

Blick in die Zukunft

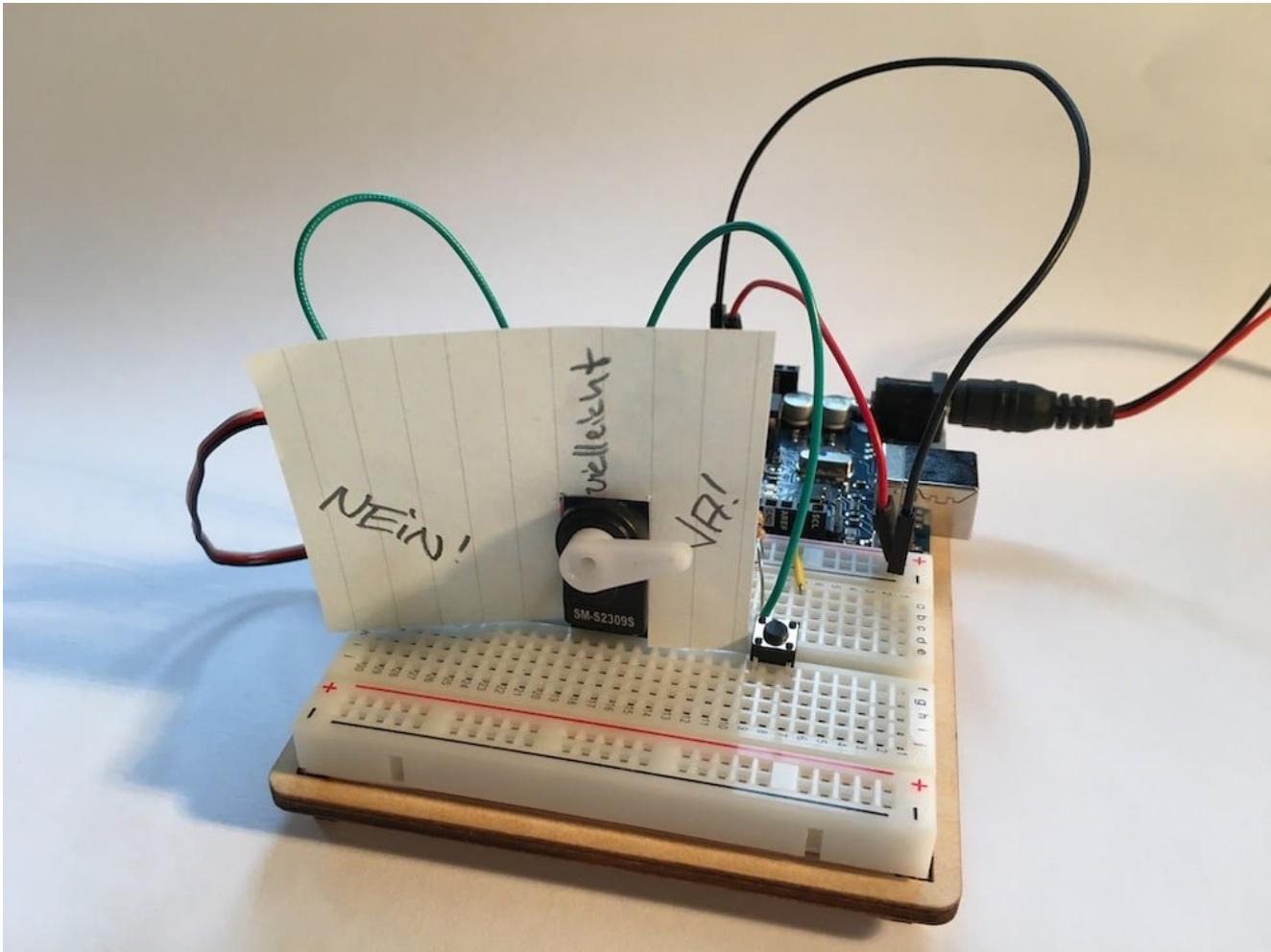


Um es gleich von vorneweg zuzugeben: Die nächsten Lottozahlen wirst du mit diesem Projekt nicht vorhersagen. Dafür lernst du aber, wie du einen Servo-Motor ansteuerst und ihn mithilfe von ein paar Zufallszahlen „in die Zukunft schauen“ lässt.

