Arduino Uno

1. Mikrocontroller 328

Das ist das „Gehirn" des Arduino. Es verarbeitet alle Befehle und steuert, was passiert.

2. Analoge Pins

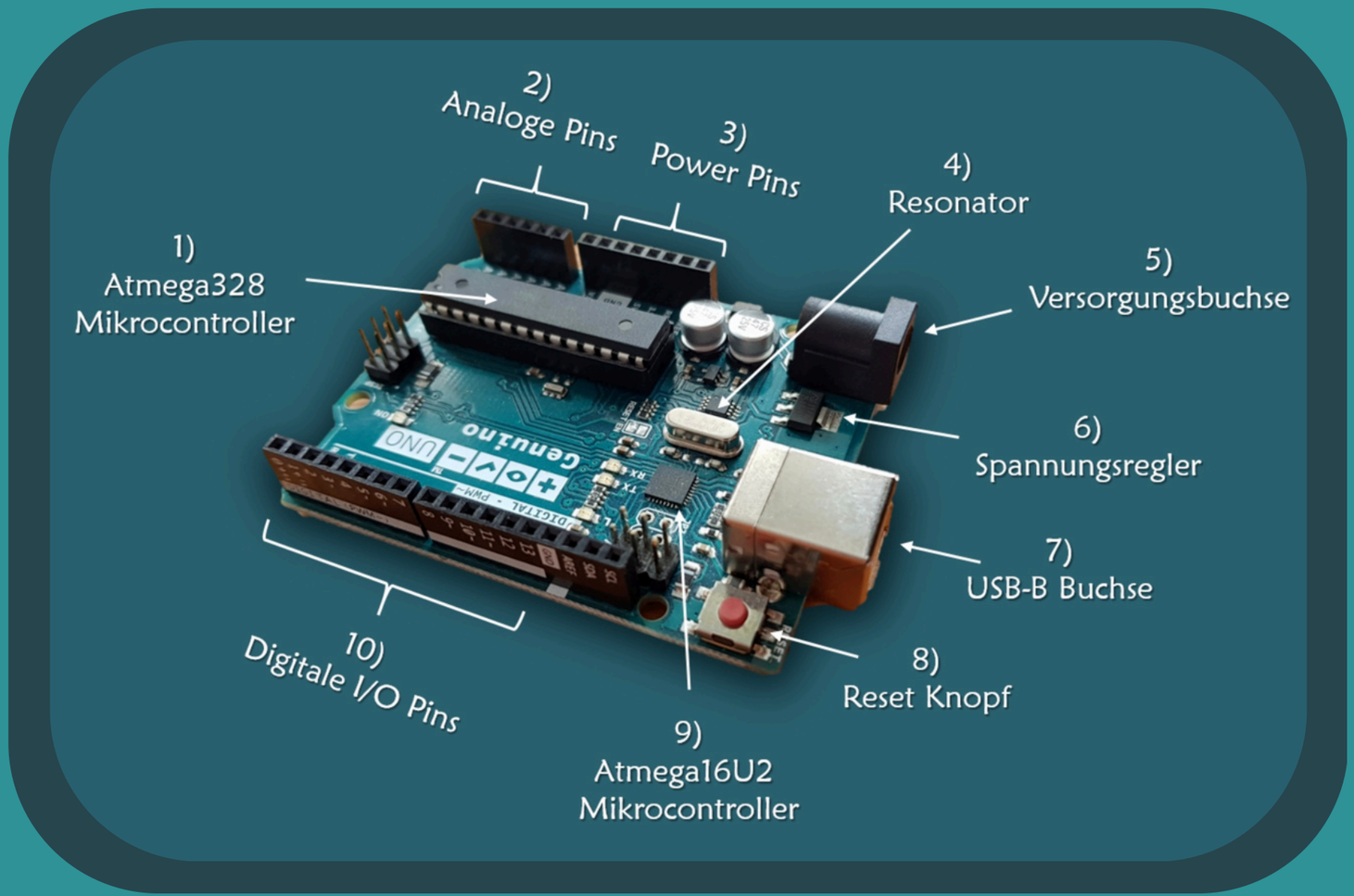
Diese Anschlüsse können feine Unterschiede messen, zum Beispiel wie warm es ist oder wie hell eine Lampe leuchtet.

