

Arduino	Modul 4
	Zustandsmodellierung

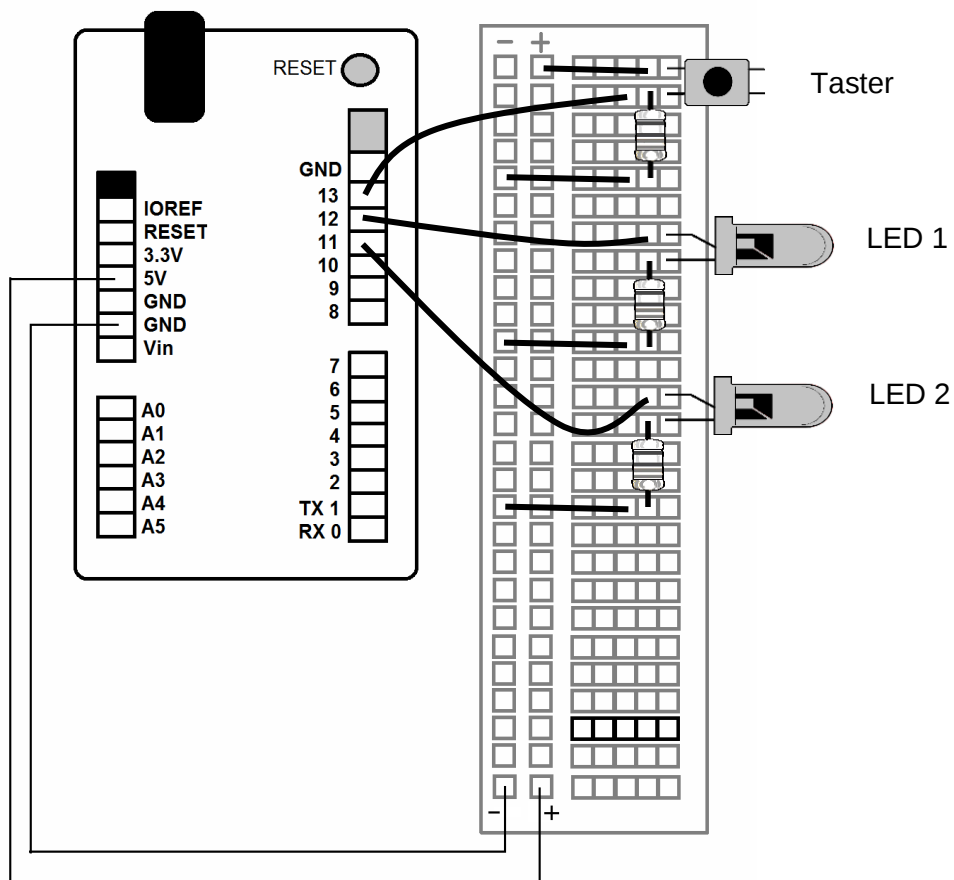
MOTIVATION

Die Klimaanlage eines Autos ist mit einem Knopf in 3 Stufen regelbar. Drückt man auf den Knopf, dann schaltet sich die Anlage nach folgendem Schema weiter:

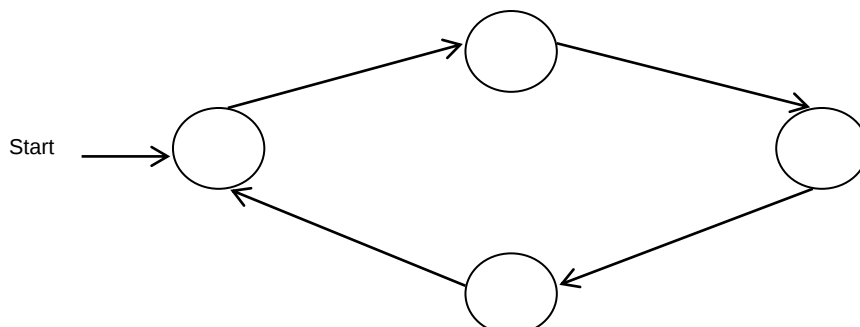
Aus --> Stufe 1 --> Stufe 2 --> Stufe 1 --> Aus --> Stufe 1 --> Stufe 2 --> Stufe 1 --> usw.