Für dieses Arduino Projekt benötigst du:

- Arduino Board
- Breadboard
- Servo-Motor
- Kondensator
- 10k Ohm Widerstand
- Schalter
- Kabel

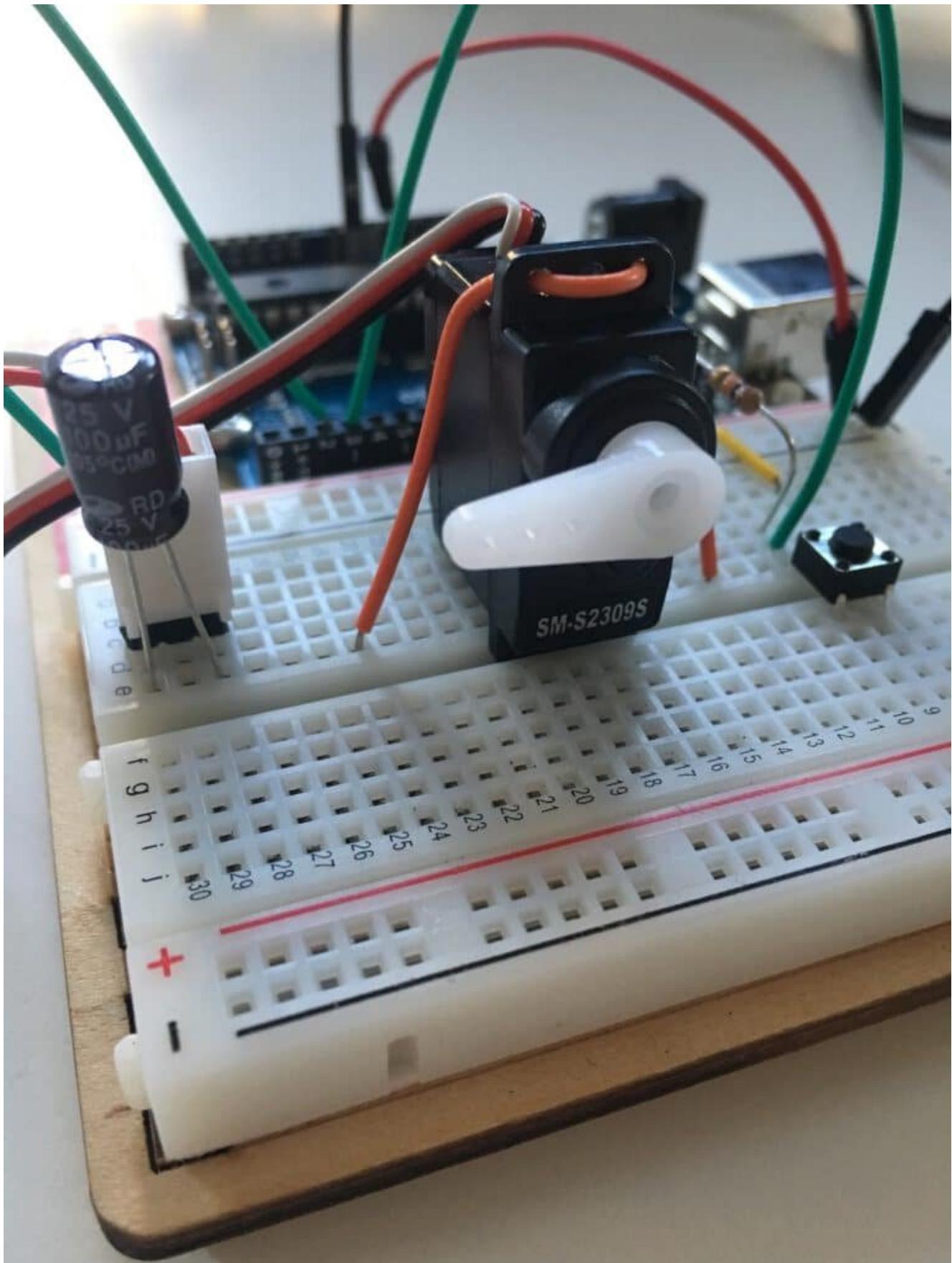
Simpler Aufbau – mit einem kleinen Haken



Dieses Projekt auf dem Breadboard aufzubauen, ist kein Problem. Setze den Servo-Motor in die Mitte des Breadboards und befestige ihn z.B. mit einem Kabel, wie auf dem Foto unten zu sehen. Das ist nicht die stabilste Lösung, aber sie reicht aus, um ihn an Ort und Stelle zu halten.

