



Universität Stuttgart
Institut für Erziehungswissenschaft
Lehrstuhl Berufspädagogik mit Schwerpunkt
Technikdidaktik (BPT)



Mikrocontroller

Lösungen zu den Arbeitsblättern



Arbeitsblatt Nr. 1

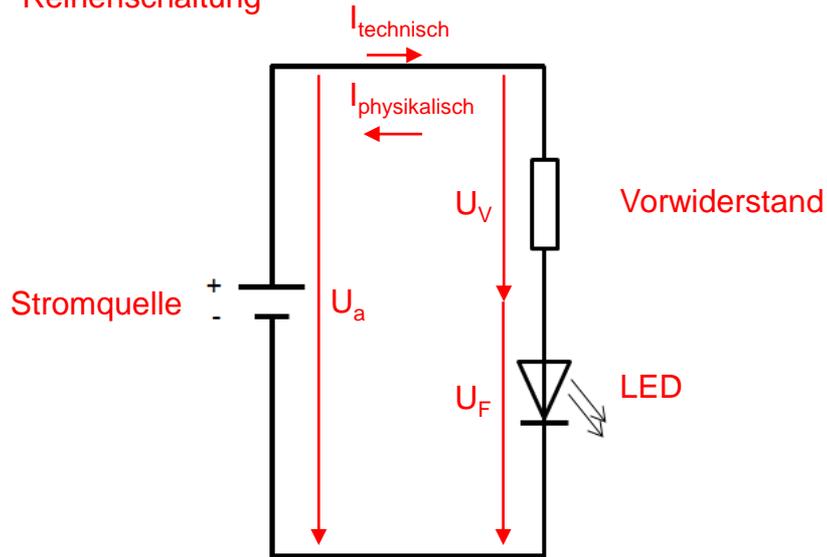
Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 1: Eine LED soll leuchten (ohne Programmierung)



Aufgabe 1 a:

Reihenschaltung



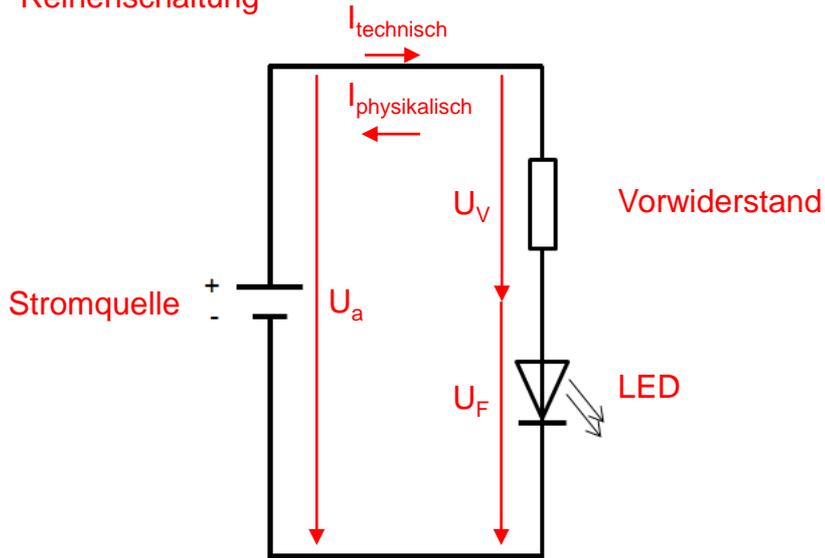
Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 1: Eine LED soll leuchten (ohne Programmierung)

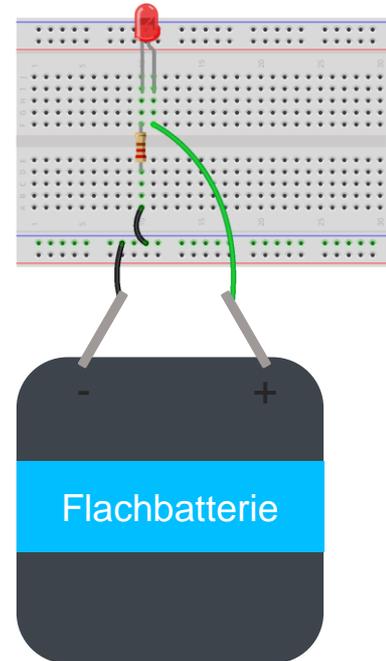


Aufgabe 1 a:

Reihenschaltung



Aufgabe 1 b:



Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 1: Eine LED soll leuchten (ohne Programmierung)



Aufgabe 1 c:

Der Vorwiderstand hat die Aufgabe die LED vor einem zu hohen Spannungsabfall zu schützen sowie den Strom, welcher durch die LED fließt, zu begrenzen.

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 1: Eine LED soll leuchten (ohne Programmierung)



Aufgabe 1 c:

Der Vorwiderstand hat die Aufgabe die LED vor einem zu hohen Spannungsabfall zu schützen sowie den Strom, welcher durch die LED fließt, zu begrenzen.