Zur Anzeige sind 2 LEDs vorhanden:

Aus >>> keine leuchtet
Stufe 1 >>> 1 LED leuchtet
Stufe 2 >>> beide leuchten



Ergänze die Modellzeichnung, indem du die Kreise mit den jeweiligen Zuständen beschriftest, die Pfeile mit der jeweiligen Eingabe (Drücken der Taste) und die Kommentarfelder mit sprechenden Zustandsbezeichnern.



Arduino	Modul 4
	Zustandsmodellierung

Vervollständige den Code für den Programmfluss:

```
int z = 1;
int Taster = 0;

void setup() {
  pinMode (13, INPUT);
  pinMode (12, OUTPUT);
  pinMode (11, OUTPUT);
}

void loop() {
  Taster = digitalRead(13);
  if (Taster == HIGH){

    switch (z) {

      case 1: {
        z = 2;
        digitalWrite(11, HIGH);
        digitalWrite(12, LOW);
        delay(300);    //--> damit der Taster nicht durchschaltet
        break;
      }

      case 2: {      . . .      }

      case 3: {      . . .      }

      case 4: {      . . .      }
```

ÜBERSICHT

Taster

```
int Taster = 0;

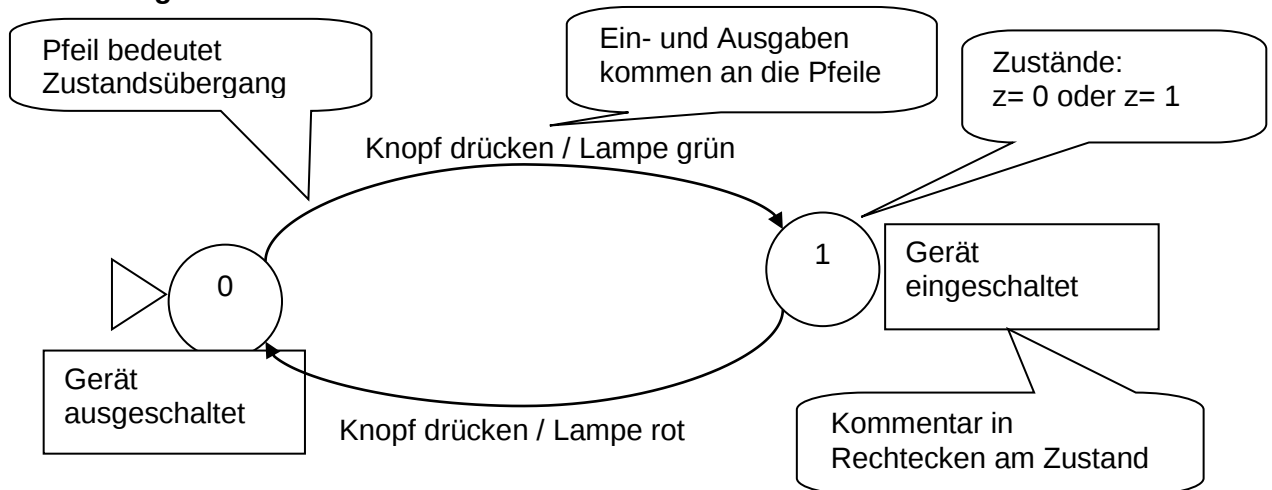
void setup() {
  pinMode(13, INPUT);
  pinMode(11, OUTPUT);
}

void loop () {
  Taster = digitalRead(13);

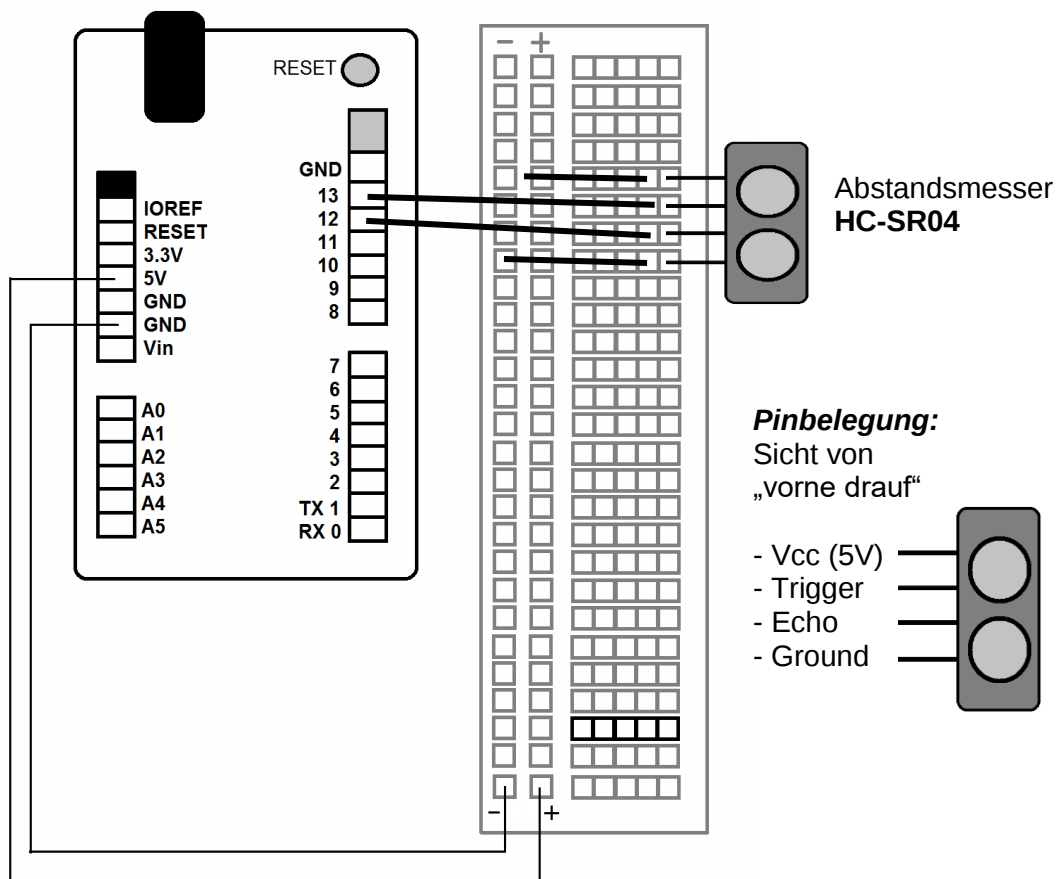
  if ((Taster == HIGH){
    digitalWrite(11, HIGH);
  } else {
    digitalWrite(11,LOW);
  }
  delay(100);
}
```

