

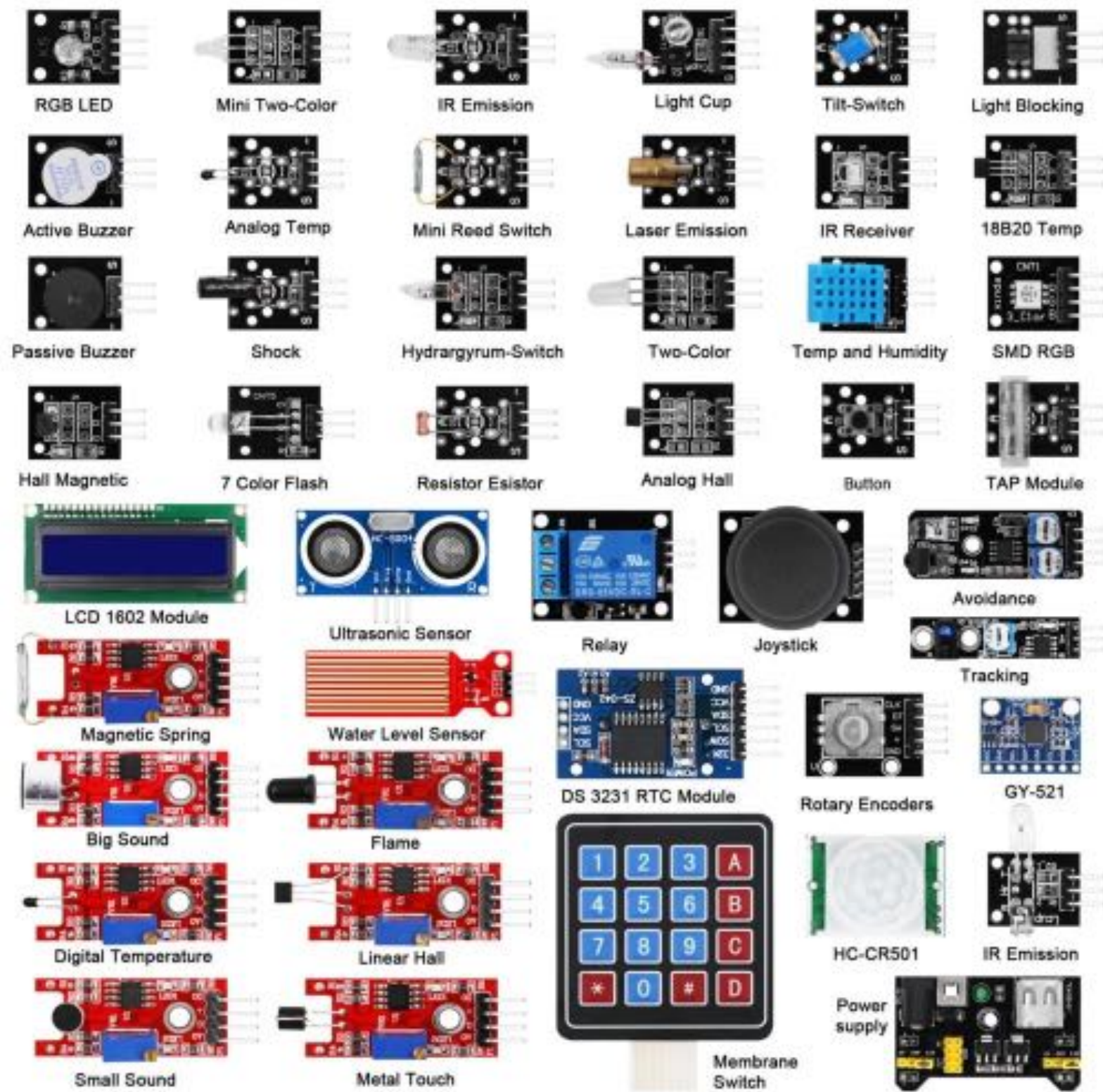
# Robotik Lab - Sensoren im Einsatz mit Arduino

Ivan Iovine - 2. Sensoren im Einsatz mit Arduino - 15.06.22

# Themen Vorlesung 8.6.22

- Was sind eigentlich Sensoren?
- Sensoren vs. Aktuatoren
- Physical Computing Ecosystem
- Sensordaten auslesen und als Eingabe verwenden + Conditional Statements
- Verwendung eines Geräuscherkennungssensors mit Arduino
- Abstand messen mit einem HC-SR04 Ultraschallsensor mit Arduino