Aufgabe 1 d:

ABSOLUTE MAXIMUM RATINGS:(Ta=25°C)

Reverse Voltage	: 5 Volt
Reverse Current(Vr=5V)	: 10uA
Operating Temperature Range	: -40 °C to+85 °C
Storage Temperature Range	: -40 °C to+100 °C
Lead Soldering Temperature	: 260 °C for 5 Seconds

PART SELECTION AND APPLICATION INFORMATION(RATINGS AT 25°C AMBIENT)

PART NO.	CHIP			LENS COLOR	Absolute Maximum Ratings				Electrical-Optical Characteris				Viewing Angle (deg)		
	Material	PEAK WAVE Length λp(nm)	Emitting Color		Δλ (nm)	pd (nW)	IF (mA)	Peak If(mA)	VF(V)			Rec If(mA)		Iv(mcd)	
									Min.	Typ.	Max.	Max.	Min.	Typ.	
513BD	InGaN	465	Blue	Blue Difused	20	120	30	100	2.9	3.3	3.6	10-20	300	800	60

$$U_V = U_a - U_F = 5 \text{ V} - 3,3 \text{ V} = 1,7 \text{ V}$$

$$R_V = \frac{1,7 \text{ V}}{15 \text{ mA}} = \frac{1,7 \text{ V}}{0,015 \text{ A}}$$

$$R_V = 113,33 \Omega$$

$$R_V \approx 120 \Omega$$

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 1: Eine LED soll leuchten (ohne Programmierung)



Aufgabe 1 e:

braun (1) – rot (2) – braun (0) – gold (5%)

	Ring 1 & 2	Ring 3
	0	-
Braun	1	0
Rot	2	00
Orange	3	000
Gelb	4	0000
Grün	5	00000
Blau	6	000000
Lila	7	0000000
Grau	8	00000000
Weiß	9	000000000

	Ring 4
Grau	0,05%
Lila	0,10%
Blau	0,25%
Grün	0,50%
Braun	1,00%
Rot	2,00%
Gold	5,00%
Silber	10,00%

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 1: Eine LED soll leuchten (ohne Programmierung)



Aufgabe 1 e:

braun (1) – rot (2) – braun (0) – gold (5%)

	Ring 1 & 2	Ring 3
	0	-
Braun	1	0
Rot	2	00
Orange	3	000
Gelb	4	0000
Grün	5	00000
Blau	6	000000
Lila	7	0000000
Grau	8	00000000
Weiß	9	000000000

	Ring 4
Grau	0,05%
Lila	0,10%
Blau	0,25%
Grün	0,50%
Braun	1,00%
Rot	2,00%
Gold	5,00%
Silber	10,00%

Kopfnuss 1:

Der Widerstand müsste bei einer höheren äußeren angelegten Spannung von 9 V größer gewählt werden.

$$(R_V = \frac{9 \text{ V} - 3,3 \text{ V}}{15 \text{ mA}} = \frac{5,7 \text{ V}}{0,015 \text{ A}} = 380 \Omega)$$

Wird dies nicht berücksichtigt, so geht die LED kaputt. Sie leuchtet kurz einmal auf und dann nie wieder.



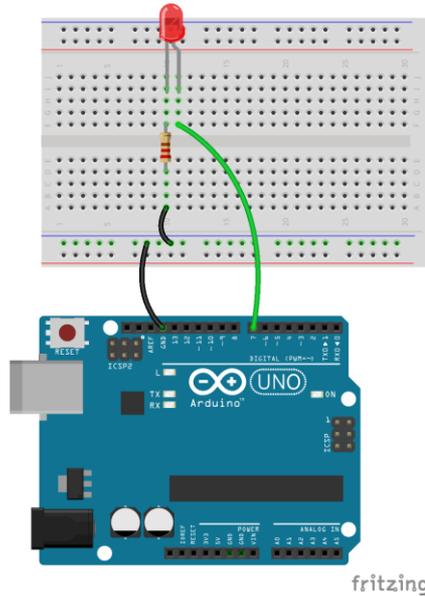
Arbeitsblatt Nr. 2

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 2: Eine LED soll leuchten (mit Programmierung)



Aufgabe 2 a:

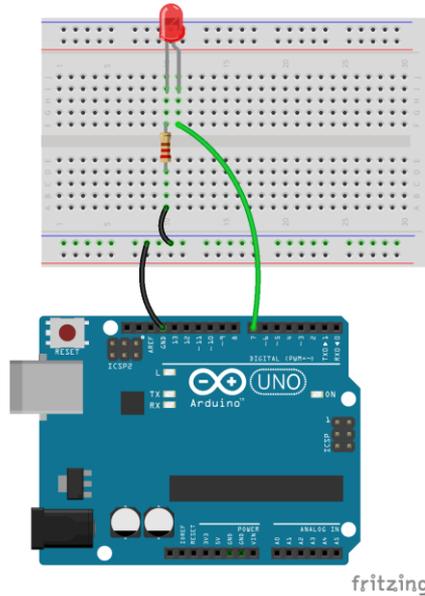


Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 2: Eine LED soll leuchten (mit Programmierung)



Aufgabe 2 a:



Aufgabe 2 b:

```
void setup() {  
  pinMode(5, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(5, HIGH);  
}
```

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 2: Eine LED soll leuchten (mit Programmierung)



Aufgabe 2 c:

Setup-Bereich:

Die Befehle in diesem Bereich werden einmal durchlaufen. Hierbei wird beispielsweise der Pin 5 verwendet, an dem eine LED angeschlossen ist. Da hier ein Spannungssignal ausgegeben wird, muss dieser als OUTPUT festgelegt werden.

