

arduino.langheiter.at

arduino lernen

PDF generated December 29, 2021 by Kalin's PDF Creation Station WordPress plugin

Table Of Contents

02. Blinken	3
03. mehr blinken	5
02. Blinken	7
03. mehr blinken	9
04.	11
05. TABs	12
06. fotowiderstand	13
07. RGB-LEDs	14
08. lauflicht	16
09. Potentiometer01	18
10.	20
11. Potentiometer03	21
4.1	22
4.6 windrad	23
4.7 wechselblinker	24
5.1 RGB-LED	25
7.1 ampel	26
8.1 ton	27
9.1 serieller monitor	28
9.2 :: gesicht	29
9.3 :: befehle	30
Bauteile	31
Beispiel-Seite	32
Datenschutzerklärung	33

02. Blinken

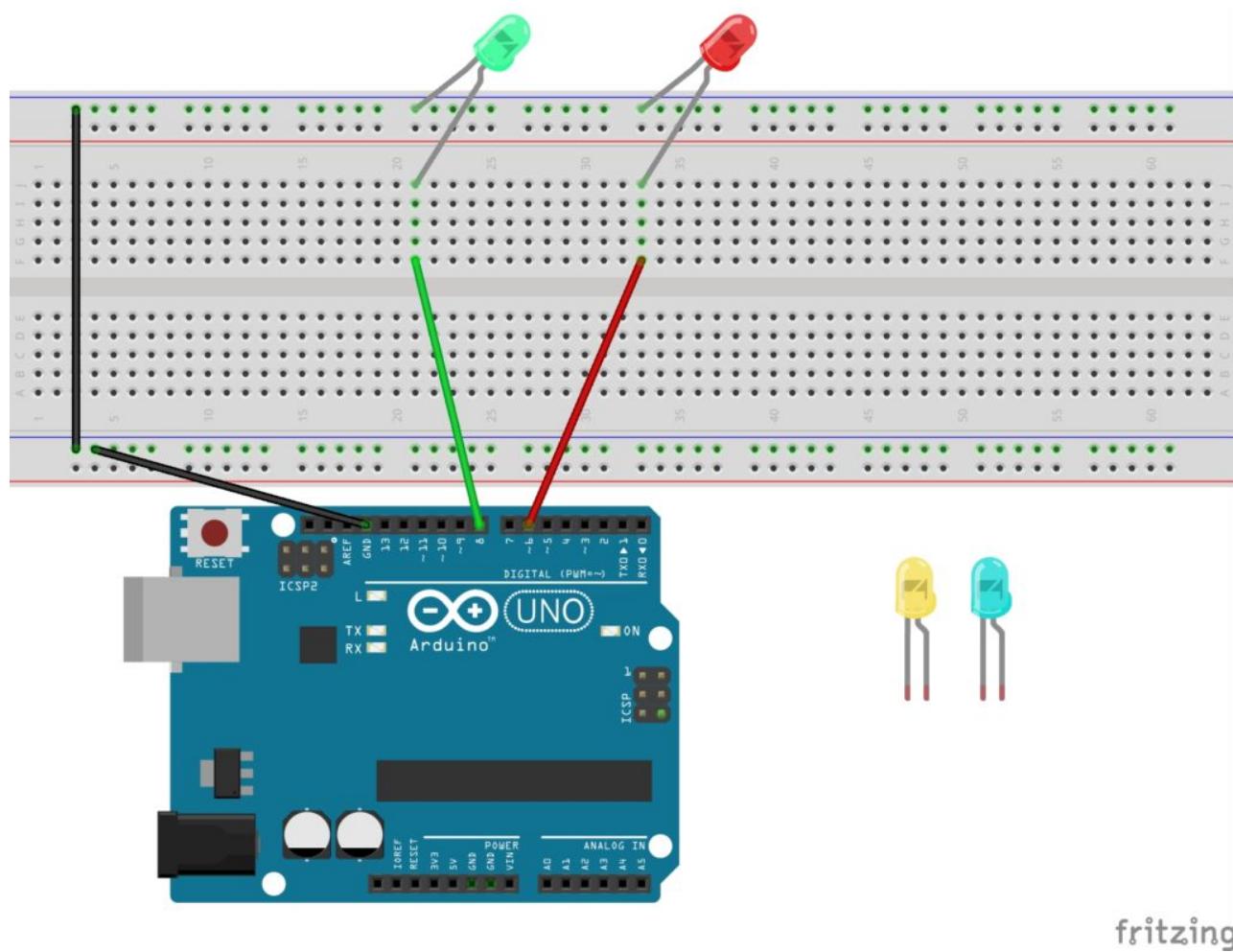
by rainer - Donnerstag, Januar 28, 2021

<https://arduino.langheiter.at/02-blinken/>

folgende anweisungen/befehle werden verwendet:

- [int](#)
- [pinMode](#)
- [digitalWrite](#)
- [delay](#)

du weisst nun, dass ein arduino programm



du sprichst wahrscheinlich deutsch und eventuell noch eine andere sprache. der arduino versteht die programmiersprache [c++](#). c++ existiert seit dem jahre 1979. die älteste noch verwendete programmiersprache heisst [fortran](#) und wurde 1954 entworfen. [hier findest du mehr infos](#) zur geschichte der programmiersprachen.

lass uns das programm zeile für zeile durchgehen:

- 1: hier wird die variable **roteLED** deklariert. das deklarieren einer variable bedeutet, dass ihr typ definiert wird und optional ein anfangswert gesetzt wird. der type int sagt aus, dass wir eine ganzzahl im bereich -2.147.483.648 bis 2.147.483.647 darin speichern können. der anfangswert ist 6, da unsere rote LED mit dem pin 6 verbunden ist.
- 2: **grueneLED** wird deklariert und erhält den anfangswert 8
- 3: leerzeile damit wir das programm leichter lesen können
- 4: // dies bedeutet: kommentar, anmerkung. mach ausreichend davon gebrauch. du wirst sehen wie leicht mensch vergisst
- 6: setup-block. ein block beginnt mit { und endet mit }. die anzahl der öffnenden klammern muss genau der anzahl der schliessenden klammern entsprechen.
- 7: konfiguriert den spezifizierten pin als input oder output. der pin an welchem die rote LED angeschlossen ist, wird als ausgabe gesetzt.
- 8: die variable **grueneLED** hat einen wert von 8. am pin 8 ist unsere grüne LED angeschlossen
- 9: die grüne LED wird eingeschaltet. **digitalWrite** hat 2 parameter: die pinnummer und [HIGH/LOW]. HIGH bedeutet, dass 5Volt gesetzt werden. dadurch kann die LED leuchten. bei LOW, wird die spannung auf 0Volt gesetzt und die LED ist somit aus.
- 14: loop-block beginnt und endet in zeile 19. mit der tastenkombination STRG+T kannst du die richtigen einrückungen erzeugen. dies erleichtert das lesen eines programms sehr. später wirst du programme schreiben, welche verschachtelte logische blöcke enthalten
- 16: unterbricht das programm für die als parameter angegebene zeit (in millisekunden). (es gibt 1000 millisekunden in einer sekunde.)
- im loop: **roteLED** wird eingeschaltet ==> 1000msec wird gewartet ==> **roteLED** aus ==> 500msec warten ==> **roteLED** an ==> warten & immer so weiter. bis der arduino keinen strom mehr hat oder du ein anderes programm rauflädtst

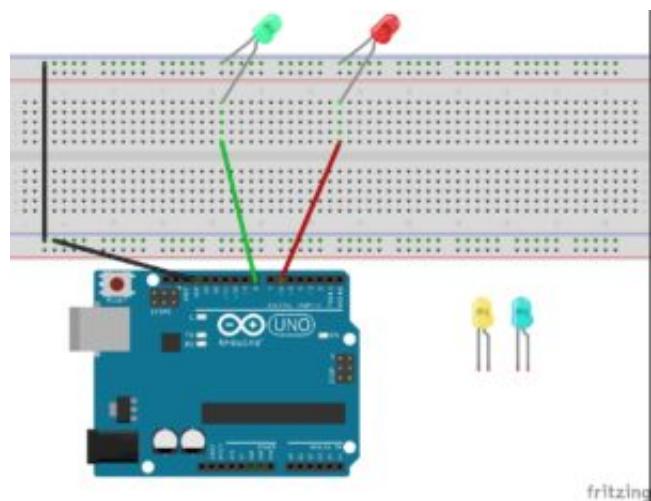
03. mehr blinken

by rainer - Donnerstag, Januar 28, 2021

<https://arduino.langheiter.at/int/>

folgende anweisungen/befehle werden verwendet:

- [int](#)
- [pinMode](#)
- [digitalWrite](#)
- [delay](#)



du sprichst wahrscheinlich deutsch und eventuell noch eine andere sprache. der arduino versteht die programmiersprache [c++](#). c++ existiert seit dem jahre 1979. die älteste noch verwendete programmiersprache heisst [fortran](#) und wurde 1954 entworfen. [hier findest du mehr infos](#) zur geschichte der programmiersprachen.

lass uns das programm zeile für zeile durchgehen:

- 1: hier wird die variable **roteLED** deklariert. das deklarieren einer variable bedeutet, dass ihr typ definiert wird und optional ein anfangswert gesetzt wird. der type int sagt aus, dass wir eine ganzzahl im bereich -2.147.483.648 bis 2.147.483.647 darin speichern können. der anfangswert ist 6, da unsere rote LED mit dem pin 6 verbunden ist.
- 2: grueneLED wird deklariert und erhält den anfangswert 8
- 3: leerzeile damit wir das programm leichter lesen können
- 4: // dies bedeutet: kommentar, anmerkung. mach ausreichend davon gebrauch. du wirst sehen wie leicht mensch vergisst
- 6: setup-block. ein block beginnt mit { und endet mit }. die anzahl der öffnenden klammern muss genau der anzahl der schliessenden klammern entsprechen.
- 7: konfiguriert den spezifizierten pin als input oder output. der pin an welchem die rote LED angeschlossen ist, wird als ausgabe gesetzt.

- 8: die variable grueneLED hat einen wert von 8. am pin 8 ist unsere grüne LED angeschlossen
- 9: die grüne LED wird eingeschaltet. digitalWrite hat 2 parameter: die pinnummer und [HIGH/LOW]. HIGH bedeutet, dass 5Volt gesetzt werden. dadurch kann die LED leuchten. bei LOW, wird die spannung auf 0Volt gesetzt und die LED ist somit aus.
- 14: loop-block beginnt und endet in zeile 19. mit der tastenkombination STRG+T kannst du die richtigen einrückungen erzeugen. dies erleichtert das lesen eines programms sehr. später wirst du programme schreiben, welche verschachtelte logische blöcke enthalten
- 16: unterbricht das programm für die als parameter angegebene zeit (in millisekunden). (es gibt 1000 millisekunden in einer sekunde.)
- im loop: roteLED wird eingeschaltet ==> 1000msec wird gewartet ==> roteLED aus ==> 500msec warten ==> roteLED an ==> warten & immer so weiter. bis der arduino keinen strom mehr hat oder du ein anderes programm rauflädtst

