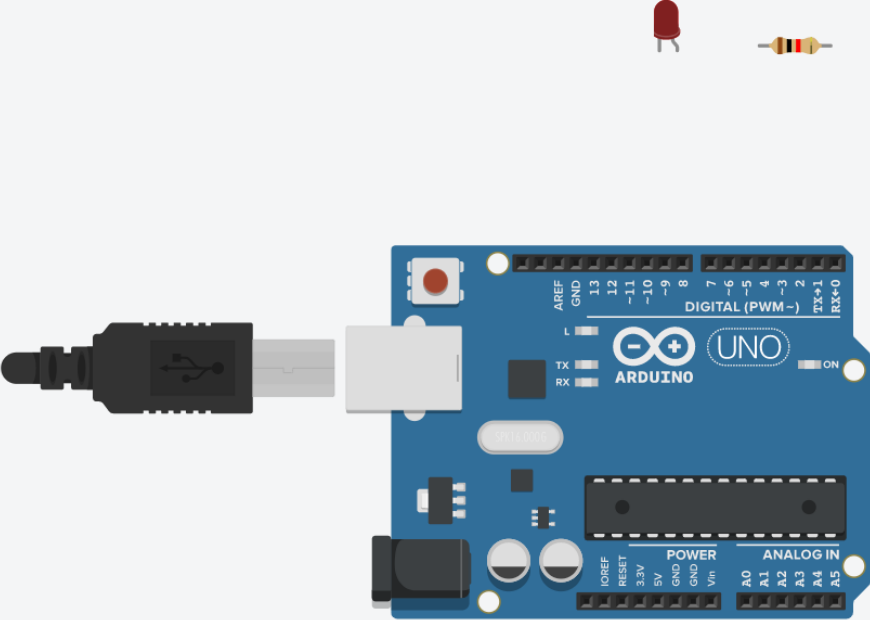



Arbeitsblatt: Übungen zum Entwickeln von Schaltplänen und Programmen

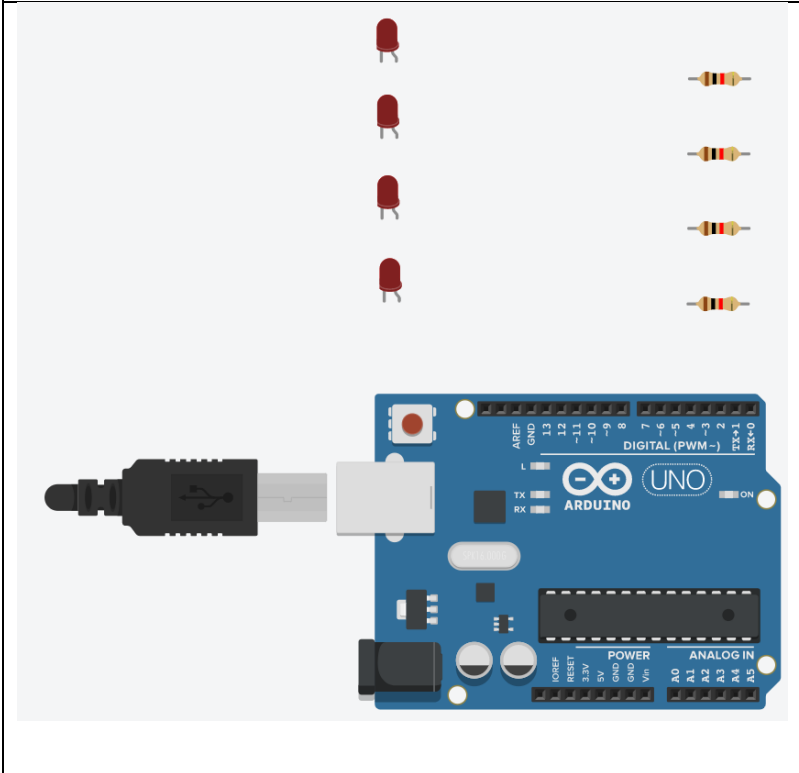
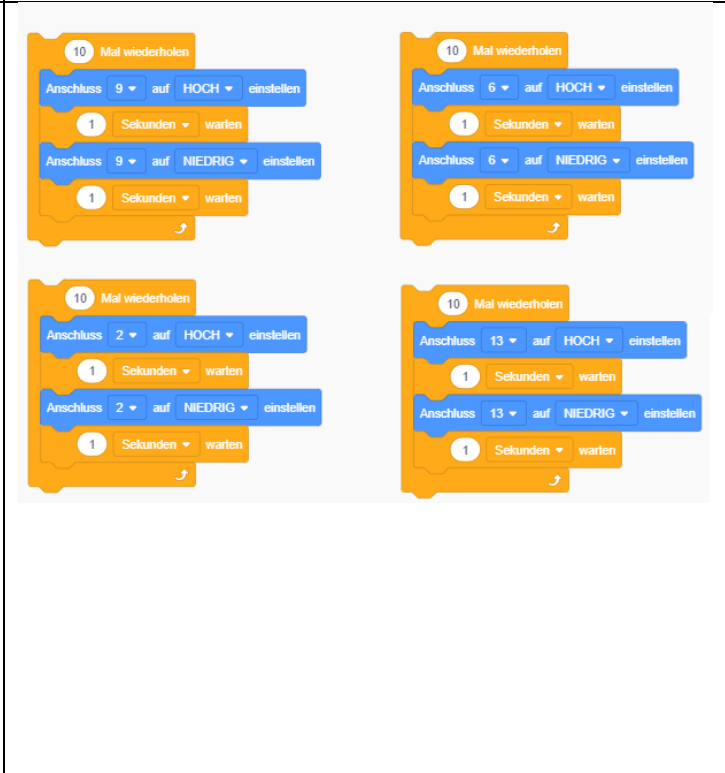
Aufgabe 1.)