Arduino	Modul 4
	Zustandsmodellierung

Zustandsdiagramm



Entfernungsmesser HC-SR04



Der Entfernungsmesser sendet über einen Lautsprecher ein Ultraschallsignal ab, das von einem Gegenstand reflektiert wird. Ein Mikro nimmt die Reflexion auf und misst die Zeit.

Trigger (hier am Pin 13): Befehlstrigger, eine Messung durchzuführen
Eche (hier am Pin 12): Ergebnis der Messung

Arduino	Modul 4
	Zustandsmodellierung

Der „Trigger“ sendet bei fallender Taktflanke ein Signal (8 mal hintereinander ein 40KHz Signal). Bei Signalende liegt das Ergebnis bei „Echo“ an. Das Ergebnis (die Entfernung) kann z.B. über den Monitor ausgegeben werden.

Code

```
long Dauer, Abstand;

void setup() {
  Serial.begin (9600);
  pinMode(13, OUTPUT);    // Trigger
  pinMode(12, INPUT);     // Echo
}

void loop() {
  digitalWrite(13, LOW);   // kurzes LOW-Signal, damit
  delayMicroseconds(2);    // das HIGH-Signal sauber ist (lt. Anleitung)
  digitalWrite(13, HIGH);  // hohes Signal, damit eine
  delayMicroseconds(20);   // fallende Taktflanke
  digitalWrite(13, LOW);   // hier erzeugt wird --> die Messung beginnt

  Dauer = pulseIn(12, HIGH); // Das Ergebnis wird ermittelt
  Abstand = Dauer * 34.3 / 2000 ; // Abstand in cm vgl. Beispielrechnung

  // Zahlenwerte im seriellen Monitor ausgeben
  if (Abstand >= 200 || Abstand <= 0){
    Serial.println("ausserhalb der Reichweite");
  } else {
    Serial.print(Abstand);
    Serial.println(" cm");
  }

  delay(250);
}
```

Beispielrechnung zur Ermittlung der Entfernung:

Bei der Schallsendung wird eine Dauer von 500 Mikrosekunden gemessen.