02. Blinken

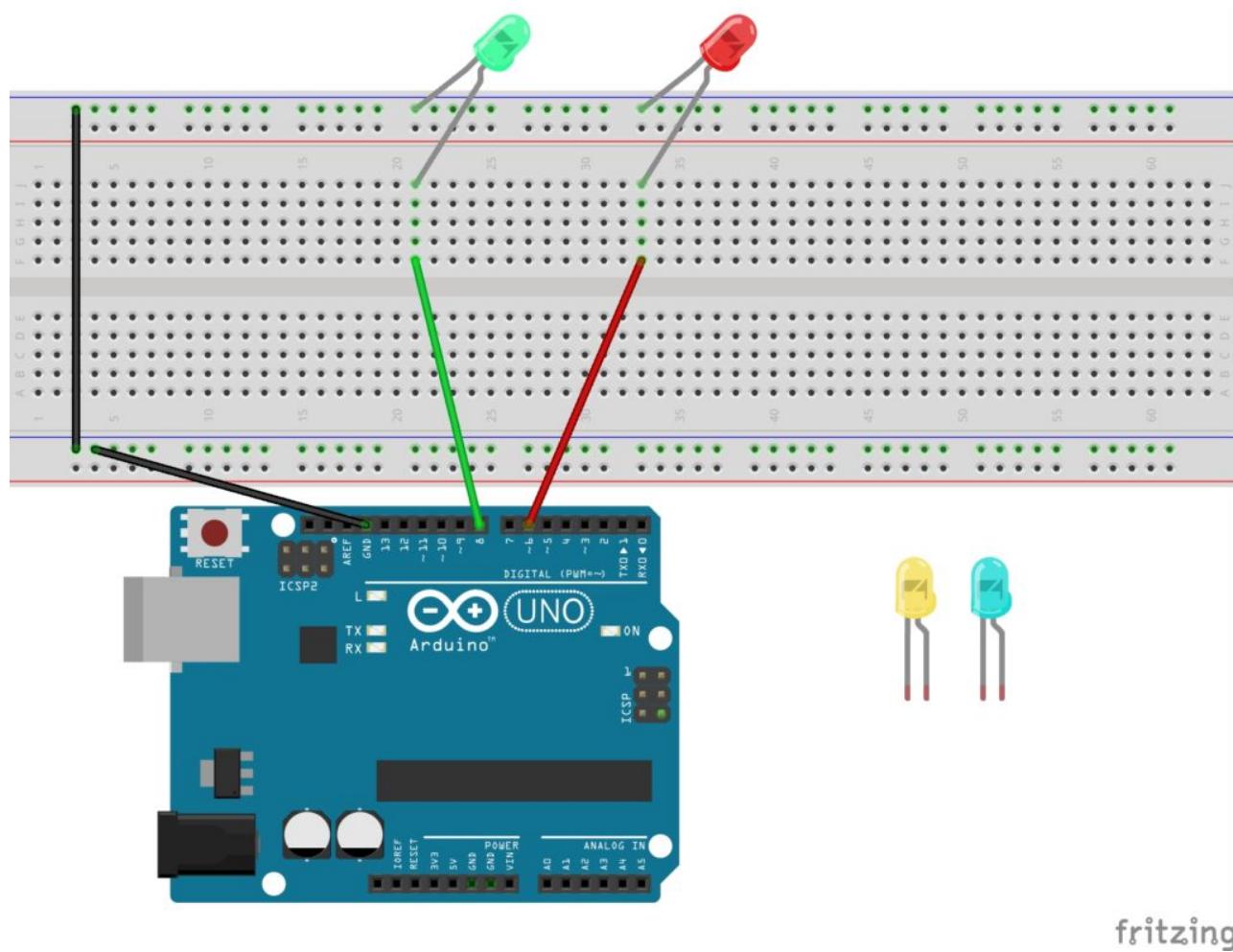
by rainer - Donnerstag, Januar 28, 2021

<https://arduino.langheiter.at/02-blinken/>

folgende anweisungen/befehle werden verwendet:

- [int](#)
- [pinMode](#)
- [digitalWrite](#)
- [delay](#)

du weisst nun, dass ein arduino programm



du sprichst wahrscheinlich deutsch und eventuell noch eine andere sprache. der arduino versteht die programmiersprache [c++](#). c++ existiert seit dem jahre 1979. die älteste noch verwendete programmiersprache heisst [fortran](#) und wurde 1954 entworfen. [hier findest du mehr infos](#) zur geschichte der programmiersprachen.

lass uns das programm zeile für zeile durchgehen:

- 1: hier wird die variable **roteLED** deklariert. das deklarieren einer variable bedeutet, dass ihr typ definiert wird und optional ein anfangswert gesetzt wird. der type int sagt aus, dass wir eine ganzzahl im bereich -2.147.483.648 bis 2.147.483.647 darin speichern können. der anfangswert ist 6, da unsere rote LED mit dem pin 6 verbunden ist.
- 2: **grueneLED** wird deklariert und erhält den anfangswert 8
- 3: leerzeile damit wir das programm leichter lesen können
- 4: // dies bedeutet: kommentar, anmerkung. mach ausreichend davon gebrauch. du wirst sehen wie leicht mensch vergisst
- 6: setup-block. ein block beginnt mit { und endet mit }. die anzahl der öffnenden klammern muss genau der anzahl der schliessenden klammern entsprechen.
- 7: konfiguriert den spezifizierten pin als input oder output. der pin an welchem die rote LED angeschlossen ist, wird als ausgabe gesetzt.
- 8: die variable **grueneLED** hat einen wert von 8. am pin 8 ist unsere grüne LED angeschlossen
- 9: die grüne LED wird eingeschaltet. **digitalWrite** hat 2 parameter: die pinnummer und [HIGH/LOW]. HIGH bedeutet, dass 5Volt gesetzt werden. dadurch kann die LED leuchten. bei LOW, wird die spannung auf 0Volt gesetzt und die LED ist somit aus.
- 14: loop-block beginnt und endet in zeile 19. mit der tastenkombination STRG+T kannst du die richtigen einrückungen erzeugen. dies erleichtert das lesen eines programms sehr. später wirst du programme schreiben, welche verschachtelte logische blöcke enthalten
- 16: unterbricht das programm für die als parameter angegebene zeit (in millisekunden). (es gibt 1000 millisekunden in einer sekunde.)
- im loop: **roteLED** wird eingeschaltet ==> 1000msec wird gewartet ==> **roteLED** aus ==> 500msec warten ==> **roteLED** an ==> warten & immer so weiter. bis der arduino keinen strom mehr hat oder du ein anderes programm rauflädtst

