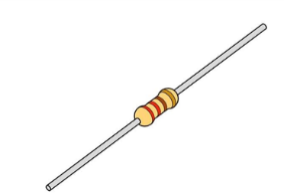
### Aufgabe 1: Wahr oder falsch?

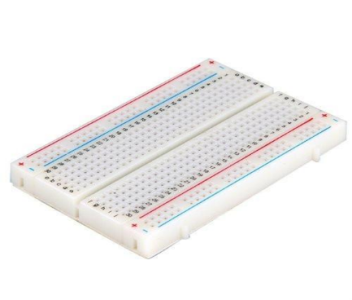
Kreuze an. (5 Punkte)

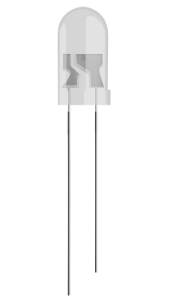
|  |  |
| --- | --- |
| Behauptung | wahr falsch |
| In der Arduino Entwicklungsumgebung ist die setup-Methode für Einstellungen zuständig und wird außerdem immer wieder fortlaufend ausgeführt. | □ X |
| Der Arduino besteht eigentlich aus Hardware und zugehöriger, kostenloser Software (Entwicklungsumgebung). | X □ |
| Die einzige Möglichkeit einen Arduino mit Strom zu versorgen, ist der USB-Anschluss. | □ X |
| In einem Sketch werden oft leserliche Namen verwendet anstelle von Zahlen (z.B. das Wort „HIGH“ steht für den Wert 1). | X □ |
| Der Begriff „Internet of Things“ (Internet der Dinge) meint ein Netzwerk, das mehrere Computer miteinander verbindet für z. B. Videokonferenzen oder Gaming. | □ X |

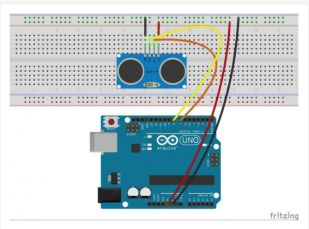
### Aufgabe 2: Begriffe zuordnen

Ordne die einzelnen Bauteile/Sensoren den passenden Begriffen zu und erläutere kurz und präzise die Funktion dieses Bauteils/Sensors. (8 Punkte)

Widerstand: Um die Stromspannung zu regulieren (zu begrenzen) verwenden wir Widerstände. Je höher die Ohm Zahl, desto höher der Widerstand. Die farbigen Ringe auf den Widerständen geben den Wert an, wie stark der elektrische Strom behindert werden soll.