3. Power Pins

Diese Anschlüsse liefern Strom, damit Sensoren, Motoren oder Lichter funktionieren.

4. Resonator

Eine Art „Metronom", das dem Arduino hilft, alles im richtigen Tempo zu machen.



5. Versorgungsbuchse

Ermöglicht die Stromversorgung des Arduino über ein externes Netzteil oder eine Batterie.

6. Spannungsregler

Wandelt die Eingangsspannung (z. B. 9V) in eine für den Arduino sichere Spannung (z. B. 5V) um.

7. USB-B Buchse

Dient zum Programmieren des Arduino und zur Stromversorgung über den Computer.

8. Reset Knopf

Startet das laufende Programm neu, ohne den Arduino vom Strom zu trennen.

9. Mikrocontroller 16U2

Ein zusätzlicher Mikrocontroller, der die Kommunikation zwischen dem PC und dem Atmega328 steuert.

10. Digitale I/O Pins

Pins für digitale Ein- und Ausgänge (z. B. zum Steuern von LEDs oder Erfassen von Knopfdruck-Signalen).

Arduino IDE

Die Arduino IDE ist ein Programm, mit dem man den Arduino steuern kann. Hier schreibt man den Code, prüft ihn auf Fehler und lädt ihn auf den Arduino. Außerdem gibt es Werkzeuge wie den seriellen Monitor, um Daten vom Arduino anzuzeigen.

1. Check Button

Prüft den Code auf Fehler, bevor er auf den Arduino geladen wird.

2. Upload Button

Schickt den Code an den Arduino, damit er ausgeführt wird.

3. Setup Funktion

Hier werden einmalig die Grundeinstellungen gemacht, z. B. welche Pins als Eingänge oder Ausgänge genutzt werden.

4. Loop Funktion

Hier steht, was der Arduino immer wieder wiederholen soll, z. B. eine LED blinken lassen.

5. Library Funktionen

Vorgefertigte Code-Bausteine, die helfen, bestimmte Dinge einfacher zu machen, z. B. Sensoren auslesen.

6. Serieller Monitor

Ein Fenster in der IDE, das zeigt, was der Arduino sendet, z. B. Messwerte von Sensoren.

7. Quellcode

Der geschriebene Code, der dem Arduino sagt, was er tun soll.

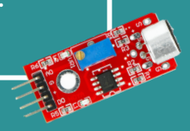
8. Terminal

Ein Textfenster für technische Eingaben und Ausgaben, ähnlich wie die Kommandozeile eines Computers.

Arduino Komponente

1. Schall-Sensor

Erkennt Geräusche und misst die Lautstärke.

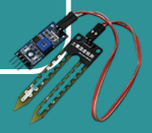


2. LED

Eine kleine Lampe, die mit wenig Strom leuchtet.

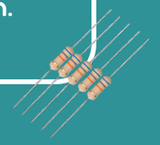
3. Power Pins

Misst, wie feucht die Erde ist, z. B. für Pflanzen.



4. Resistor

Begrenzt den Stromfluss, um Bauteile zu schützen.



5. Servo Motor

Ein kleiner Motor, der sich genau in eine bestimmte Position drehen kann.

6. USB-A-Kabel

Ein Kabel, um den Arduino mit dem Computer zu verbinden und mit Strom zu versorgen.

7. Button

Ein Knopf, der ein Signal sendet, wenn man ihn drückt.

8. Jumper Kabel

Kleine Kabel, um Bauteile miteinander zu verbinden.

9. Feuchtigkeitssensor

Misst die Luftfeutchtigkeit.

10. Batterie

Eine Stromquelle für den Arduino und andere Bauteile.