G-Niveau + M-Niveau:

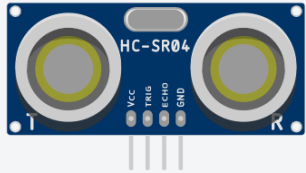
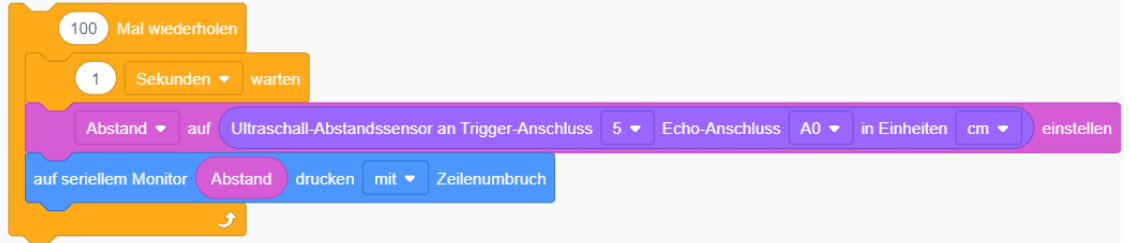
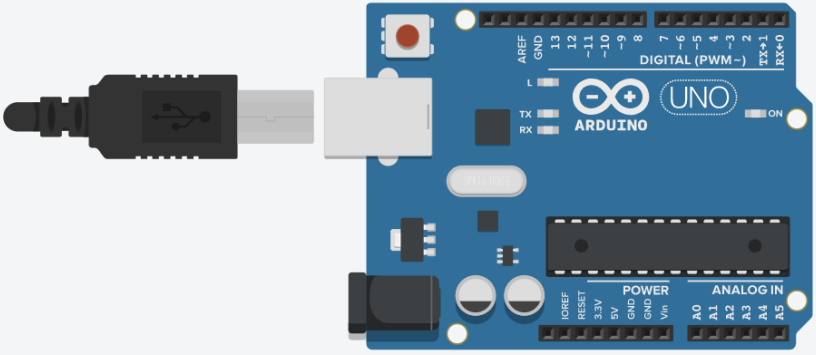
a.) Schaue dir den Programmcode genau an. Verbinde anschließend die elektrischen Bauteile so, dass die LED ordnungsgemäß funktioniert!