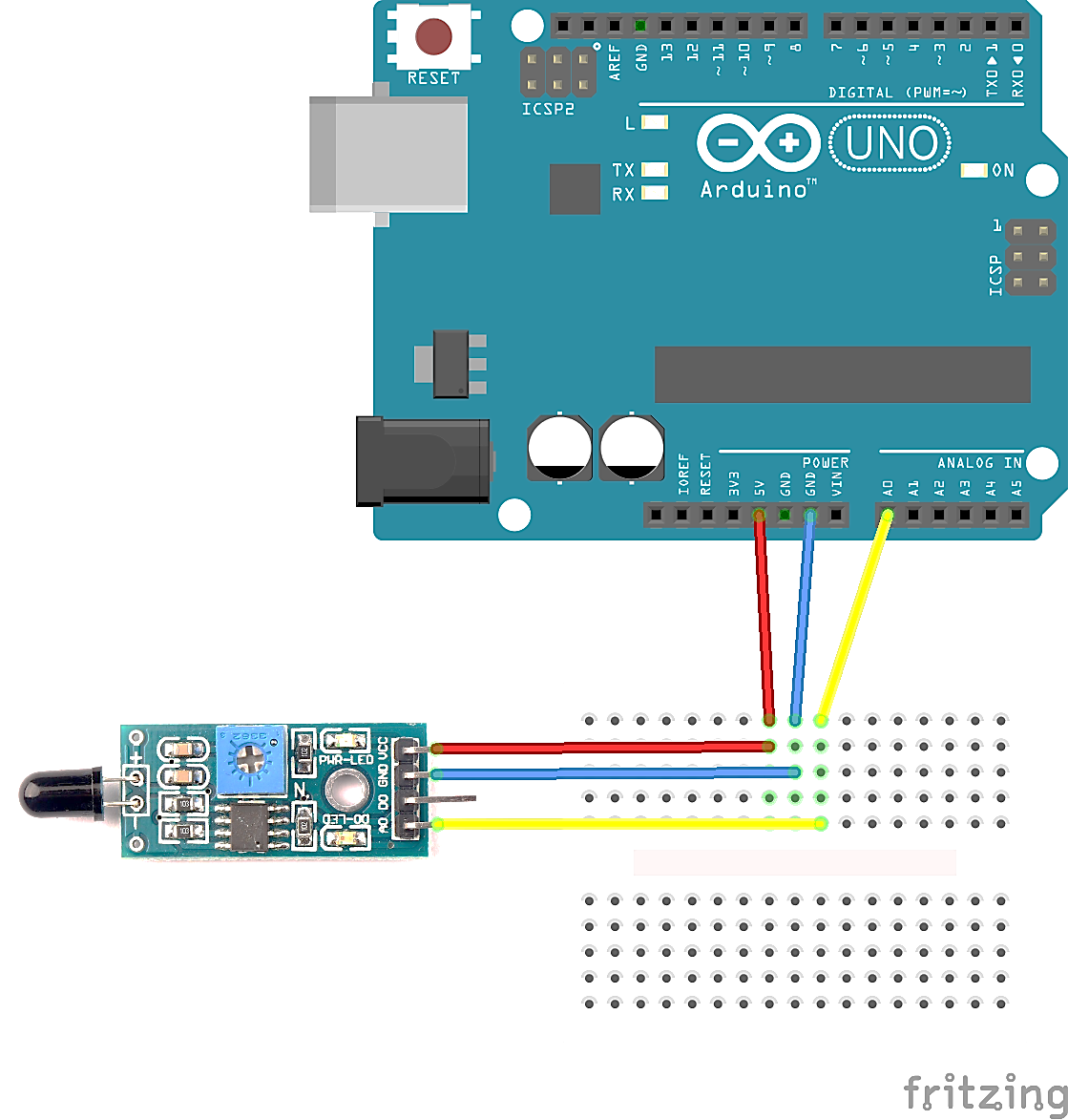
Streckbrett: Ein Steckbrett (Breadboard) ist ein Hilfsmittel, um Schaltungen aufzubauen ohne zu löten. Bauteile werden direkt darauf in die Kontakte gesteckt und mittels Steckbrücken mit dem Arduino verbunden. Die Kontaktreihen werden *Terminal Stripes* genannt und sind immer in Gruppen vertikal (jeweils „a“ - „e“ und „f“ - „j“) oder horizontal (außen + und -) miteinander verbunden.

Leuchtdiode: LED ist die Abkürzung für „Light Emitting Diode“ und wird auch Leuchtdiode genannt. Strom kann hier nur in eine Richtung fließen. Der längere Kontakt ist „**+“** (Anode) und der kürzere ist „**–“** (Kathode).

Ultraschallsensor: Der HC-Sr04 kann die Entfernung zu einem Hindernis messen. Die Pinbelegung ist von links nach rechts: Plus (5V), Trigger, Echo und Minus (GND).

### Aufgabe 3: Schaltung bauen

Ergänze die Grafik rechts zu einer vollständigen Schaltung, indem du einen Widerstand (als farbige Linie) und Kabel (als schwarze Linien) einzeichnest. Verwende den digitalen Pin 3. (5 Punkte)



### Aufgabe 4: Die Ampel springt um!

Erstelle einen **Ampel-Sketch** mit drei LEDs! Schreibe dazu den notwendigen Programmcode für eine Ampel mit vier Phasen, wie es in der Abbildung unten zu sehen ist. Gehe davon aus, dass die entsprechende Schaltung die **Pins 7 (Rot), 8 (Gelb), 9 (Grün)** benutzt. (12 Punkte)

Phase 1: ROT für 10 Sek.  
Phase 2: ROT+GELB für 2 Sek.  
Phase 3: GRÜN für 6 Sek.  
Phase 4: GELB für 2 Sek.