Loop-Bereich:

Die Befehle in diesem Bereich werden nach Erreichen des Endes des Loop-Bereichs erneut wiederholt (Loop= Schleife). Die LED, welche an Pin 5 angeschlossen ist, soll leuchten. Das bedeutet, es soll ein hohes Spannungssignal am Pin 5 ($\text{HIGH} \triangleq 5 \text{ V}$) ausgegeben werden.

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 2: Eine LED soll leuchten (mit Programmierung)



Aufgabe 2 c:

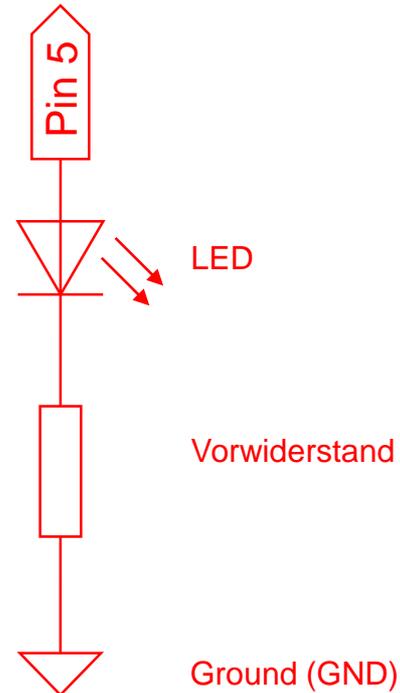
Setup-Bereich:

Die Befehle in diesem Bereich werden einmal durchlaufen. Hierbei wird beispielsweise der Pin 5 verwendet, an dem eine LED angeschlossen ist. Da hier ein Spannungssignal ausgegeben wird, muss dieser als OUTPUT festgelegt werden.

Loop-Bereich:

Die Befehle in diesem Bereich werden nach Erreichen des Endes des Loop-Bereichs erneut wiederholt (Loop= Schleife). Die LED, welche an Pin 5 angeschlossen ist, soll leuchten. Das bedeutet, es soll ein hohes Spannungssignal am Pin 5 ($\text{HIGH} \triangleq 5 \text{ V}$) ausgegeben werden.

Kopfnussaufgabe 2:





Arbeitsblatt Nr. 3

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 3: Blinklicht



Aufgabe 3 a:

```
void setup()
{
  pinMode(3, OUTPUT);
}

void loop()
{
  digitalWrite(3, HIGH);
  delay(2000);
  digitalWrite(3, LOW);
  delay(2000);
}
```

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 3: Blinklicht



Aufgabe 3 a:

```
void setup()
{
  pinMode(3, OUTPUT);
}

void loop()
{
  digitalWrite(3,HIGH);
  delay(2000);
  digitalWrite(3,LOW);
  delay(2000);
}
```

Aufgabe 3 b:

Um die LED schneller blinken zu lassen, muss im delay-Befehl eine kurze Sekundenangabe (bspw. 200 ms) eingegeben werden.

```
void setup()
{
  pinMode(3, OUTPUT);
}

void loop()
{
  digitalWrite(3,HIGH);
  delay(200);
  digitalWrite(3,LOW);
  delay(200);
}
```

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 3: Blinklicht



Aufgabe 3 c:

```
void setup()
{
  pinMode(3, OUTPUT); //LED rot
  pinMode(4, OUTPUT); //LED gelb
  pinMode(5, OUTPUT); //LED grün
}

void loop()
{
  digitalWrite(3,HIGH);
  digitalWrite(4,LOW);
  digitalWrite(5,LOW);
  delay(3000);
  digitalWrite(3,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW);
  delay(3000);
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
  digitalWrite(5,HIGH);
  delay(3000);
}
```

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 3: Blinklicht



Aufgabe 3 c:

```
void setup()
{
  pinMode(3, OUTPUT); //LED rot
  pinMode(4, OUTPUT); //LED gelb
  pinMode(5, OUTPUT); //LED grün
}

void loop()
{
  digitalWrite(3,HIGH);
  digitalWrite(4,LOW);
  digitalWrite(5,LOW);
  delay(3000);
  digitalWrite(3,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW);
  delay(3000);
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
  digitalWrite(5,HIGH);
  delay(3000);
}
```

Aufgabe 3 d:

```
void setup() {
  pinMode(5, INPUT);
}

void loop() {
  digitalWrite(5, HIGH);
  delay(200);
  digitalWrite(5,LOW);
  delay(5000);
}
```

Eine LED wird im Setup-Bereich als OUTPUT festgelegt.