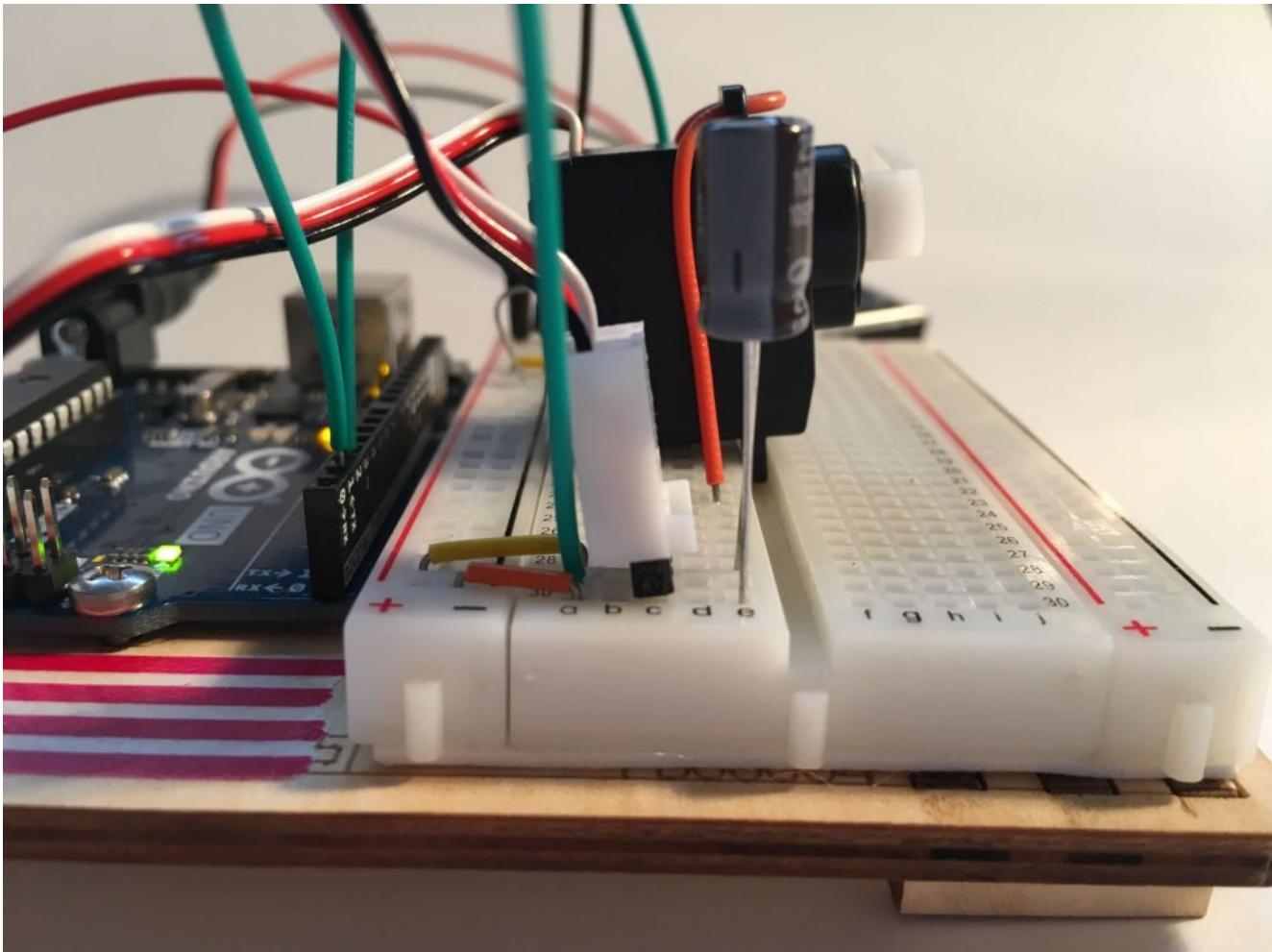
Den Stecker des Motors steckst du links daneben auf das Breadboards und verbindest Plus (rot) und Minus (schwarz) mit der Stromleiste. In die Mitte – meist das weiße Kabel – setzt du eine Verbindung mit dem Digitalpin 2 am Arduino.