# Was sind eigentlich Sensoren?

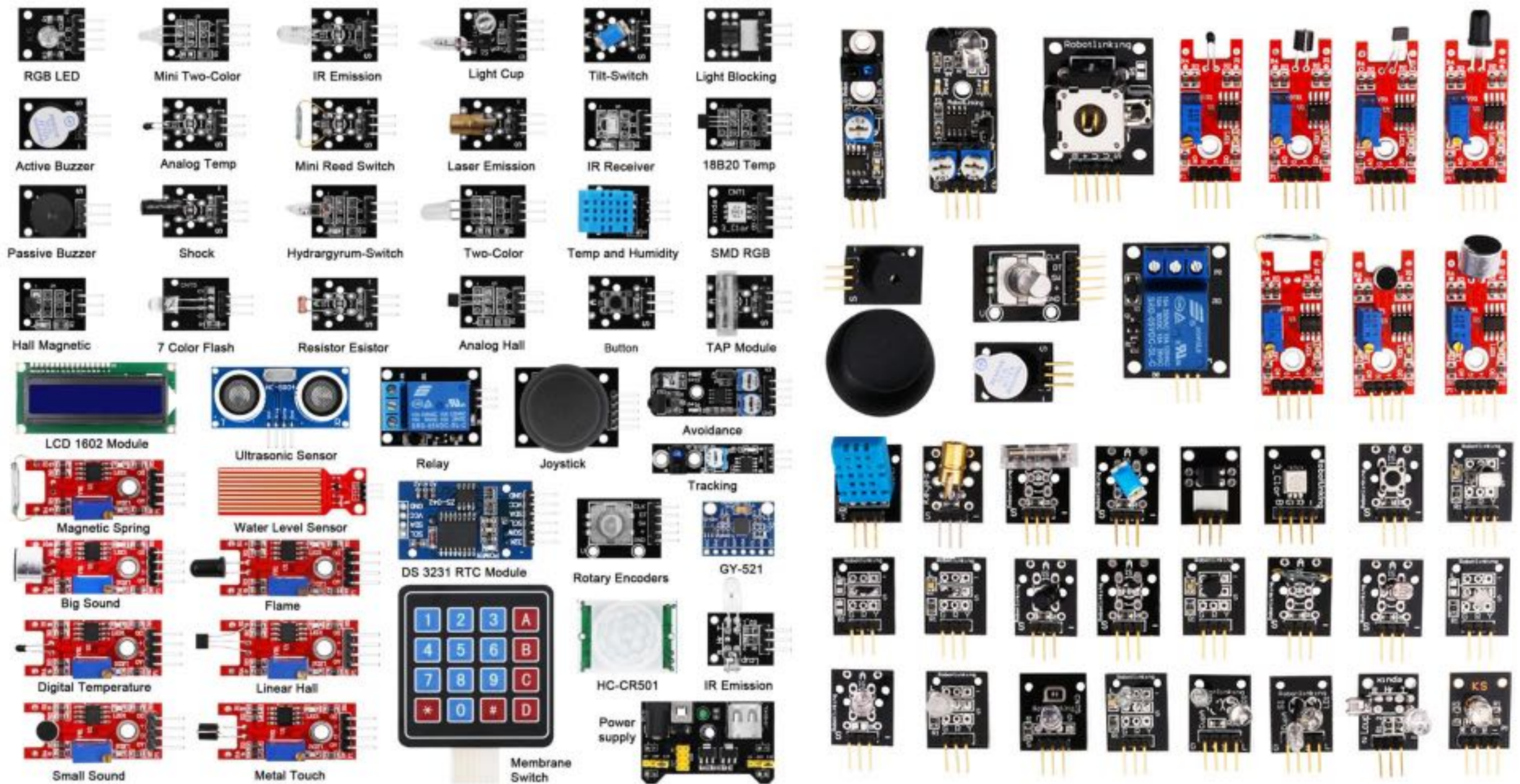


## Was sind eigentlich Sensoren?

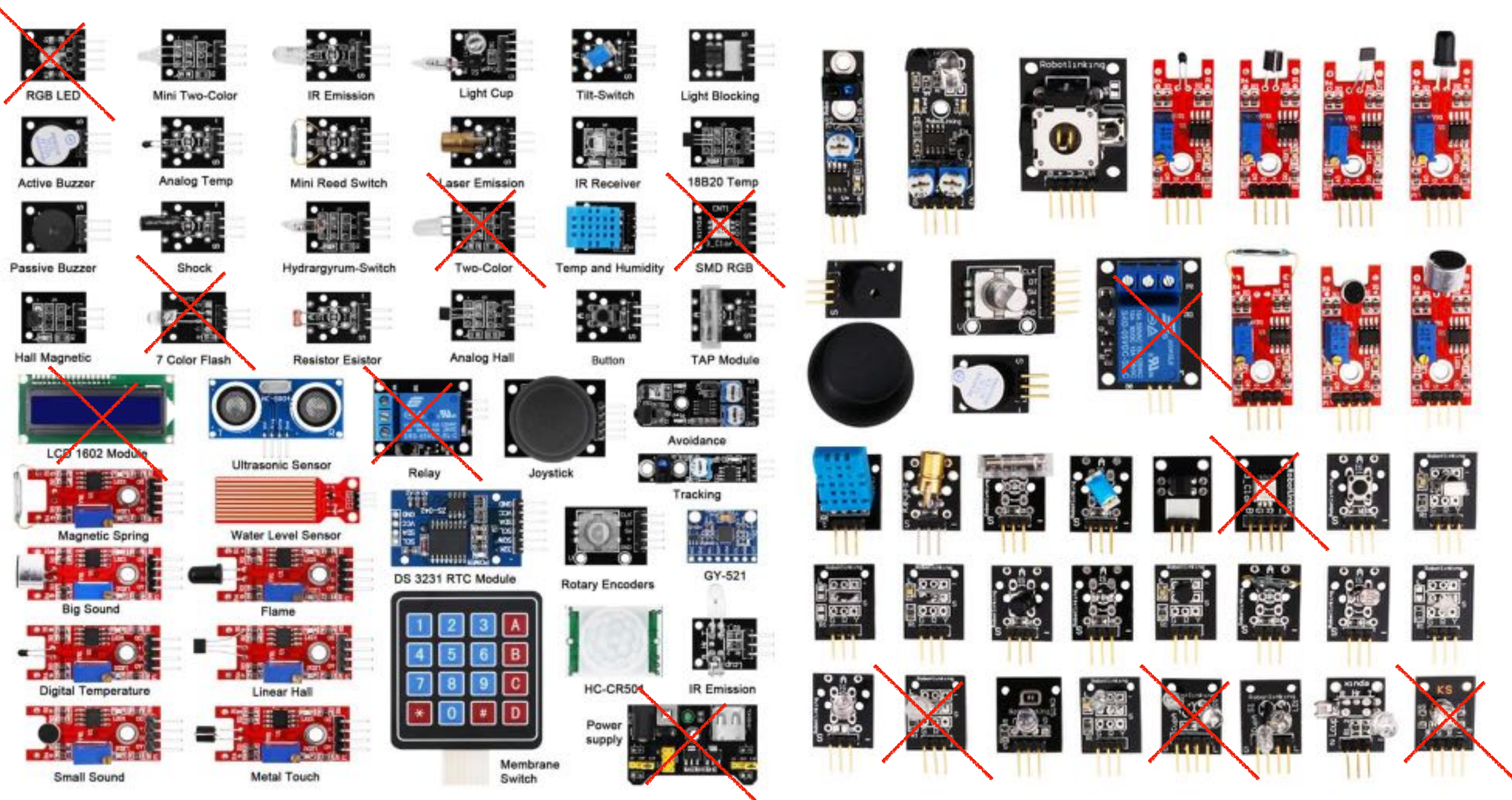
Sensoren sind die Augen und Fühler eines Arduino-Boards und ermöglichen es, verschiedene Messwerte rund um den Menschen und seine Umgebung aufzunehmen. Die von den Sensoren aufgezeichneten Messwerte können auf dem Arduino-Board verarbeitet und angezeigt werden. Sie werden oft als Antriebsquelle für die Steuerung von Aktuatoren wie zB. Motoren und Leuchten verwendet.

Inspiziert von der Einleitung des Buches "Sensoren im Einsatz mit Arduino" von Thomas Brüllman