Die LED soll für 2 s leuchten.
1 s $\hat{=}$ 1000 ms
Es müsse daher delay(2000); geschrieben werden.

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 3: Blinklicht



Kopfnussaufgabe 3:

```
void setup()
{
  pinMode(3, OUTPUT); //LED rot
  pinMode(4, OUTPUT); //LED gelb
  pinMode(5, OUTPUT); //LED grün
}

void loop()
{
  digitalWrite(3,HIGH);
  digitalWrite(4,LOW);
  digitalWrite(5,LOW);
  delay(100);
  digitalWrite(3,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW);
  delay(100);
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
  digitalWrite(5,HIGH);
  delay(100);
  digitalWrite(3,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW);
  delay(100);
}
```

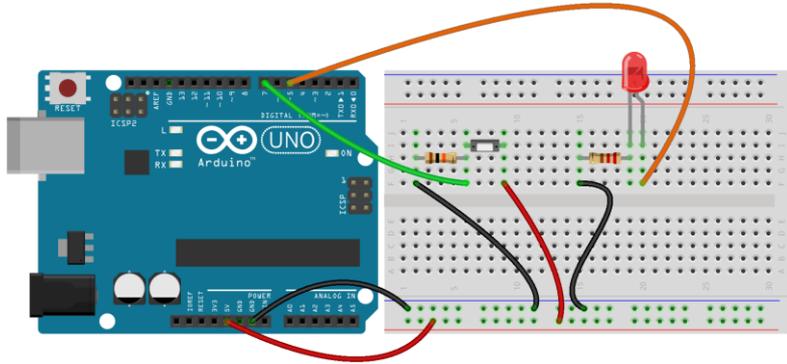


Arbeitsblatt Nr. 4

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 4: Die LED leuchtet auf Knopfdruck

Aufgabe 4 a:

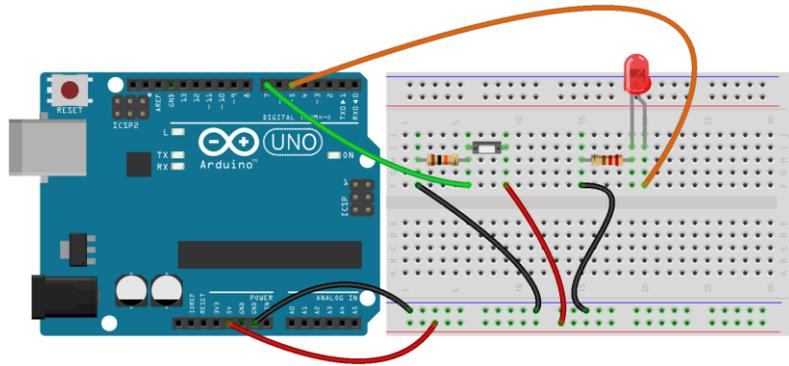


fritzing

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 4: Die LED leuchtet auf Knopfdruck

Aufgabe 4 a:



fritzing

Aufgabe 4 b:

```
void setup()
{
  pinMode(5,OUTPUT);//LED
  pinMode(7,INPUT);//Taster
}

void loop()
{
  if(digitalRead(7) == HIGH)
  {
    /*wenn der Taster gedrückt ist, soll eine Anweisung
    ausgeführt werden */
  }
  else
  {
    /*wenn der Taster gedrückt ist, soll eine
    andere Anweisung ausgeführt werden */
  }
}
```

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 4: Die LED leuchtet auf Knopfdruck



Aufgabe 4 c:

```
void setup()
{
  pinMode(5, OUTPUT); //LED rot
  pinMode(7, INPUT); //Taster
}

void loop()
{
  if(digitalRead(7)==HIGH)
  {
    digitalWrite(5, HIGH);
    delay(2000);
  }
  else
  {
    digitalWrite(5, LOW);
  }
}
```

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 4: Die LED leuchtet auf Knopfdruck



Kopfnuss 4:

```
void setup()
{
  pinMode(5, OUTPUT); //LED rot
  pinMode(6, OUTPUT); //LED gelb
  pinMode(7, INPUT); //Taster
}
```

```
void loop()
{
  if(digitalRead(7)==HIGH)
  {
    digitalWrite(6, LOW);
    digitalWrite(5, HIGH);
    delay(1000);
    digitalWrite(5, LOW);
    delay(1000);
    digitalWrite(5, HIGH);
    delay(1000);
    digitalWrite(5, LOW);
    delay(1000);
  }
}
```

```
else
{
  digitalWrite(6, HIGH);
  digitalWrite(5, LOW);
}
}
```



Arbeitsblatt Nr. 5

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 5: Der Mikrocontroller macht Töne



Aufgabe 5 a:

```
void setup()
{
  pinMode(5, OUTPUT); //Piezo
}

void loop()
{
  tone(5, 440, 3000);
  delay(3000);
  noTone(5);
  delay(3000);
}
```