11. CO_2 — Sensor

Misst den Kohlendioxidgehalt in der Luft.

12. Temperatur Sensor

Misst die Temperatur.

13. Kapazitor

Speichert kurzzeitig elektrische Energie.

14. Buzzer

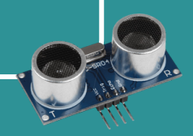
Gibt Töne oder Piepsgeräusche von sich.

15. Licht-Sensor

Erkennt, wie hell oder dunkel es ist.

16. Ultraschall- Sensor

Misst Entfernungen mit Schallwellen, ähnlich wie Fledermäuse.



Morse Code LED

Schaffst du es Hilfe zu holen ?

Bauteile

- Arduino Uno
- 1x LED
- 1x Widerstand (220 Ω)
- Jumper-Kabel

Schritte

1) Laptop mit Arduino verbinden

2) Code Hochladen

3) Komponente verbinden

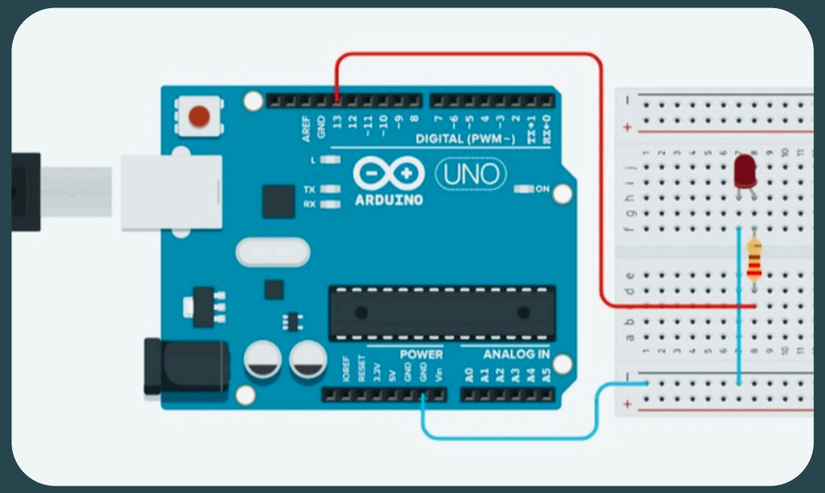
4) Stromquelle anschließen

5) Testen

Ziel

Eine LED blinkt „SOS“ in Morsecode.

Schaltplan



Quellcode

```
const int ledPin = 13;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // Buchstaben "S" (· · ·)
  blinkDot(); blinkDot(); blinkDot();
  // Buchstabe "O" (- - -)
  blinkDash(); blinkDash(); blinkDash();
  // Buchstaben "S" (· · ·)
  blinkDot(); blinkDot(); blinkDot();

  delay(3000); // Pause vor dem erneuten Durchlauf
}

void blinkDot() {
  digitalWrite(ledPin, HIGH);
  delay(200);                  // LED leuchtet kurz
  digitalWrite(ledPin, LOW);
  delay(200);                  // Kurze Pause
}

void blinkDash() {
  digitalWrite(ledPin, HIGH);
  delay(600);                  // LED leuchtet länger
  digitalWrite(ledPin, LOW);
  delay(200);                  // Kurze Pause
}
```

Pflanzendedektiv

Welche Pflanze braucht Wasser ?