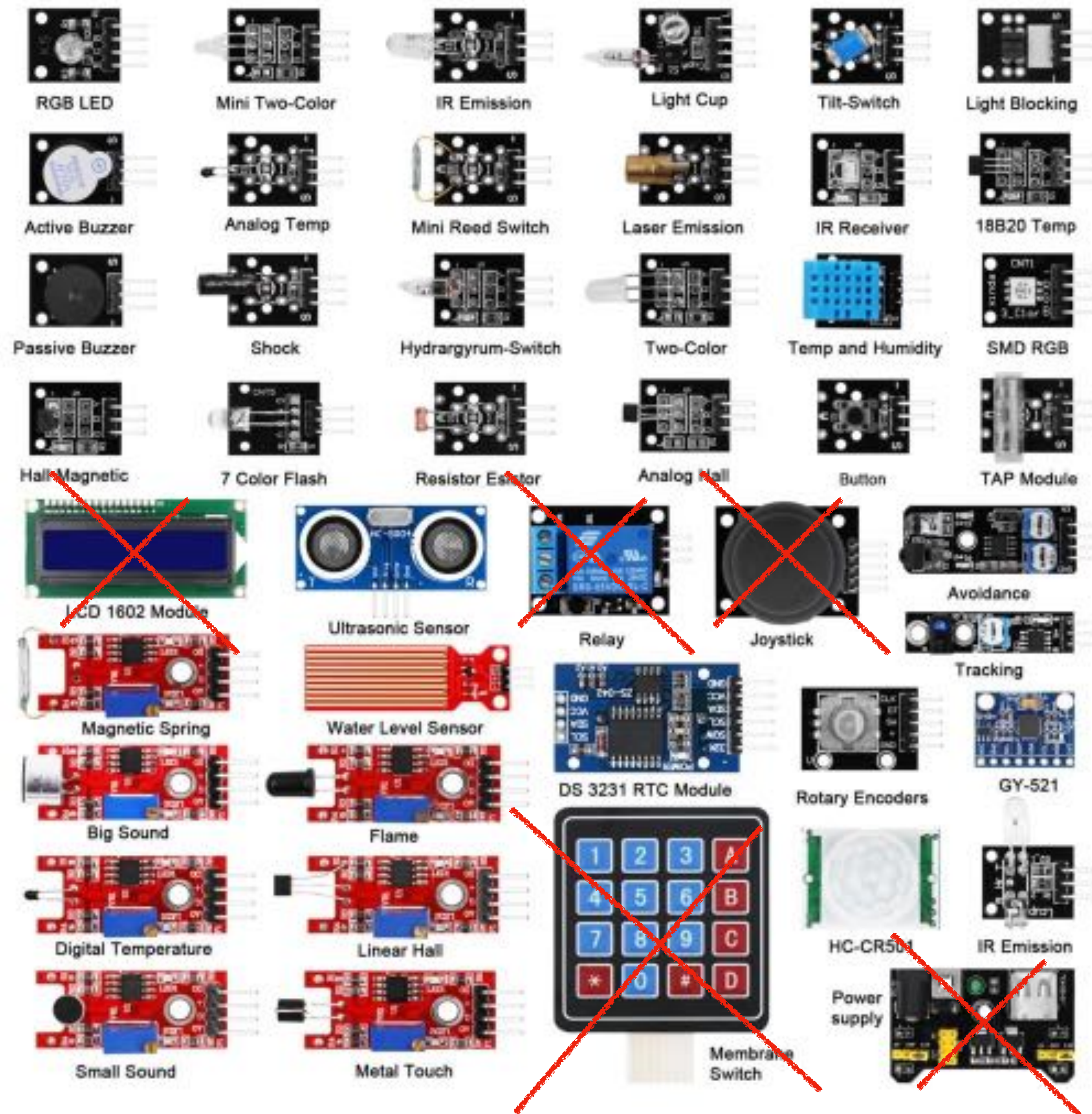








# Sensoren



# Aktuatoren





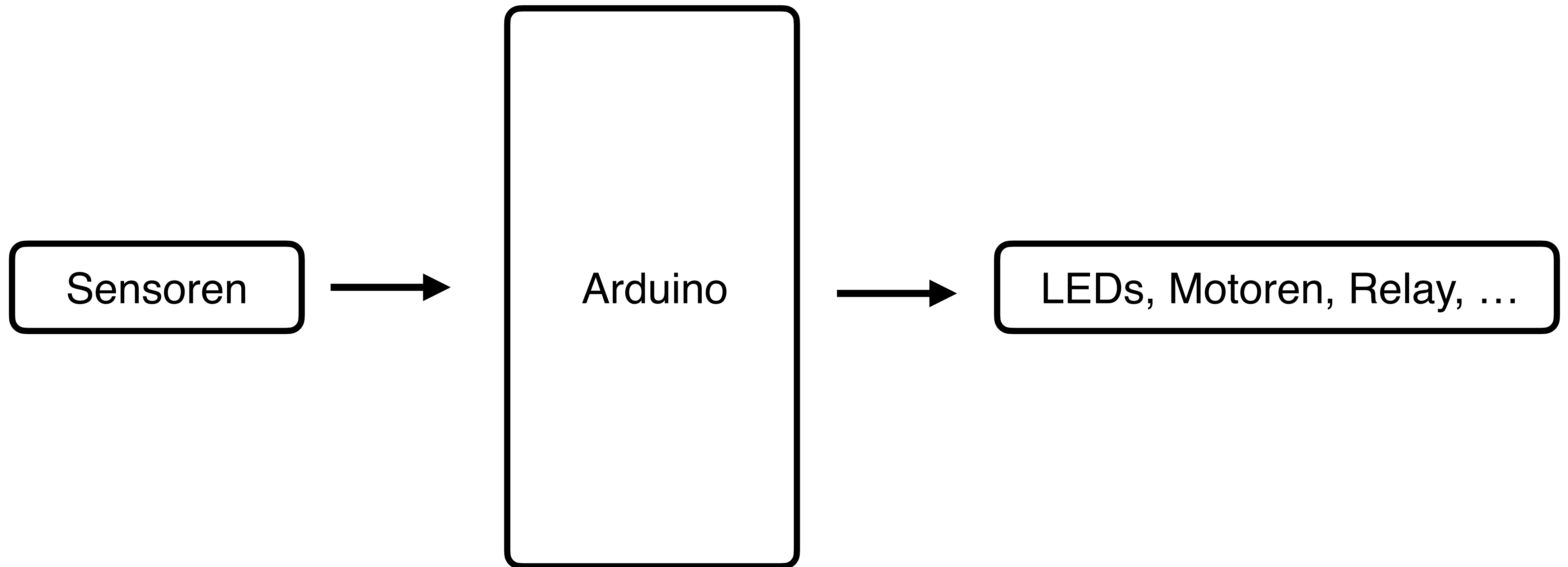
# Physical Computing Ecosystem



**Messung**

**Logik**