Übrigens: [Alles Wissenswerte über Servos erfährst du in diesem Tutorial.](#)



Wenn du einen Elektrolytkondensator verwendest (so wie ich in diesem Projekt), musst du jetzt aufpassen. Setze den Kondensator auf das Board und verbinde das kurze Ende mit dem Minuspol. Das lange steckst du zwei Löcher weiter an den

Pluspol.



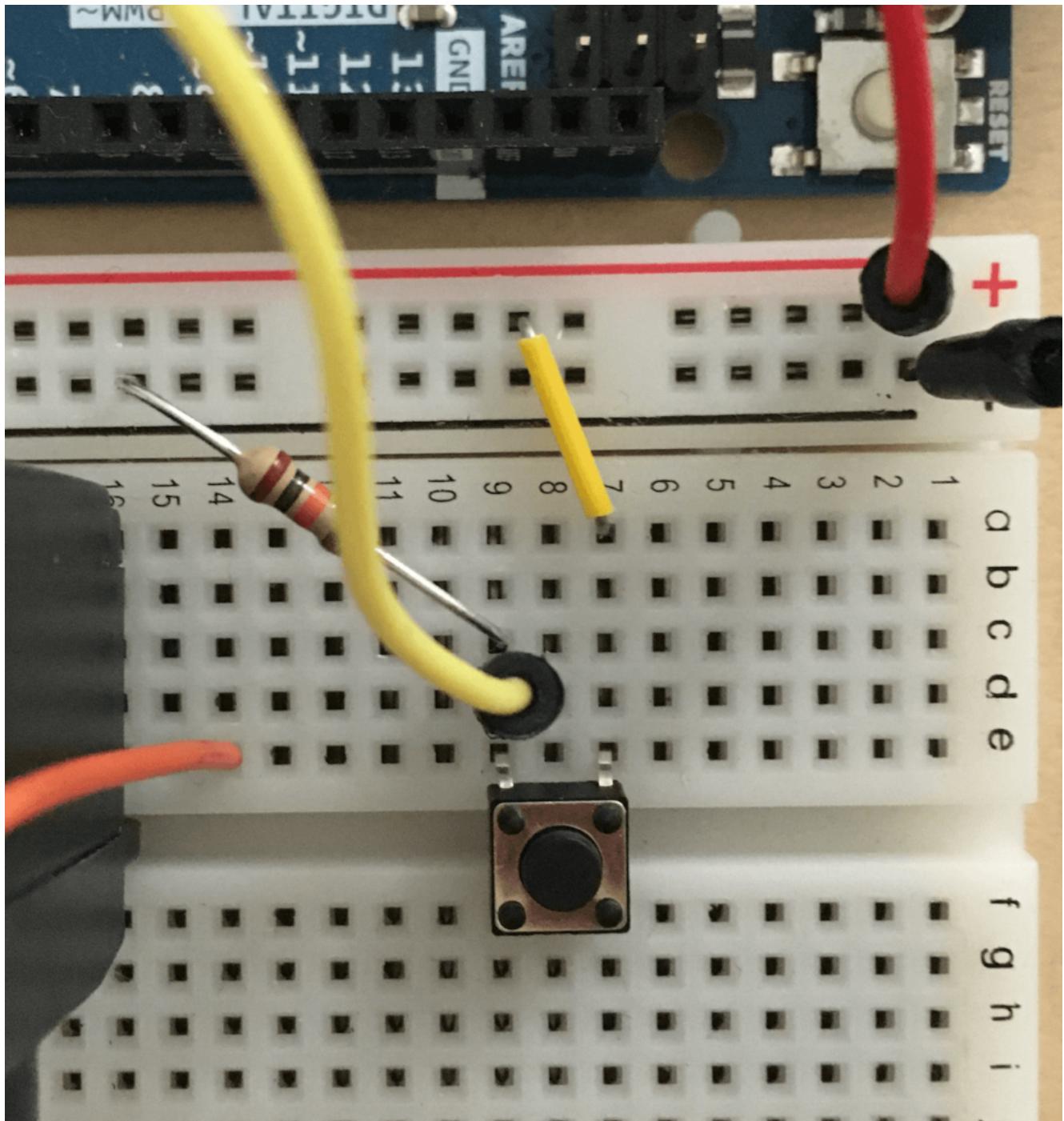
Wichtig: Achte unbedingt darauf, den Kondensator richtig herum anzuschließen.