Schaltung	(grafikbasierter) Programmcode	(textbasierter) Programmcode
		<pre> 1 int counter; 2 3 void setup() 4 { 5 pinMode(9, OUTPUT); 6 } 7 8 void loop() 9 { 10 for (counter = 0; counter < 10; ++counter) { 11 digitalWrite(9, HIGH); 12 delay(1000); // Wait for 1000 millisecond(s) 13 digitalWrite(9, LOW); 14 delay(1000); // Wait for 1000 millisecond(s) 15 } 16 } </pre>

b.) Schau dir den Programmcode genau an. Verbinde anschließend die elektrischen Bauteile so, dass die LED's ordnungsgemäß funktionieren!

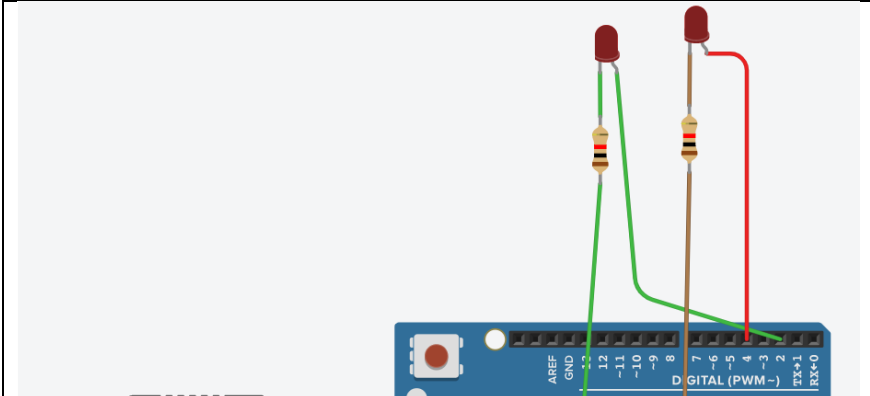
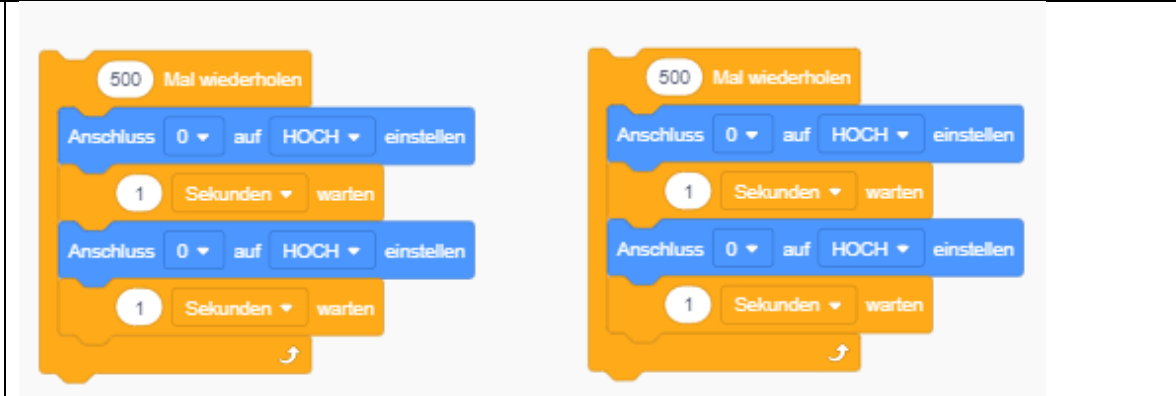
Schaltung	(grafikbasierter) Programmcode	(textbasierter) Programmcode
		<pre data-bbox="1610 304 2150 1080"> 1 int counter; 2 3 int counter2; 4 5 int counter3; 6 7 int counter4; 8 9 void setup() 10 { 11 pinMode(9, OUTPUT); 12 pinMode(6, OUTPUT); 13 pinMode(2, OUTPUT); 14 pinMode(13, OUTPUT); 15 } 16 17 void loop() 18 { 19 for (counter = 0; counter < 10; ++counter) { 20 digitalWrite(9, HIGH); 21 delay(1000); // Wait for 1000 millisecond(s) 22 digitalWrite(9, LOW); 23 delay(1000); // Wait for 1000 millisecond(s) 24 } 25 26 for (counter2 = 0; counter2 < 10; ++counter2) { 27 digitalWrite(6, HIGH); 28 delay(1000); // Wait for 1000 millisecond(s) 29 digitalWrite(6, LOW); 30 delay(1000); // Wait for 1000 millisecond(s) 31 } 32 33 for (counter3 = 0; counter3 < 10; ++counter3) { 34 digitalWrite(2, HIGH); 35 delay(1000); // Wait for 1000 millisecond(s) 36 digitalWrite(2, LOW); 37 delay(1000); // Wait for 1000 millisecond(s) 38 } 39 40 for (counter4 = 0; counter4 < 10; ++counter4) { 41 digitalWrite(13, HIGH); 42 delay(1000); // Wait for 1000 millisecond(s) 43 digitalWrite(13, LOW); 44 delay(1000); // Wait for 1000 millisecond(s) 45 } 46 } </pre>