Bauteile

- Arduino Uno
- Bodenfeuchtigkeitssensor
- 1x Widerstand (220 Ω)
- Jumper-Kabel

Schritte

1) Laptop mit Arduino verbinden

2) Code Hochladen

3) Komponente verbinden

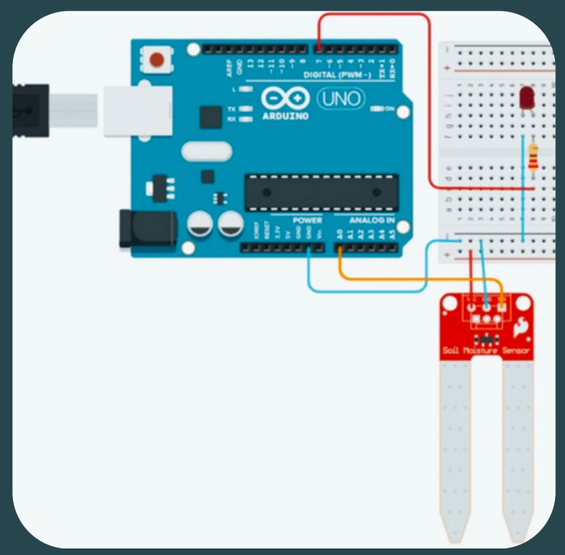
4) Stromquelle anschließen

5) Testen

Ziel

Misst die Bodenfeuchtigkeit. Bei Wassermangel leuchtet eine entsprechende LED,

Schaltplan



Quellcode

```
const int moisturePin = A0;    // Analoger Eingang
const int ledPin = 7;          // LED-Ausgang

void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);          // Serieller Monitor
}

void loop() {
  int moistureValue = analogRead(moisturePin);
  Serial.print("Feuchtigkeitswert: ");
  Serial.println(moistureValue);

  // Schwellenwert anpassen (z.B. 400):
  if (moistureValue < 400) {
    // Erde ist trocken
    digitalWrite(ledPin, HIGH);
  } else {
    // Erde ist feucht genug
    digitalWrite(ledPin, LOW);
  }

  delay(1000); // 1 Sekunde warten
```

Snack Hero

Vertilgen deine Geschwister immer dein Lieblingssnack ?

Bauteile

- Arduino Uno
- Bewegungssensor
- Buzzer
- Jumper-Kabel

Schritte

1) Laptop mit Arduino verbinden

2) Code Hochladen

3) Komponente verbinden

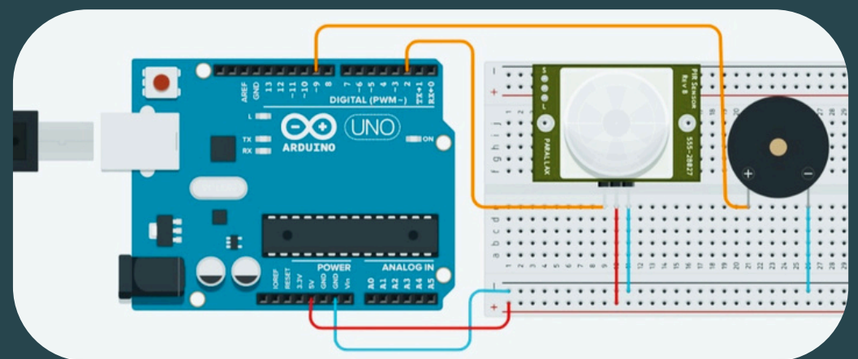
4) Stromquelle anschließen

5) Testen

Ziel

Wenn Bewegung erkannt wird, piept ein Buzzer.

Schaltplan



Quellcode

```
const int pirPin = 2;
const int buzzerPin = 9;

void setup() {
  pinMode(pirPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int motion = digitalRead(pirPin);
  if (motion == HIGH) {
    // Bewegung erkannt
    tone(buzzerPin, 1000);     // Piepton
    Serial.println("Bewegung erkannt!");
    delay(1000);               // 1 Sekunde piep
    noTone(buzzerPin);
  }
}
```

ROBO ARM

Schonmal einen Roboter Arm gewünscht ?