**Handlung**

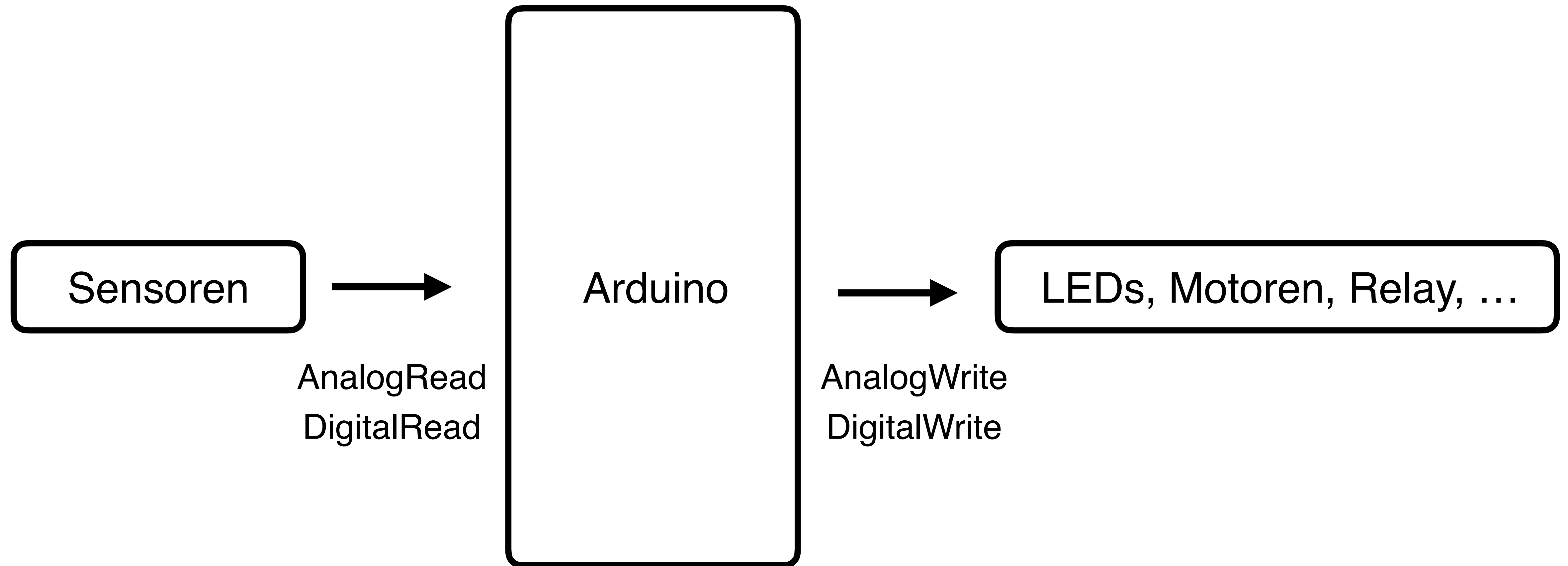




## Messung

## Logik

## Handlung

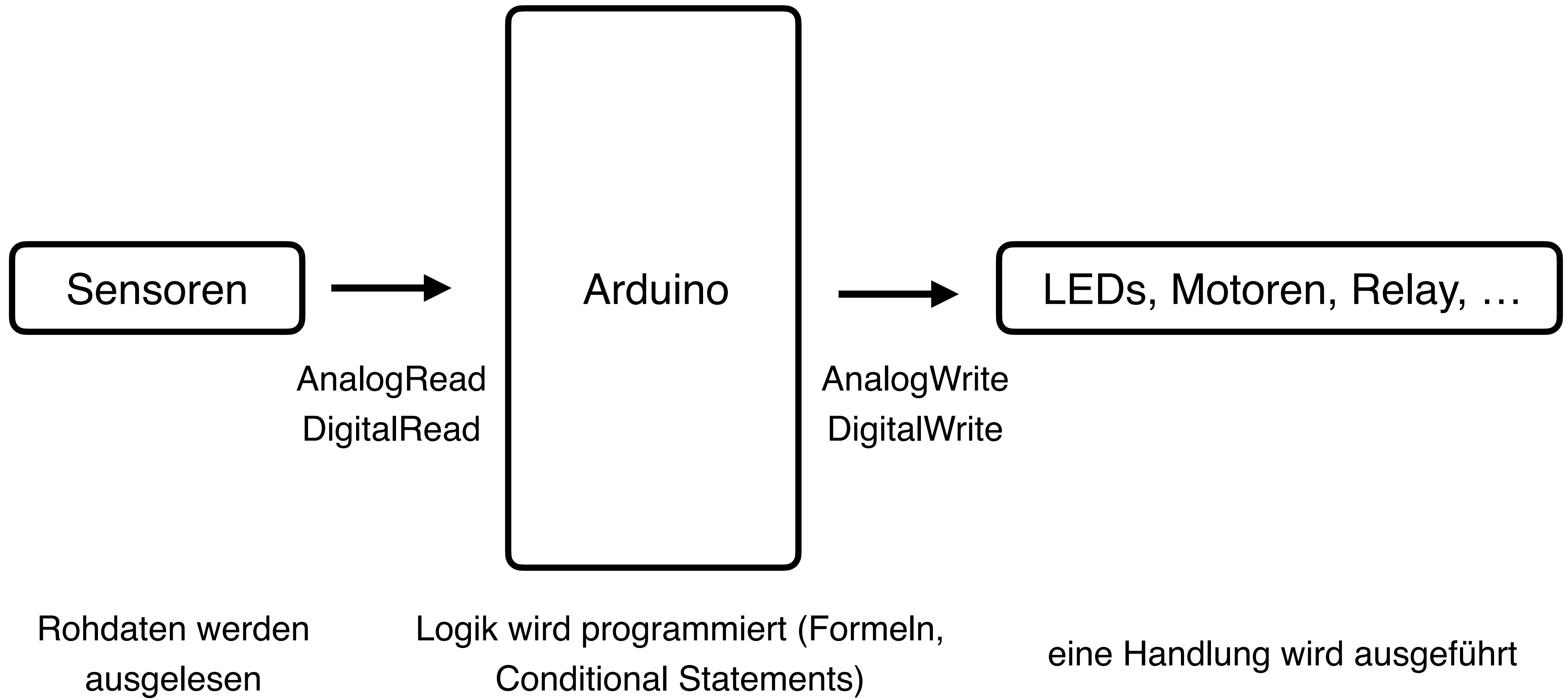




## Messung

## Logik

## Handlung





## Messung



Sensoren

AnalogRead  
DigitalRead

Rohdaten werden  
ausgelesen

## Logik



Arduino

AnalogWrite  
DigitalWrite

Logik wird programmiert (Formeln,  
Conditional Statements)