**void setup()   
{**  
 pinMode(7, OUTPUT); // rote LED 1P  
 pinMode(8, OUTPUT); // gelbe LED 1P  
 pinMode(9, OUTPUT); // grüne LED 1P **}**

**void loop()   
{**   
 digitalWrite(7, HIGH);   
 delay(10000); // rot für 10 Sekunden 2P  
  
 digitalWrite(8, HIGH);  
 delay(2000); // rot-orange für 2 Sekunden 2P  
  
 digitalWrite(7,LOW);  
 digitalWrite(8,LOW);  
 digitalWrite(9, HIGH);  
 delay(6000); // grün für 6 Sekunden 2P  
  
 digitalWrite(9,LOW);  
 digitalWrite(8,HIGH);  
 delay(2000); // orange für 2 Sekunden 2P

digitalWrite(8,LOW); // am Ende noch Orange ausstellen, dann von vorne... 1P  
**}**

### Aufgabe 5: Was macht das Programm?

Ergänze in dem folgenden Sketch jeden fehlenden Kommentar, der den jeweiligen Befehl (linke Seite) erklärt. (12 Punkte)

|  |  |
| --- | --- |
|  | Beschreibungen |
| **int trigger=7;** | Der Variablen trigger wird der Wert 7 zugewiesen. (Trigger-Pin des Ultraschallsensors an Pin 7 des Arduino-Boards) |
| int echo=6; | Der Variablen echo wird der Wert 6 zugewiesen. (Echo-Pin des Ultraschallsensors an Pin 6 des Arduino-Boards) |
| long dauer=0; | Unter der Variablen dauer wird die Zeit gespeichert wird, die eine Schallwelle bis zur Reflektion und zurück benötigt. Startwert ist 0. |
| long entfernung=0; | Unter der Variablen entfernung wird die berechnete Entfernung gespeichert . Anstelle von „int“ steht hier vor den beiden Variablen „long“. Das hat den Vorteil, dass eine größere Zahl gespeichert werden kann. Nachteil: Die Variable benötigt mehr Platz im Speicher. |
| void setup() { | Das Setup beginnt. |
| Serial.begin (9600); | Die serielle Kommunikation wird gestartet, damit man sich die Werte auf dem seriellen Monitor ansehen kann. |
| pinMode(trigger, OUTPUT); | Der Trigger-Pin (Pin 7) ist ein Ausgang. |
| pinMode(echo, INPUT); | Der Echo-Pin (Pin 6) ist ein Eingang. |
| } | Das Setup wird beendet. |
| void loop(){ | Das Hauptprogramm beginnt. |
| digitalWrite(trigger, LOW); | Die Spannung wird für kurze Zeit vom Trigger-Pin genommen, damit man beim Senden des Trigger-Signals ein rauschfreies Signal hat. |
| delay(5); |  |
| digitalWrite(trigger, HIGH); | Es wird eine Ultraschallwelle gesendet. |
| delay(10); | Das Programm macht eine Pause von 10 Millisekunden. |
| digitalWrite(trigger, LOW); | Der „Ton“ wird abgeschaltet. |
| dauer = pulseIn(echo, HIGH); | Der Mikrokontroller zählt die Zeit in Millisekunden, bis der Schall zum Ultraschallsensor zurückkehrt. |
| entfernung = (dauer/2) \* 0.03432; | Die ermittelte Zeit wird durch zwei geteilt, weil man nur eine Strecke berechnen möchte = „Hin“. Dann multipliziert man mit der Schallgeschwindigkeit in der Einheit Zentimeter/Mikrosekunde und erhält den Wert in Zentimetern. |
| Serial.print(entfernung); | Der Wert der Entfernung wird auf dem seriellen Monitor ausgegeben, der Cursor bleibt in der Zeile. |
| Serial.println(" cm"); | Hinter dem Wert der Entfernung wird am seriellen Monitor die Einheit "cm" angegeben. |
| delay(1000); } | Das Programm macht eine Pause von 1 Sekunde. Dadurch gibt es etwa in jeder Sekunde einen neuen Messwert. |