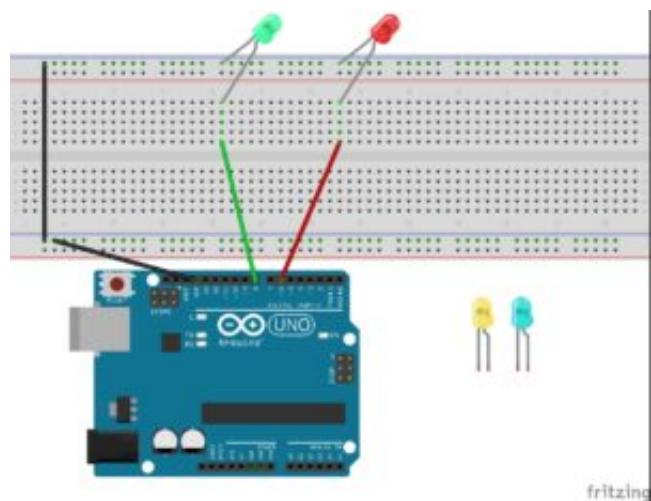
03. mehr blinken

by rainer - Donnerstag, Januar 28, 2021

<https://arduino.langheiter.at/int/>

folgende anweisungen/befehle werden verwendet:

- [int](#)
- [pinMode](#)
- [digitalWrite](#)
- [delay](#)



du sprichst wahrscheinlich deutsch und eventuell noch eine andere sprache. der arduino versteht die programmiersprache [c++](#). c++ existiert seit dem jahre 1979. die älteste noch verwendete programmiersprache heisst [fortran](#) und wurde 1954 entworfen. [hier findest du mehr infos](#) zur geschichte der programmiersprachen.

lass uns das programm zeile für zeile durchgehen:

- 1: hier wird die variable **roteLED** deklariert. das deklarieren einer variable bedeutet, dass ihr typ definiert wird und optional ein anfangswert gesetzt wird. der type int sagt aus, dass wir eine ganzzahl im bereich -2.147.483.648 bis 2.147.483.647 darin speichern können. der anfangswert ist 6, da unsere rote LED mit dem pin 6 verbunden ist.
- 2: grueneLED wird deklariert und erhält den anfangswert 8
- 3: leerzeile damit wir das programm leichter lesen können
- 4: // dies bedeutet: kommentar, anmerkung. mach ausreichend davon gebrauch. du wirst sehen wie leicht mensch vergisst
- 6: setup-block. ein block beginnt mit { und endet mit }. die anzahl der öffnenden klammern muss genau der anzahl der schliessenden klammern entsprechen.
- 7: konfiguriert den spezifizierten pin als input oder output. der pin an welchem die rote LED angeschlossen ist, wird als ausgabe gesetzt.

- 8: die variable grueneLED hat einen wert von 8. am pin 8 ist unsere grüne LED angeschlossen
- 9: die grüne LED wird eingeschaltet. digitalWrite hat 2 parameter: die pinnummer und [HIGH/LOW]. HIGH bedeutet, dass 5Volt gesetzt werden. dadurch kann die LED leuchten. bei LOW, wird die spannung auf 0Volt gesetzt und die LED ist somit aus.
- 14: loop-block beginnt und endet in zeile 19. mit der tastenkombination STRG+T kannst du die richtigen einrückungen erzeugen. dies erleichtert das lesen eines programms sehr. später wirst du programme schreiben, welche verschachtelte logische blöcke enthalten
- 16: unterbricht das programm für die als parameter angegebene zeit (in millisekunden). (es gibt 1000 millisekunden in einer sekunde.)
- im loop: roteLED wird eingeschaltet ==> 1000msec wird gewartet ==> roteLED aus ==> 500msec warten ==> roteLED an ==> warten & immer so weiter. bis der arduino keinen strom mehr hat oder du ein anderes programm rauflädtst

04.

by rainer - Donnerstag, Februar 04, 2021

<https://arduino.langheiter.at/04-2/>

05. TABs

by rainer - Freitag, Januar 29, 2021

<https://arduino.langheiter.at/05-2/>

- [pinMode](#)
- [analogRead](#)
- [digitalWrite](#)
- [delay](#)
- [map](#)
- [for](#)

06. fotowiderstand

by rainer - Donnerstag, Februar 04, 2021

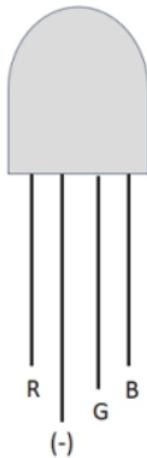
<https://arduino.langheiter.at/06-fotowiderstand/>

- [pinMode](#)
- [analogRead](#)
- [digitalWrite](#)
- [delay](#)
- [map](#)
- [for](#)

07. RGB-LEDs

by rainer - Donnerstag, Februar 04, 2021

<https://arduino.langheiter.at/07-2/>



RGB steht für Rot Grün Blau. Eine RGB-LED verwendet drei Leuchtdioden in den Farben Rot, Grün und Blau und mischt diese in unterschiedlichen Intensitäten zu verschiedenfarbigem Licht.

wir wollen über die serielle schnittstelle eine eingabe realisieren und unsere RGB-LED in unterschiedlichen farben leuchten lassen.