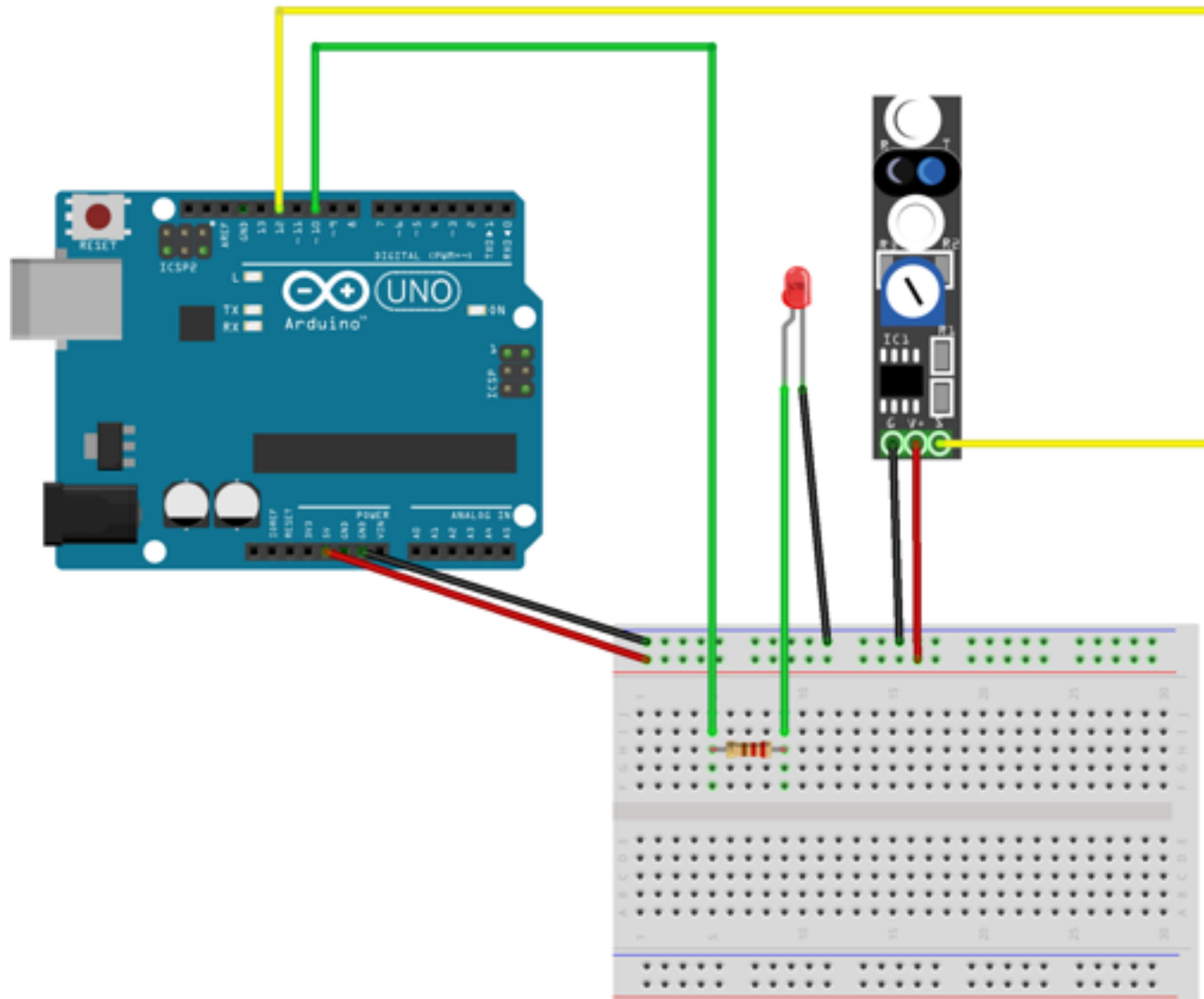
## Handlung



LEDs, Motoren, Relay, ...

eine Handlung wird ausgeführt







```
if (condition1) {  
    // do Thing A  
}  
else if (condition2) {  
    // do Thing B  
}  
else {  
    // do Thing C  
}
```

## Conditional Statement: if / else

Im **if** wird ein Codeblock angegeben, der ausgeführt werden soll, wenn eine bestimmte Bedingung **true** ist.

Im **else** wird ein Codeblock angegeben, der ausgeführt werden soll, wenn die gleiche Bedingung **false** ist.

Im **else if** wird eine neue Bedingung angegeben, die getestet wird, wenn die erste Bedingung **false** ist.