d.) Schau dir den Programmcode genau an. Verbinde anschließend die elektrischen Bauteile so, dass der Abstandssensor ordnungsgemäß funktioniert!

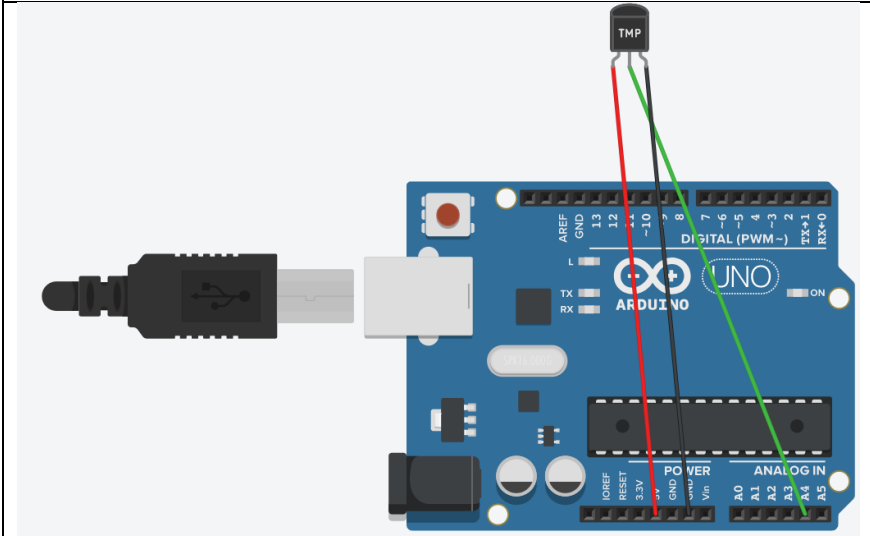
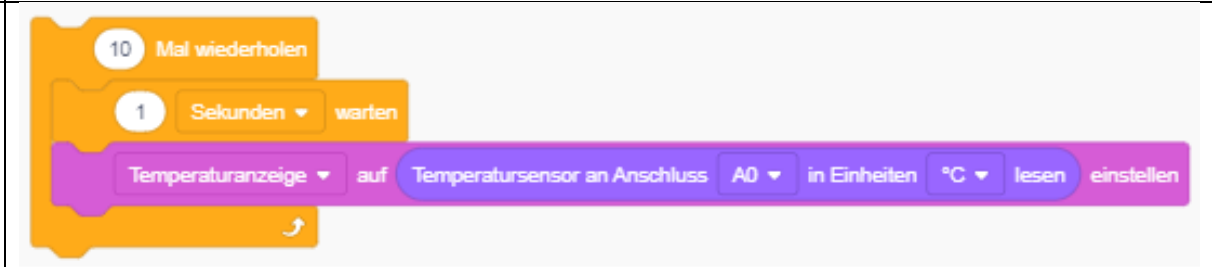
Schaltung	(grafikbasierter) Programmcode
 <p>The image shows an HC-SR04 ultrasonic sensor module. It is a blue PCB with two circular sensors. The pins are labeled: VCC, TRIG, ECHO, and GND.</p>	 <p>The image shows a block-based code snippet. It starts with a '100 Mal wiederholen' (Repeat 100 times) block. Inside, there is a '1 Sekunden warten' (Wait 1 second) block, followed by a block to 'Abstand auf Ultraschall-Abstandssensor an Trigger-Anschluss 5 Echo-Anschluss A0 in Einheiten cm einstellen' (Set distance on ultrasonic sensor at trigger pin 5, echo pin A0, in units cm). The final block is 'auf seriellen Monitor Abstand drucken mit Zeilenumbruch' (Print distance on serial monitor with line break).</p>
 <p>The image shows an Arduino Uno board with a USB cable plugged into the USB-B port. The board has various pins labeled: AREF, GND, 13, 12, -11, -10, -9, 8, 7, -6, -5, 4, 3, -3, -2, TX+, TX-, RX+, RX-, TX, RX, ARDUINO, ON, ICOREF, RESET, 3.3V, 5V, GND, GND, Vin, A0, A1, A2, A3, A4, A5.</p>	<p>(textbasierter) Programmcode</p> <pre> 1 int Abstand = 0; 2 3 long readUltrasonicDistance(int triggerPin, int echoPin) 4 { 5 pinMode(triggerPin, OUTPUT); // Clear the trigger 6 digitalWrite(triggerPin, LOW); 7 delayMicroseconds(2); 8 // Sets the trigger pin to HIGH state for 10 microseconds 9 digitalWrite(triggerPin, HIGH); 10 delayMicroseconds(10); 11 digitalWrite(triggerPin, LOW); 12 pinMode(echoPin, INPUT); 13 // Reads the echo pin, and returns the sound wave travel time in microseconds 14 return pulseIn(echoPin, HIGH); 15 } 16 17 int counter; 18 19 void setup() 20 { 21 Serial.begin(9600); 22 } 23 24 25 void loop() 26 { 27 for (counter = 0; counter < 100; ++counter) { 28 delay(1000); // Wait for 1000 millisecond(s) 29 Abstand = 0.01723 * readUltrasonicDistance(5, A0); 30 Serial.println(Abstand); 31 } 32 } </pre>