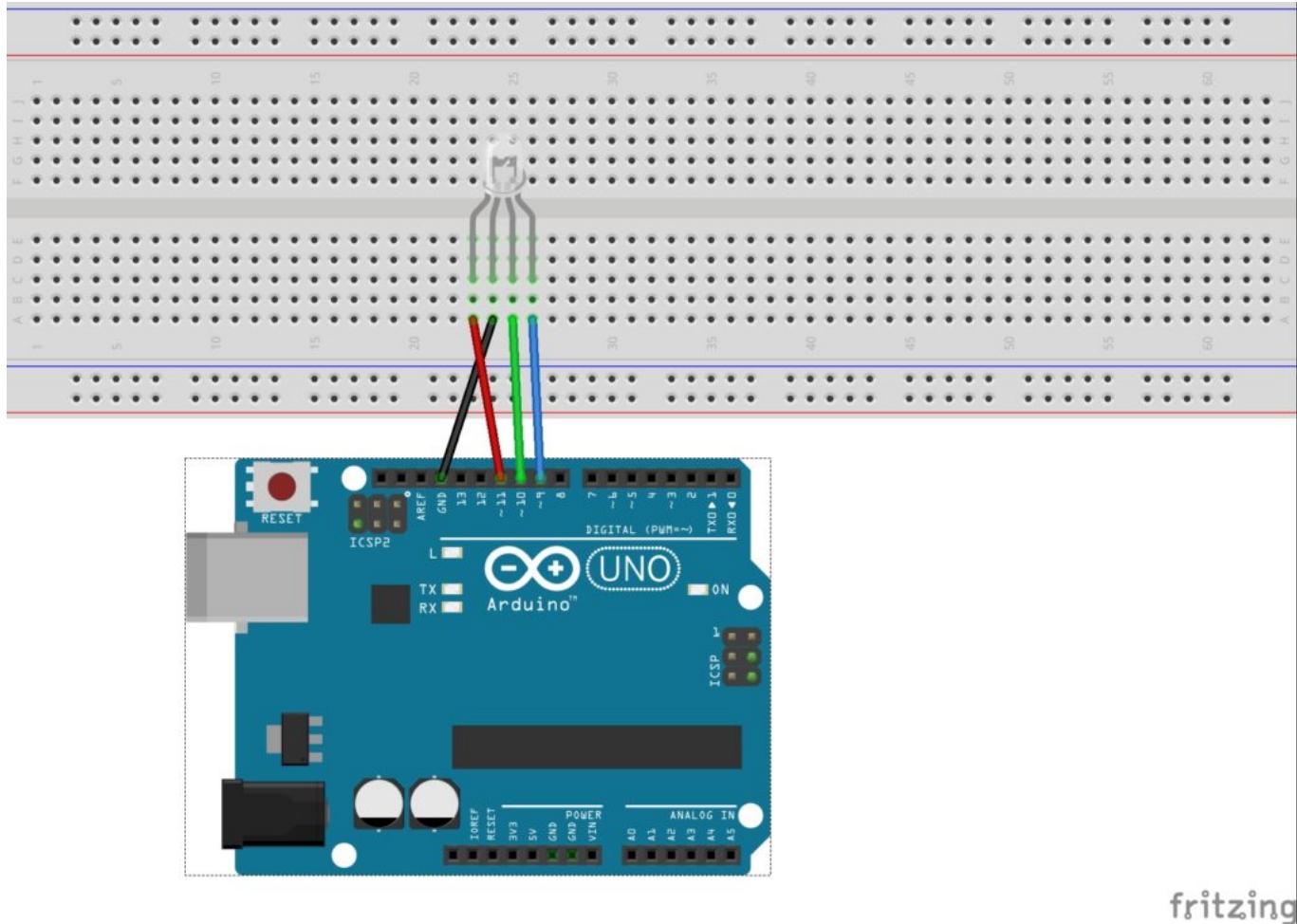
beispieleingabe über serielle console: 1,150 == unsere RGB-LED leuchtet dadurch intensiv grün gib folgendes ein:

0,1

1,1

2,1

dadurch werden alle 3 farben der RGB-LED ganz wenig zum leuchten gebracht und du kannst die 3 leuchtpunkte erkennen



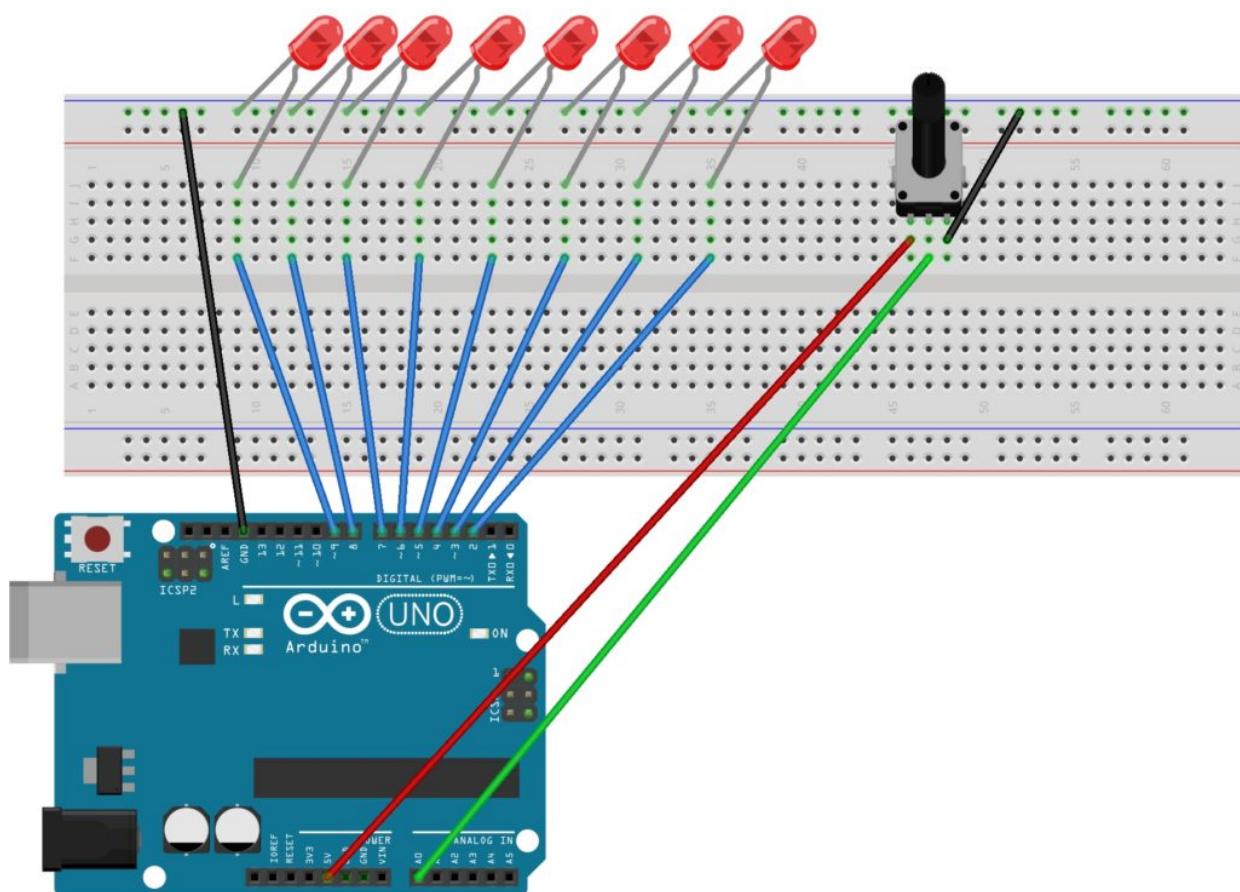
08. lauflicht

by rainer - Donnerstag, Februar 04, 2021

<https://arduino.langheiter.at/08-lauflicht/>

folgende Anweisungen/Befehle werden verwendet:

- pinMode
- analogRead
- digitalWrite
- delay
- map
- for



fritzing

schreibe

ein Programm welches die LEDs von links nach rechts einzeln ein und ausschaltet und dann wieder retour

[rider-01](#)

je nach potentiometer stellung bewegt sich das lauflicht schnell oder langsam

lauflicht

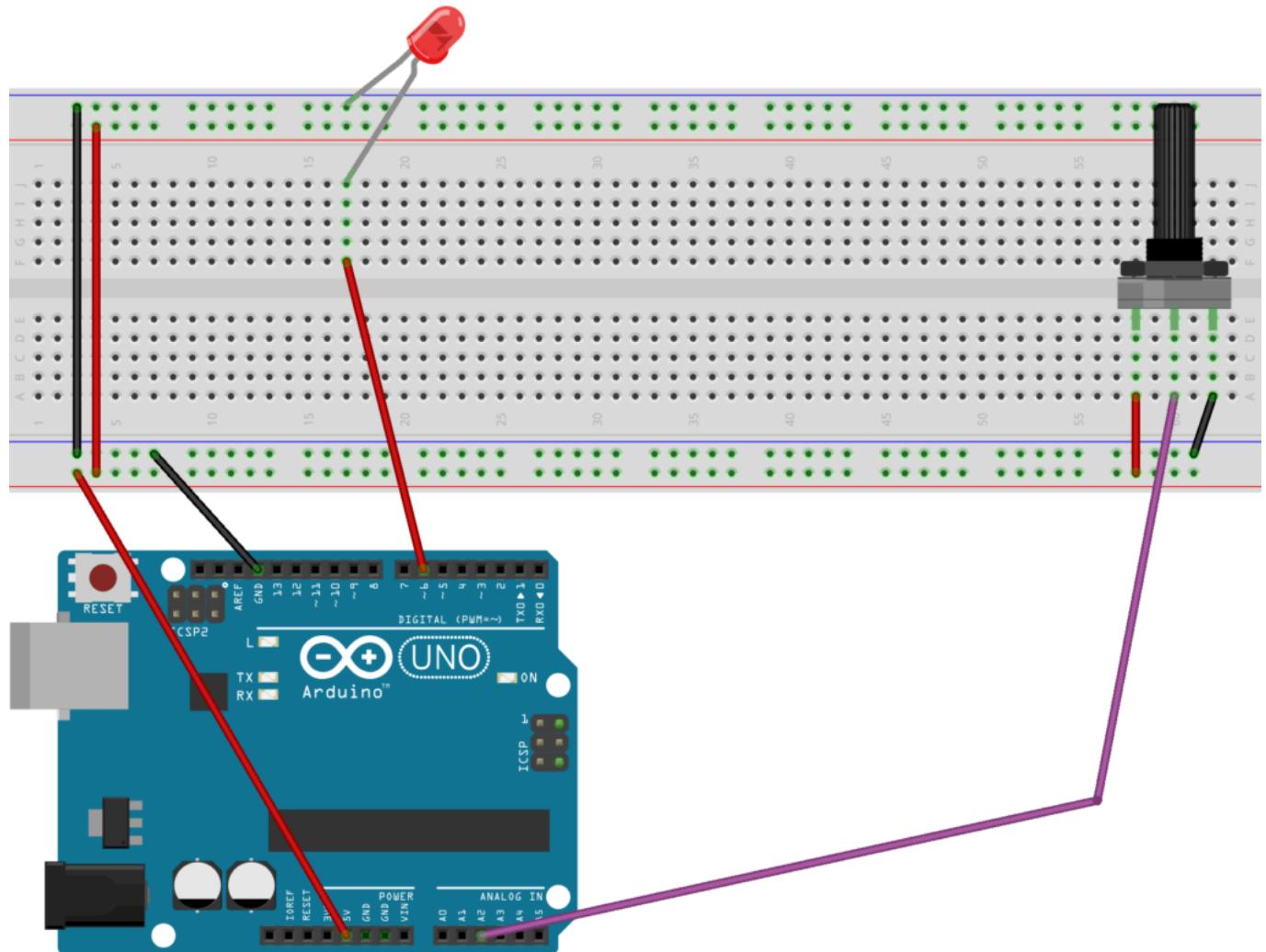
09.Potentiometer01

by rainer - Donnerstag, Januar 28, 2021

<https://arduino.langheiter.at/potentiometer/>

folgende anweisungen/befehle werden verwendet:

- [Serial.begin](#)
- [pinMode](#)
- [analogRead](#)
- [Serial.println](#)
- [digitalWrite](#)
- [delay](#)



fritzing

10.

by rainer - Donnerstag, Februar 04, 2021

<https://arduino.langheiter.at/10-2/>

11. Potentiometer03

by rainer - Freitag, Januar 29, 2021

<https://arduino.langheiter.at/11-potentiometer03/>

4.1

by rainer - Dienstag, Dezember 21, 2021

<https://arduino.langheiter.at/4-1/>

4.6 windrad

by rainer - Dienstag, Dezember 21, 2021

<https://arduino.langheiter.at/4-6/>

Wie andere hohe Masten müssen auch Windkraftanlagen ein rotes Blinklicht an ihrer Spitze haben, um Flugzeuge zu warnen. Da die Spitze des Mastes bei einem Windrad aber nicht wirklich die Spitze der Gefahrenzone ist, blinkt ihr rotes Licht in einem besonderen Rhythmus: 1 Sekunde an, dann 0,5 Sekunden aus, dann wieder 1 Sekunde an, dann aber 1,5 Sekunden aus.