Bauteile

- Arduino Uno
- Servo SG90
- Jumper-Kabel

Schritte

1) Laptop mit Arduino verbinden

2) Code Hochladen

3) Komponente verbinden

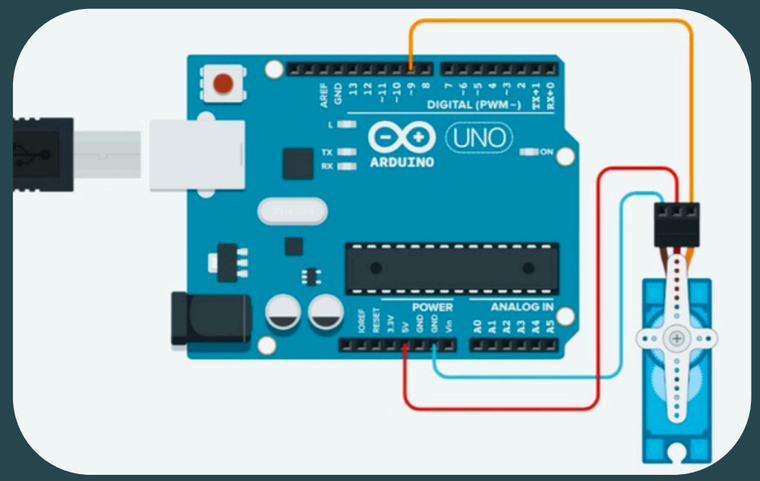
4) Stromquelle anschließen

5) Testen

Ziel

Ein Servo, der sich in verschiedenen Winkeln bewegen kann.

Schaltplan



Quellcode

```ino
#include <Servo.h>

Servo myServo;

void setup() {
  myServo.attach(9);  // Signal-Pin an D9
}

void loop() {
  // Von 0° auf 180°
  for (int angle = 0; angle <= 180; angle++) {
    myServo.write(angle);
    delay(15);  // kleiner Wartewert für flüssige Bewegung
  }
  // Zurück von 180° auf 0°
  for (int angle = 180; angle >= 0; angle--) {
    myServo.write(angle);
    delay(15);
  }
}
```

Bat-Eye

Unsichtbares Lineal gefällig ?

Bauteile

- Arduino Uno
- Ultraschallsensor
- Jumper-Kabel

Schritte

1) Laptop mit Arduino verbinden

2) Code Hochladen

3) Komponente verbinden

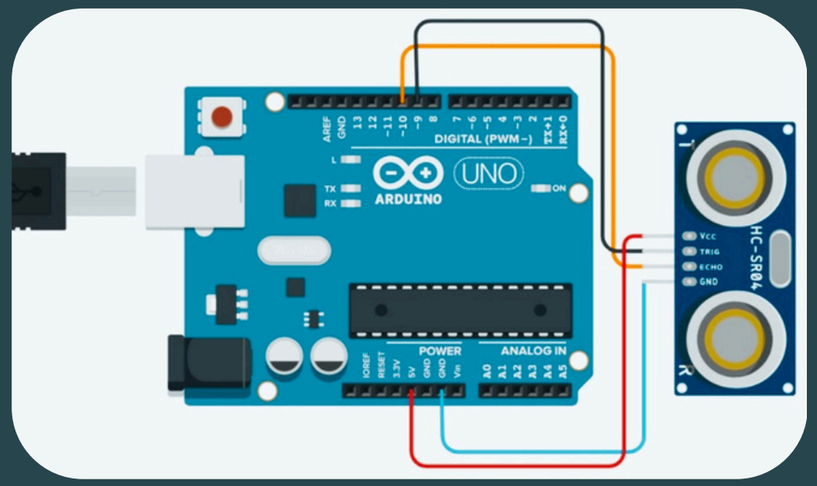
4) Stromquelle anschließen

5) Testen

Ziel

Misst Abstände mithilfe von Schallwellen. Zeigt den Wert im seriellen Monitor an.

Schaltplan



Quellcode

```
const int trigPin = 9;
const int echoPin = 10;

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  // Signal senden
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Echo-Zeit messen
  long duration = pulseIn(echoPin, HIGH);

  // Entfernung in cm berechnen
  long distanceCm = duration * 0.034 / 2;

  Serial.print("Entfernung: ");
  Serial.print(distanceCm);
  Serial.println(" cm");

  delay(500); // halbe Sekunde warten
}
```