Aufgabe 2.)

a.) Schaue dir die Schaltung und den Programmcode genau an! Finde den Fehler und verbessere ihn entsprechend!

Schaltung	G-Niveau: grafikbasierter Programmcode
	
<p>M-Niveau: textbasierter Programmcode</p>	
<pre> 1 int counter; 2 3 int counter2; 4 5 void setup() 6 { 7 pinMode(0, OUTPUT); 8 } 9 10 void loop() 11 { 12 for (counter = 0; counter < 500; ++counter) { 13 digitalWrite(0, HIGH); 14 delay(1000); // Wait for 1000 millisecond(s) 15 digitalWrite(0, HIGH); 16 delay(1000); // Wait for 1000 millisecond(s) 17 } 18 19 for (counter2 = 0; counter2 < 500; ++counter2) { 20 digitalWrite(0, HIGH); 21 delay(1000); // Wait for 1000 millisecond(s) 22 digitalWrite(0, HIGH); 23 delay(1000); // Wait for 1000 millisecond(s) 24 } 25 } </pre>	

b.) Schaue dir die Schaltung und den Programmcode genau an! Finde den Fehler und verbessere ihn entsprechend!

Schaltung	G-Niveau: Grafikbasierter Programmcode
	
M-Niveau: Textbasierter Programmcode	
<pre> 1 int Temperaturanzeige = 0; 2 3 int counter; 4 5 void setup() 6 { 7 pinMode(A0, INPUT); 8 } 9 10 void loop() 11 { 12 for (counter = 0; counter < 10; ++counter) { 13 delay(1000); // Wait for 1000 millisecond(s) 14 Temperaturanzeige = -40 + 0.488155 * (analogRead(A0) - 20); 15 } 16 } </pre>	