10 von 0 auf 60 Grad und zurück bewegen
      for (int angle = 0; angle <= 60; angle++) {
        servoAngle2 = angle;
        servo2.write(servoAngle2);

        // Zufällige Servo-Positionen für Servo an Pin 9 generieren
        servoAngle1 = random(0, 121); // Zufälliger Wert zwischen 0 und 120
Grad
        servo1.write(servoAngle1);

        delay(40); // Kleine Verzögerung für sanfte Bewegung
      }

      for (int angle = 60; angle >= 0; angle--) {
        servoAngle2 = angle;
        servo2.write(servoAngle2);

        // Zufällige Servo-Positionen für Servo an Pin 9 generieren
        servoAngle1 = random(0, 121); // Zufälliger Wert zwischen 0 und 120
Grad
        servo1.write(servoAngle1);

        delay(40); // Kleine Verzögerung für sanfte Bewegung
      }
    }
  }
}

```

ChatGPT + Arduino Task 4 Zeichenmaschine

```
    // Servo an Pin 9 soll sich mindestens 5 Mal zufällig bewegen
    while (servoMovementCount < 5) {
        servoAngle1 = random(0, 121); // Zufälliger Wert zwischen 0 und 120
Grad
        servol.write(servoAngle1);
        delay(1000); // Verzögerung von 1 Sekunde zwischen den Bewegungen
        servoMovementCount++;
    }

    servoMovementCount = 0; // Zurücksetzen des Bewegungszählers
}
}
```

Mit dieser Änderung sollte der Code funktionieren und die `previousMillis`-Variable deklarieren, wie sie benötigt wird.