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 5: Der Mikrocontroller macht Töne



Aufgabe 5 a:

```
void setup()
{
  pinMode(5, OUTPUT); //Piezo
}

void loop()
{
  tone(5, 440, 3000);
  delay(3000);
  noTone(5);
  delay(3000);
}
```

Aufgabe 5 b:

Siehe Schulungsheft

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 5: Der Mikrocontroller macht Töne



Aufgabe 5 c:

```
void setup()
{
  pinMode(5, OUTPUT); //Piezo
  pinMode(6, OUTPUT); //LED rot
  pinMode(7, INPUT); //Taster
}

void loop()
{
  if(digitalRead(7) == HIGH)
  {
    digitalWrite(6,HIGH);
    tone(5,440,2000);
    delay(2000);
  }
  else
  {
    digitalWrite(6,LOW);
    noTone(5);
    delay(3000);
  }
}
```

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 5: Der Mikrocontroller macht Töne

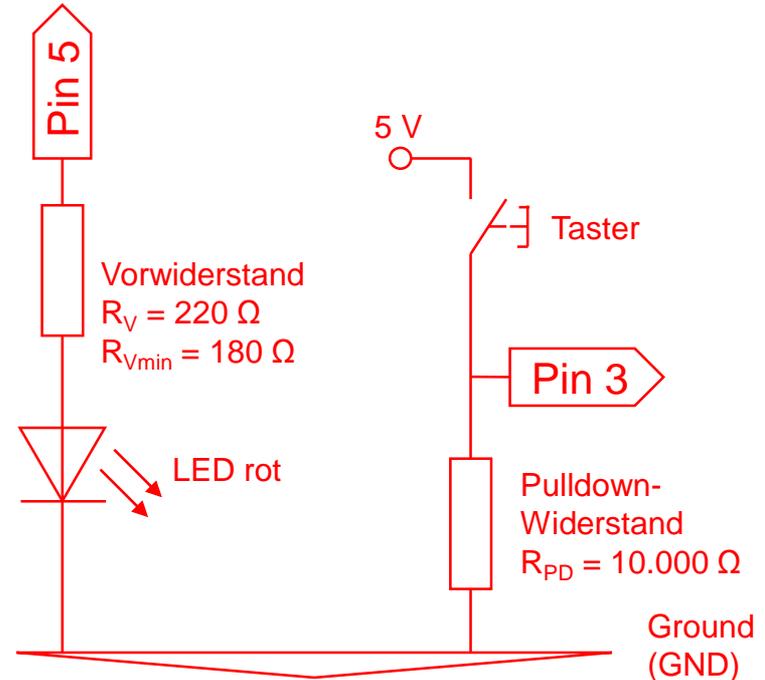


Aufgabe 5 c:

```
void setup()
{
  pinMode(5, OUTPUT); //Piezo
  pinMode(6, OUTPUT); //LED rot
  pinMode(7, INPUT); //Taster
}

void loop()
{
  if(digitalRead(7) == HIGH)
  {
    digitalWrite(6, HIGH);
    tone(5, 440, 2000);
    delay(2000);
  }
  else
  {
    digitalWrite(6, LOW);
    noTone(5);
    delay(3000);
  }
}
```

Kopfnuss 5:





Arbeitsblatt Nr. 6

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 6: Die Verkehrsampel I



```
void setup()
{
  pinMode(3, OUTPUT); //Piezo
  pinMode(4, OUTPUT); //Auto rot
  pinMode(5, OUTPUT); //Auto gelb
  pinMode(6, OUTPUT); //Auto grün
  pinMode(7, OUTPUT); //Fuss rot
  pinMode(8, OUTPUT); //Fuss grün
  pinMode(9, INPUT); //Taster
}

void loop()
{
  if(digitalRead(9) == HIGH)
  {
    //Fußgängerampel ist rot, Autoampel ist grün
    digitalWrite(4,LOW);
    digitalWrite(5,LOW);
    digitalWrite(6,HIGH);
    digitalWrite(7,HIGH);
    digitalWrite(8,LOW);
    delay(200);
  }
}
```

```
//Fußgängerampel ist rot, Autoampel ist gelb
digitalWrite(4,LOW);
digitalWrite(5,HIGH);
digitalWrite(6,LOW);
digitalWrite(7,HIGH);
digitalWrite(8,LOW);
delay(2000);

//Fußgängerampel ist rot, Autoampel ist rot
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,LOW);
digitalWrite(7,HIGH);
digitalWrite(8,LOW);
delay(2000);

//Fußgängerampel ist grün, Autoampel ist rot
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
digitalWrite(8,HIGH);
tone(3,440,6000);
delay(6000);
```