Vergewissere dich noch einmal, dass der Kondensator richtig herum eingebaut ist – auch anhand der Beschriftung am Kondensator selbst. **Ihn falsch herum einzubauen, kann gefährlich werden.** Denn dann kann er explodieren. Wie das aussieht und weitere Grundlagen zu Kondensatoren lernst du im folgenden Video.

Jetzt benötigst du noch einen Schalter, den du drückst, nachdem du deine Frage gestellt hast und der die Wahrsagerei in Gang setzt.

Den Schalter baust du über die Brücke deines Breadboards,

sodass die zwei oberen Pins auf der einen und zwei unteren Pins auf der unteren Seite eingesteckt sind. Den Pin rechts oben verbindest du mit Plus. Der Pin links oben kommt per Kabel an den digitalen Eingang 3 deines Arduinos. Zusätzlich verbindest du ihn noch einen 10k Ohm Widerstand mit Minus. Wenn du fertig bist, sollte das ungefähr so aussehen:



Damit bist du fertig mit dem Aufbau dieses Projekts. Das einzige, was noch fehlt, ist eine Anzeige für die Antworten.

Die einfachste aller Anzeigen

Ich habe hierfür einfach ein Blatt Papier so ausgeschnitten, dass es um den Servo-Moto herum passt. Auf die linke Seite kommt das **Nein**, rechts das **Ja** und in die Mitte das **Vielleicht**.

Sicher bist du kreativer und machst etwas hübscheres. ☺ Damit der Code unten jedoch zur Anzeige passt, achte darauf, die Antworten wie oben genannt zu verteilen.

Zufallszahlen generieren

Das Herzstück deines Projekts sind Zufallszahlen. Dein Arduino soll ja deine Fragen entweder mit *Ja*, *Nein* oder *Vielleicht* beantworten. Das heißt, dass du 3 Zahlen generieren musst, die jeweils für eine der 3 Antworten stehen.

Das geht ziemlich einfach. Alles was du brauchst, ist die Funktion *random()*.

Für dieses Projekt möchten wir einfach die Zahlen 0, 1 oder 2. Hierfür tragen wir zwischen den Klammern der random-Funktion die Zahl 3 ein. Verwirrend? Auf den ersten Blick schon, aber du musst – wie so oft beim Programmieren – beachten, dass die letzte Zahl exklusiv ist. Das heißt, sie ist die äußere Grenze und wird nicht hinzugezählt. So als würdest du schreiben: < 3.

[Lerne hier mehr über Zufallszahlen auf dem Arduino.](#)

Der fertige Code

Jetzt benötigst du nur noch den Sketch, der deinen Wahrsager zum Laufen bringt. Die Generierung der Zufallszahlen haben wir uns schon angeschaut. Außerdem wichtig ist die Steuerung des Servo-Motors.

Lade den folgenden Code auf deinen Arduino und schon kann es losgehen.

```
#include <Servo.h> //Servo-Bibliothek einbinden
Servo myServo; //Servo-Objekt "myServo" erstellen

int angle; //Position, die der Servo anfahren soll
int switchState = 0; //Status des Schalters
int reply; //Generierte Zufallszahl

void setup() {

pinMode(3, INPUT);
myServo.attach(2); //Der Servo liegt an Pin 2
myServo.write(90); //1. Position des Servos - 90°

Serial.begin(9600);
}

void loop() {

switchState = digitalRead(3); //Lesen, ob der Schalter
gedrückt wurde

if(switchState == 1){
    reply = random(3); //Zufallszahl von 0 bis 2 generieren
    Serial.print(reply);
    switch(reply){
        case 0:
            myServo.write(0); //Falls 0, Servo nach links fahren -
0°
            delay(2000);
            break;

        case 1:
            myServo.write(90); //Falls 1, Servo nach oben fahren -
90°
            delay(2000);
            break;

        case 2:
            myServo.write(180); //Falls 2, Servo nach rechts fahren
- 180°
            delay(2000);
    }
}
```

```
        break;  
    }  
}  
}
```

Wie geht es weiter?

Du hast jetzt ein einfaches Grundgerüst für deinen Wahrsager. Sei kreativ und lass dir ein paar Verbesserungen einfallen! Zum Beispiel eine [Anzeige der Antworten auf einem LCD-Display](#). Oder eine LED, die aufleuchtet, wenn der Schalter gedrückt wird.