1.) 500 Mikrosekunden = 0,5 Millisekunden (ms)

2.) Der Schall hat eine Geschwindigkeit von $343 \frac{m}{sec} = 343 \cdot \frac{100 cm}{1000 ms} = 34,3 \frac{cm}{ms}$

3.) $34,3 \frac{cm}{ms} \cdot 0,5 ms = 17,15 cm$ --> Da der Schall den doppelten Weg zurückgelegt hat, muss das Ergebnis noch durch 2 geteilt werden: 8,575 cm.

Damit leitet sich die Formel her:

Messergebnis in Mikrosekunden / 1000 * 34,3 : 2 (cm) =

Messergebnis * 34,3 / 2000 (cm)

Arduino	Modul 4
	Zustandsmodellierung

TRAINIEREN

Aufgabe 1 Quizverriegelungsschaltung

Zwei Tasten und zwei LEDs. Wer als erster auf den Knopf drückt, dessen LED leuchtet. Der andere kann dann nicht mehr seine LED zum Leuchten bringen.

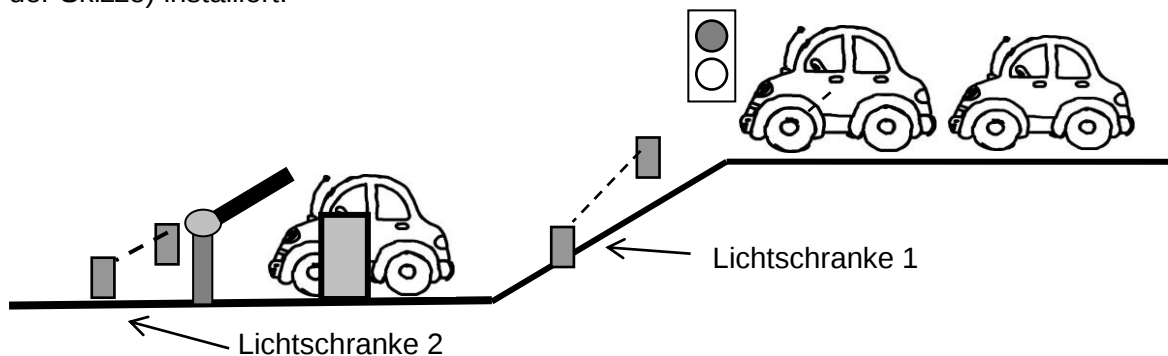
- Erstelle ein Zustandsdiagramm.
- Setze als Schaltung und Programm um.

ANWENDEN

Aufgabe 2

Die schmale, unübersichtliche und steile Einfahrt zu einem Parkhaus ist mit einer Einfahrtsschranke gesichert. Vor der Einfahrt ist die Schranke geschlossen. Auf Knopfdruck wird ein Parkticket ausgegeben, die Schranke öffnet sich und das Auto kann ins Parkhaus fahren. Danach schließt sich die Schranke wieder.

Da es in der letzten Zeit immer wieder zu Auffahrunfällen auf vor der geschlossenen Schranke stehenden Autos gegeben hat, wurde vor dem Zufahrtsgefälle eine Ampel installiert. Sie schaltet auf Rot, sobald ein Auto an die Schranke herankommt und anhält. Erst wenn das Ticket gezogen, das Auto durchgefahren ist und die Schranke wieder geschlossen wurde, schaltet die Ampel wieder auf grün. Das nächste Auto kann an die Schranke herankommen. Zur Steuerung der Ampel und der Schranke wurden zwei Lichtschranken (wie in der Skizze) installiert.



Setzte als Simulation um.

Aufgabe 3 Ampel mit Fußgängerschaltung

- Baue eine Ampelschaltung für eine Autoampel und
- erweitere die Schaltung um einen Fußgängerknopf, bei dem auf Knopfdruck die Autoampel auf Rot und die Fußgängerampel auf Grün schaltet.