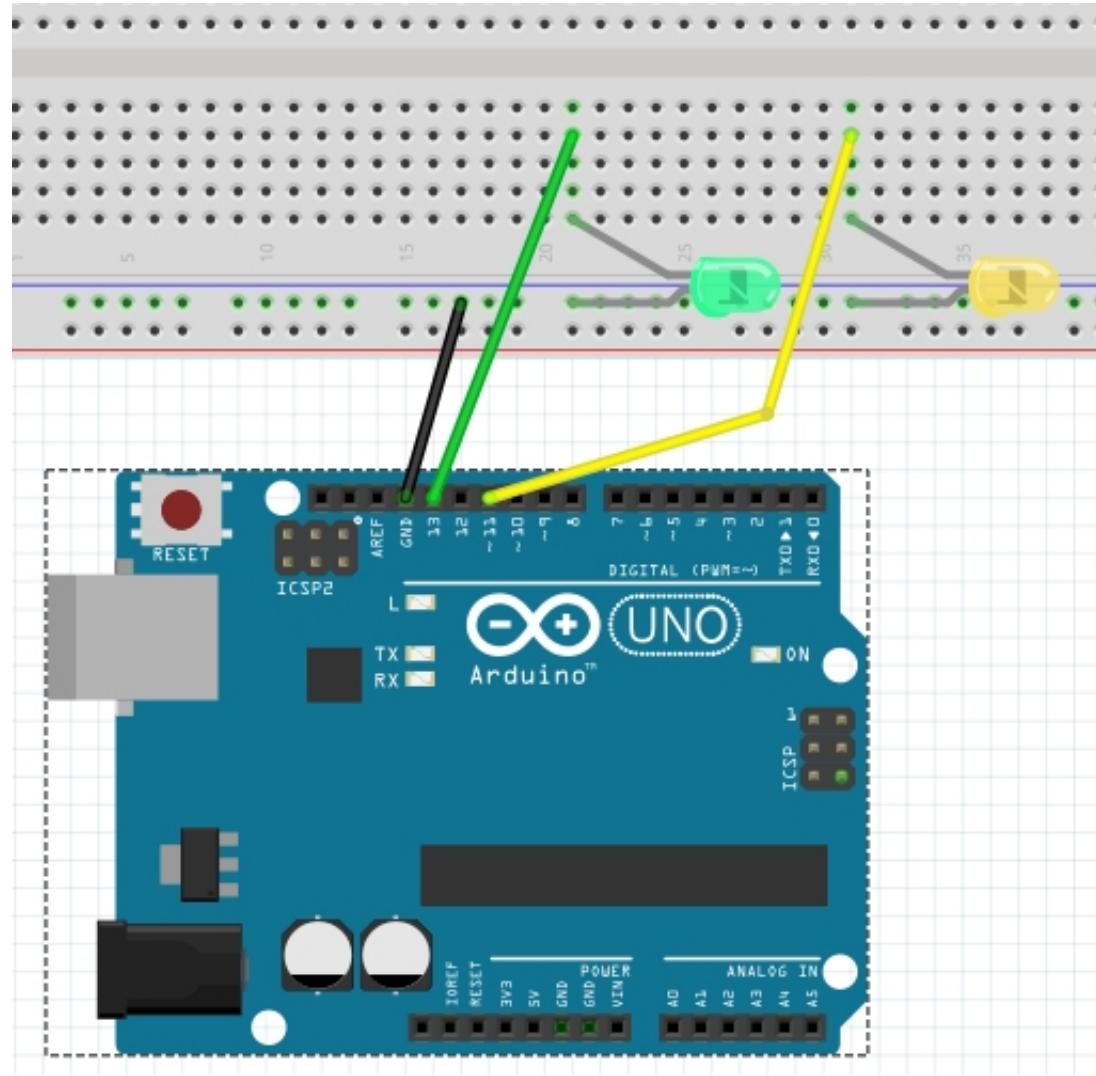
Programmiere so ein Dauerblinklicht und speichere es unter Aufgabe_4_6_Windrad.

4.7 wechselblinker

by rainer - Mittwoch, Dezember 22, 2021

<https://arduino.langheiter.at/4-7-wechselblinker/>

Schließe eine zweite LED an den Arduino an. Lass beide LEDs abwechselnd blinken.

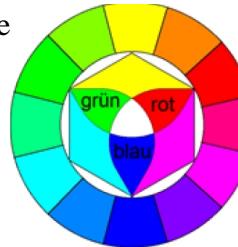


5.1 RGB-LED

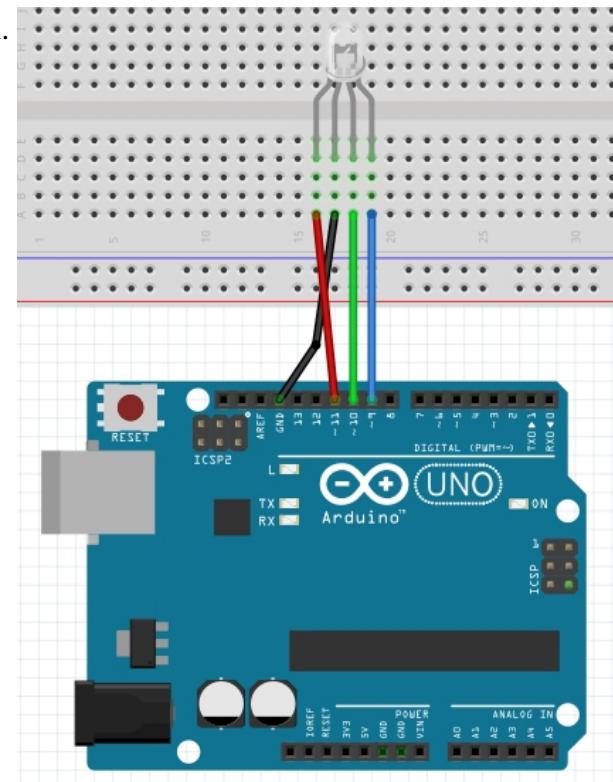
by rainer - Mittwoch, Dezember 22, 2021

<https://arduino.langheiter.at/5-1/>

1. Baue die Schaltung von unten nach. Stülpe



einen Tischtennisball über die RGB-LED, um die Farben besser erkennen zu können.



Lass die RGB-LED nacheinander in den Grundfarben rot, grün und blau leuchten.