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 6: Die Verkehrsampel I



```
//Fußgängerampel ist rot, Autoampel ist rot
digitalWrite (4,HIGH);
digitalWrite (5,LOW);
digitalWrite (6,LOW);
digitalWrite (7,HIGH);
digitalWrite (8,LOW);
noTone (3);
delay(2000);

//Fußgängerampel ist rot, Autoampel ist rot-gelb
digitalWrite (4,HIGH);
digitalWrite (5,HIGH);
digitalWrite (6,LOW);
digitalWrite (7,HIGH);
digitalWrite (8,LOW);
delay(1000);
}
else
{
//Fußgängerampel ist rot, Autoampel ist grün
digitalWrite (4,LOW);
digitalWrite (5,LOW);
digitalWrite (6,HIGH);
digitalWrite (7,HIGH);
digitalWrite (8,LOW);
delay(200);
}
}
```



Arbeitsblatt Nr. 7

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 7: Die Verkehrsampel II - optional



Aufgabe 7:

Nach der Initialisierung fehlt das Semikolon

```
int RAuto=12;
int GAuto=11;
int GrAuto=10;
int RFuss=7;
int GrFuss=6;
int Pieps=2;
int Button=4;
int Druck
```

Der Piezo-speaker mit dem Namen „Pieps“ ist ein OUTPUT

```
void setup()
{
  pinMode(RAuto,OUTPUT);
  pinMode(GAuto,OUTPUT);
  pinMode(GrAuto,OUTPUT);
  pinMode(RFuss,OUTPUT);
  pinMode(GrFuss,OUTPUT);
  pinMode(Pieps,INPUT);
  pinMode(Button,INPUT);
}
```

Der Befehl digitalRead (...) ist falsch geschrieben, daher auch nicht farblich hervorgehoben!

```
void loop() {
  Druck = Digitalread(Button);

  if (Druck == HIGH)
  {
    digitalWrite(RAuto, LOW);
    digitalWrite(GAuto, LOW);
    digitalWrite(GrAuto, HIGH);
    digitalWrite(RFuss, HIGH);
    digitalWrite(GrFuss, LOW);
    delay(200);
  }
}
```

```
digitalWrite(RAuto, LOW);
digitalWrite(GAuto, HIGH);
digitalWrite(GrAuto, LOW);
digitalWrite(RFuss, HIGH);
digitalWrite(GrFuss, LOW);
delay(2000);
```

```
digitalWrite(RAuto, HIGH);
digitalWrite(GAuto, LOW);
digitalWrite(GrAuto, LOW);
digitalWrite(RFuss, HIGH);
digitalWrite(GrFuss, LOW);
delay(2000);
```

```
digitalWrite(RAuto, HIGH);
digitalWrite(GAuto, LOW);
digitalWrite(GrAuto, LOW);
digitalWrite(RFuss, LOW);
digitalWrite(GrFuss, HIGH);
tone(Pieps, 450);
delay(6000);
```

```
digitalWrite(RAuto, HIGH);
digitalWrite(GAuto, LOW);
digitalWrite(GrAuto, LOW);
digitalWrite(RFuss, HIGH);
digitalWrite(GrFuss, LOW);
noTone(Pieps);
delay(2000);
```

```
digitalWrite(RAuto, HIGH);
digitalWrite(GAuto, HIGH);
digitalWrite(GrAuto, LOW);
digitalWrite(RFuss, HIGH);
digitalWrite(GrFuss, LOW);
delay(2000);
```

```
else
{
  digitalWrite(RAuto, LOW);
  digitalWrite(GAuto, LOW);
  digitalWrite(GrAuto, HIGH);
  digitalWrite(RFuss, HIGH);
  digitalWrite(GrFuss, LOW);
}
}
```

Der if-Bereich muss mit einer geschweiften Klammer abgeschlossen werden, bevor der else-Bereich beginnt.



Arbeitsblatt Nr. 8

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 8: Variablen definieren

Aufgabe 8:

```
int Piepser = 3;
int Autorot = 4;
int Autogelb = 5;
int Autogruen = 6;
int Fussrot = 7;
int Fussgruen = 8;
int Taster = 9;

void setup()
{
  pinMode(Piepser, OUTPUT); //Piezo
  pinMode(Autorot, OUTPUT); //Auto rot
  pinMode(Autogelb, OUTPUT); //Auto gelb
  pinMode(Autogruen, OUTPUT); //Auto grün
  pinMode(Fussrot, OUTPUT); //Fuss rot
  pinMode(Fussgruen, OUTPUT); //Fuss grün
  pinMode(Taster, INPUT); //Taster
}

void loop()
{
  if(digitalRead(Taster) == HIGH)
  {
    //Fußgängerampel ist rot, Autoampel ist grün
    digitalWrite(Autorot,LOW);
    digitalWrite(Autogelb,LOW);
    digitalWrite(Autogruen,HIGH);
  }
}
```