## Syntax

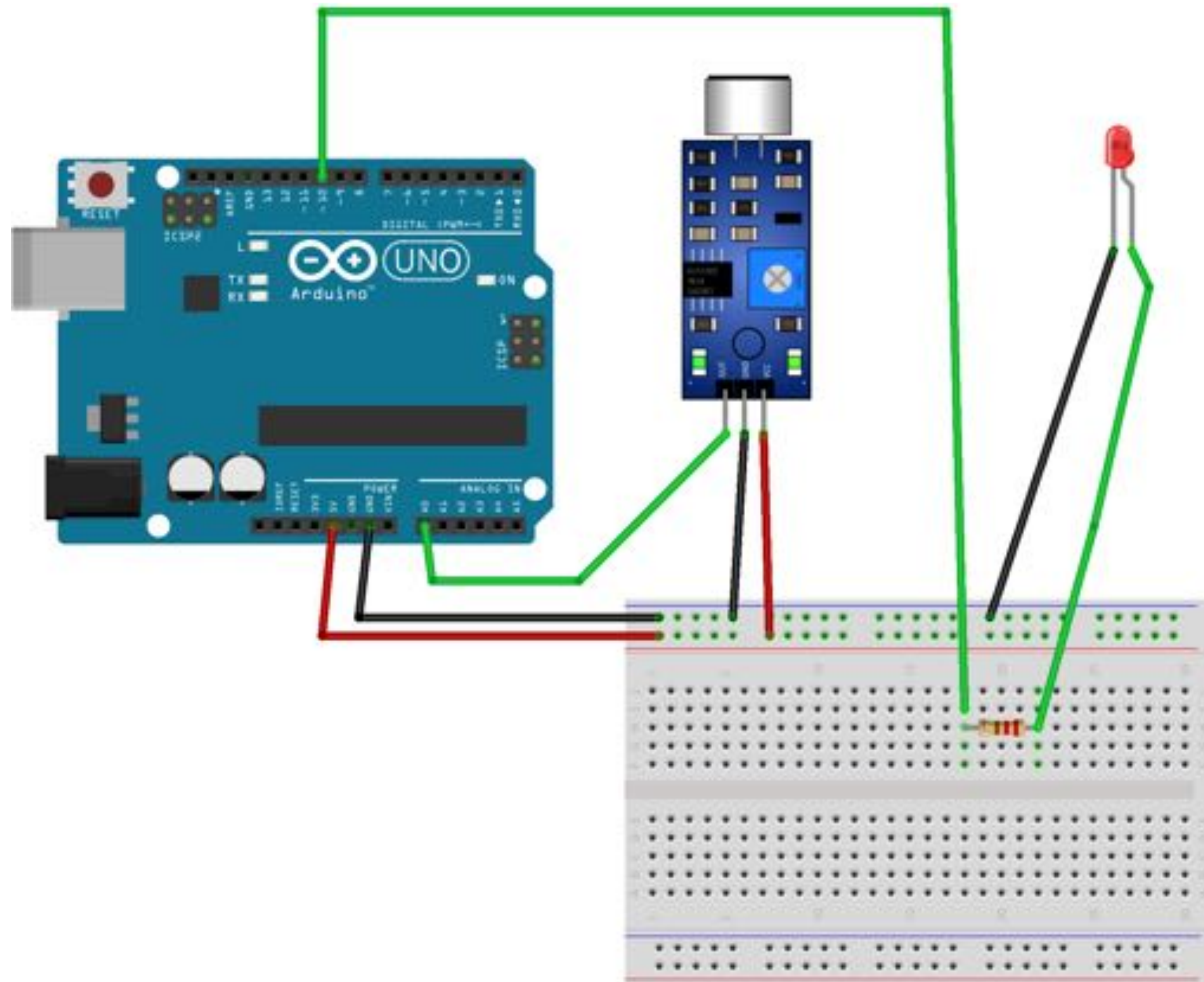
```
if (condition1) {  
    // do Thing A  
}  
else if (condition2) {  
    // do Thing B  
}  
else {  
    // do Thing C  
}
```

## Example Code

Below is an extract from a code for temperature sensor system

```
if (temperature >= 70) {  
    // Danger! Shut down the system.  
}  
else if (temperature >= 60) { // 60 <= temperature < 70  
    // Warning! User attention required.  
}  
else { // temperature < 60  
    // Safe! Continue usual tasks.  
}
```







```
for (initialization; condition; increment) {  
    // statement(s);  
}
```

## Conditional Statement: for cycle

Die **for** schleife durchläuft einen Codeblock mehrmals

## Syntax

```
for (initialization; condition; increment) {  
  // statement(s);  
}
```

## Parameters

**initialization:** happens first and exactly once.

**condition:** each time through the loop, condition is tested; if it's **true**, the statement block, and the **increment** is executed, then the **condition** is tested again. When the **condition** becomes **false**, the loop ends.

**increment:** executed each time through the loop when condition is true.

## Example Code

```
// Dim an LED using a PWM pin  
int PWMpin = 10; // LED in series with 470 ohm resistor on pin 10  
  
void setup() {  
  // no setup needed  
}  
  
void loop() {  
  for (int i = 0; i <= 255; i++) {  
    analogWrite(PWMpin, i);  
    delay(10);  
  }  
}
```