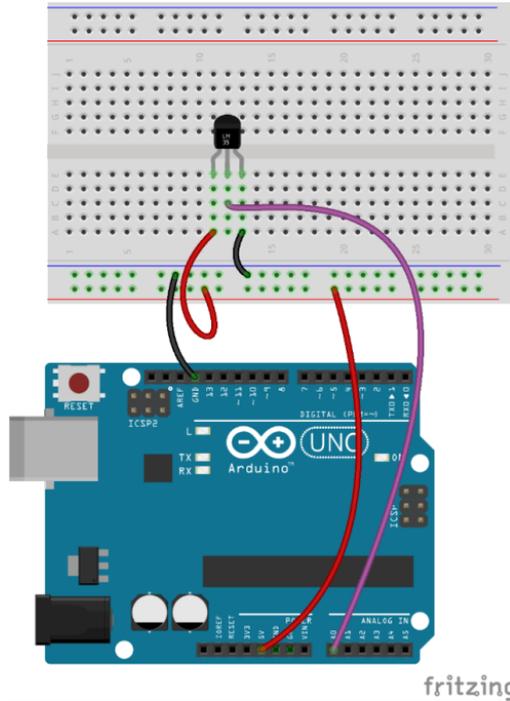


Arbeitsblatt Nr. 9

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 9: Der Temperatursfühler

Aufgabe 9 a:

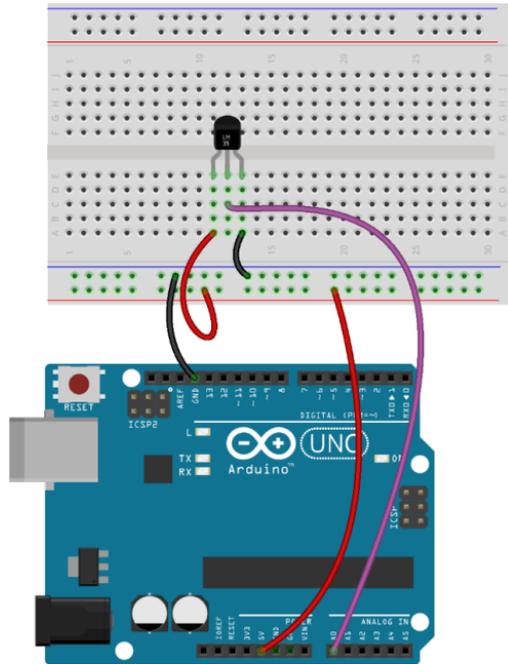


Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 9: Der Temperatursfühler



Aufgabe 9 a:



fritzing

Aufgabe 9 b:

```
int TempSensor=A0;
int Wert;
int temperatur=0;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  Wert=analogRead(TempSensor);
  temperatur=map(Wert,0,410,-50,150);
  delay(500);
  Serial.print(temperatur);
  Serial.println(" Grad Celcius!");
}
```

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 9: Der Temperaturfühler



Aufgabe 9 c:

Initialisierungs- und Deklarationsbereich:

Mit dem Variablentyp „int“ werden die benötigten Variablen festgelegt. Das ist zum einen der Temperatursensor „TempSensor“, der an den analogen Eingang A0 angeschlossen ist und unter der zweiten Variable „Wert“ ausgelesen wird und mit dem entsprechenden Messwert abgespeichert wird. Dieser gemessene Wert liegt zwischen 0 und 410. (Eine ausführlichere Erklärung ist im Skript zu finden.)

Setup-Bereich:

Über den Befehl `serial.begin(9600);` wird die Datenübertragungsrate festgelegt.

Loop-Bereich:

Mit der „map“ Funktion wird dem ausgelesenen Messwert eine Zahl zwischen -50°C und 150°C zugeordnet und unter der Variable „temperatur“ abgespeichert. Nach jeder Messung erfolgt eine Pause von 500 ms. Anschließend wird der zugeordnete Wert am seriellen Monitor mit dem Befehl „Serial.print“ dargestellt und mit der Einheit Grad Celcius mit dem Befehl „Serial.println“ versehen

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 9: Der Temperaturfühler



Kopfnussaufgabe 9.1:

```
int Temperatursensor = A0;
int Wert;
int Temperatur = 0;

int LEDgruen = 3;
int LEDrot = 4;
int LEDblau = 5

void setup()
{
  pinMode(LEDgruen, OUTPUT);
  pinMode(LEDrot, OUTPUT);
  pinMode(LEDblau, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  Wert = analogRead(Temperatursensor);
  Temperatur = map(Wert, 0, 410, -50, 150);
  delay(500);
  Serial.print(Temperatur);
  Serial.println (" Grad Celcius!");
  delay(500);
}
```

```
if(20 <= Temperatur && Temperatur >= 22)
{
  digitalWrite(LEDgruen, HIGH);
  digitalWrite(LEDrot, LOW);
  digitalWrite(LEDblau, LOW);
}

if(Temperatur > 22)
{
  digitalWrite(LEDgruen, LOW);
  digitalWrite(LEDrot, HIGH);
  digitalWrite(LEDblau, LOW);
}

if(Temperatur < 20)
{
  digitalWrite(LEDgruen, LOW);
  digitalWrite(LEDrot, LOW);
  digitalWrite(LEDblau, HIGH);
}
}
```

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 9: Der Temperaturfühler



Kopfnussaufgabe 9.2:

```
int Temperatursensor = A0;
int Wert;
int Temperatur = 0;

int Piezo = 7;

void setup()
{
  pinMode(Piezo,OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  Wert = analogRead(Temperatursensor);
  Temperatur = map(Wert,0,410,-50,150);
  delay(500);
  Serial.print(Temperatur);
  Serial.println (" Grad Celcius!");
  delay(500);
}
```

```
if(Temperatur > 60)
{
  tone(Piezo, 440);
  delay(500);
  noTone(Piezo);
  delay(500);
}

else
{
  noTone(Piezo);
}
}
```



Arbeitsblatt Nr. 10

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 10: Der Tropfensensor

Aufgabe 10 a:

Eingabe: Bei der Eingabe handelt es sich um das Spannungssignal vom Tropfensensor, welches am μ C-Board anliegt.

Verarbeitung: Die Verarbeitung der Daten erfolgt mit Hilfe des geschriebenen Sketches, welcher im μ C gespeichert ist.

Ausgabe: Bei der Ausgabe handelt es sich zum Beispiel um ein Lichtsignal, welches von einer LED ausgegeben wird.



Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 10: Der Tropfensensor

Aufgabe 10 a:

Eingabe: Bei der Eingabe handelt es sich um das Spannungssignal vom Tropfensensor, welches am μ C-Board anliegt.

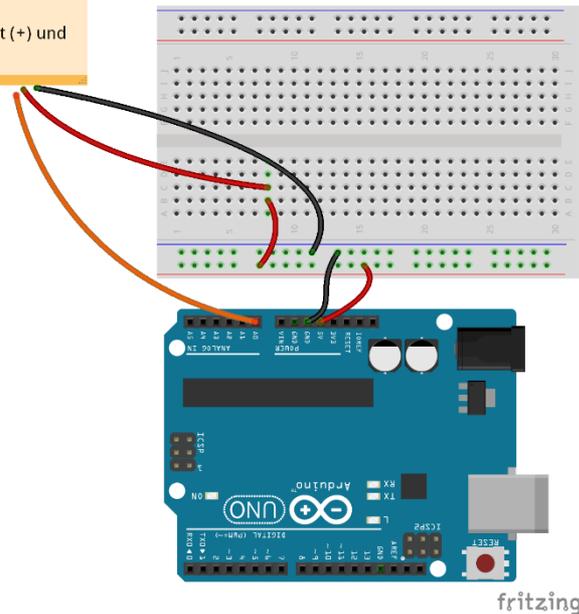
Verarbeitung: Die Verarbeitung der Daten erfolgt mit Hilfe des geschriebenen Sketches, welcher im μ C gespeichert ist.

Ausgabe: Bei der Ausgabe handelt es sich zum Beispiel um ein Lichtsignal, welches von einer LED ausgegeben wird.



Aufgabe 10 b:

Regensensor
Anschlüsse
orange (S), rot (+) und
schwarz (-)



fritzing

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 10: Der Tropfensensor



Aufgabe 10 c:

```
int Tropfensensor = A0;
int Wert = 0;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  Wert = analogRead(Tropfensensor);
  Serial.println(Wert);
  delay(1000);
}
```

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 10: Der Tropfensensor



Kopfnussaufgabe 10:

```
int Tropfensensor = A0;
int Wert = 0;
int LED = 3;
int Piepser = 4;

void setup()
{
  pinMode(LED, OUTPUT);
  pinMode(Piepser, OUTPUT);

  Serial.begin(9600);
}

void loop()
{
  Wert = analogRead(Tropfensensor);
  Serial.println(Wert);
  delay(1000);

  if(Wert > 60 && Wert < 600)
  {
    digitalWrite(LED, HIGH);
    tone(Piepser, 440, 2000);
    delay(2000);
  }
}
```

```
if(Wert > 600)
{
  digitalWrite(LED, HIGH);
  tone(Piepser, 880, 2000);
  delay(2000);
}

else()
{
  digitalWrite(LED, LOW);
  noTone(Piepser);
}
}
```



Arbeitsblatt Nr. 11

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 11: Der Helligkeitssensor



Aufgabe 11 a:

Siehe Schulungsheft

```
int Helligkeitssensor=A2;
int Helligkeitswert=0;

void setup() {

  Serial.begin(9600);
}

void loop() {

  Helligkeitswert=analogRead(Helligkeitssensor);
  Serial.println(Helligkeitswert);
}
```

Mikrocontroller Fortbildung

Lösungen Arbeitsblatt Nr. 11: Der Helligkeitssensor



Aufgabe 11 b:

```
#include <Servo.h>

Servo Servolein;
int Helligkeitssensor = A2;
int Wert = 0;

void setup()
{
  Servolein.attach(3);
  Serial.begin(9600);
}

void loop()
{
  Wert = analogRead(Helligkeitssensor);
  Serial.println(Wert);

  if(Wert > 125)
  {
    Servolein.write(180);
    delay(1000);
  }

  else
  {
    Servolein.write(90);
    delay(1000);
  }
}
```