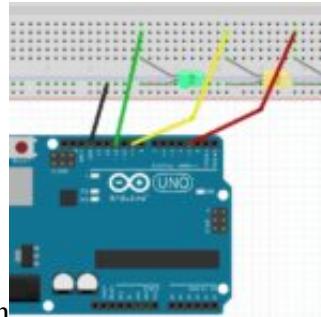
Hinweis: Im setup-Teil benötigst du für jeden der drei LED-Anschlüsse ein pinMode(...,OUTPUT)

2. Mit der RGB-LED lassen sich auch andere Farbeindrücke erzeugen. Lasse die RGB-LED gelb leuchten und notiere alle Farbeindrücke, die auch noch möglich sind. Tipp: Additive Farbmischung (der Farbkreis unten rechts hilft dir weiter).
3. Stell dir vor, du hättest eine LED mit vier verschiedenen Farben. Wie viele verschiedene Farbeindrücke sind hier möglich? Kann man das auch schnell berechnen?

7.1 ampel

by rainer - Sonntag, Dezember 26, 2021

<https://arduino.langheiter.at/7-1-ampel/>

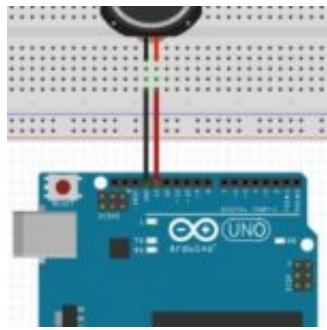


versuche eine ampelsteuerung zu programmieren

8.1 ton

by rainer - Freitag, Dezember 24, 2021

<https://arduino.langheiter.at/8-1-ton/>



9.1 serieller monitor

by rainer - Donnerstag, Dezember 23, 2021

<https://arduino.langheiter.at/9-1-serieller-monitor/>

In verschiedenen Situationen kann es sinnvoll sein, dass der Mikrocontroller eine Rückmeldung auf einen Ausgabebildschirm gibt (Beispiele: bei der Ampel: „Stopp“ und „Gehen“; beim Lautsprecher: Liedtitel angeben; bei Sensoren: Messwerte anzeigen). Oft wird ein Ausgabebildschirm auch für die Fehlersuche in Programmen eingesetzt: Wenn an bestimmten Stellen im Programm eine Rückmeldung einprogrammiert war, wusste man bei Erscheinen des Textes, dass das Programm bis zu dieser Stelle fehlerfrei durchgelaufen war.

Der Arduino kann über den seriellen Monitor Daten, die vom Arduino gesammelt oder produziert werden, anzeigen. Schließe dazu an Port D11 eine LED mit Vorwiderstand an und übernehme folgendes Programm:

9.2 :: gesicht

by rainer - Donnerstag, Dezember 23, 2021

<https://arduino.langheiter.at/9-2-gesicht/>

```
— ~~~ —  
x     x  
o  
l ___ l
```

Schreibe ein Programm, das ein „Gesicht“ auf dem Ausgabebildschirm erzeugt:

Kannst du das Gesicht nur einmal anzeigen?

9.3 :: befehle

by rainer - Freitag, Dezember 24, 2021

<https://arduino.langheiter.at/9-3-befehle/>

hier die ersten 10 gelernten befehle.

schreib für jeden eine vollständige ausgabe

zb: *wert = analogRead(D11);*

hier findest du alle möglichen befehle: <https://www.arduino.cc/reference/de/>

Bauteile

by rainer - Donnerstag, Januar 28, 2021

<https://arduino.langheiter.at/bauteile/>

die wichtigsten bauteile in aller kürze.

Beispiel-Seite

by rainer - Donnerstag, Januar 28, 2021

<https://arduino.langheiter.at/beispiel-seite/>

Dies ist eine Beispiel-Seite. Sie unterscheidet sich von Beiträgen, da sie stets an derselben Stelle bleibt und (bei den meisten Themes) in der Website-Navigation angezeigt wird. Die meisten starten mit einem Impressum, der Datenschutzerklärung oder einer „Über uns“-Seite, um sich potenziellen Besuchern der Website vorzustellen. Dort könnte zum Beispiel stehen:

Hallo! Tagsüber arbeite ich als Fahrradkurier, nachts bin ich ein aufstrebender Schauspieler und dies hier ist meine Website. Ich lebe in Berlin, habe einen großen Hund namens Jack, mag Piña Coladas, jedoch weniger (ohne Schirm) im Regen stehen gelassen zu werden.

...oder so etwas wie das hier:

Das Unternehmen XYZ wurde 1971 gegründet und versorgt die Öffentlichkeit seither mit qualitativ hochwertigen Produkten. An seinem Standort in einer kleinen Großstadt beschäftigt der Betrieb über 2.000 Menschen und unterstützt die Stadtbewohner in vielfacher Hinsicht.

Als neuer WordPress-Benutzer solltest du [dein Dashboard](#) aufrufen, um diese Seite zu löschen und neue Seiten und Beiträge für deine Website erstellen. Viel Spaß!

Datenschutzerklärung

by rainer - Donnerstag, Januar 28, 2021

https://arduino.langheiter.at/?page_id=3

Wer wir sind

Die Adresse unserer Website ist: <https://arduino.langheiter.at>.

Welche personenbezogenen Daten wir sammeln und warum wir sie sammeln

Kommentare

Wenn Besucher Kommentare auf der Website schreiben, sammeln wir die Daten, die im Kommentar-Formular angezeigt werden, außerdem die IP-Adresse des Besuchers und den User-Agent-String (damit wird der Browser identifiziert), um die Erkennung von Spam zu unterstützen.

Aus deiner E-Mail-Adresse kann eine anonymisierte Zeichenfolge erstellt (auch Hash genannt) und dem Gravatar-Dienst übergeben werden, um zu prüfen, ob du diesen benutzt. Die Datenschutzerklärung des Gravatar-Dienstes findest du hier: <https://automattic.com/privacy/>. Nachdem dein Kommentar freigegeben wurde, ist dein Profilbild öffentlich im Kontext deines Kommentars sichtbar.

Medien

Wenn du ein registrierter Benutzer bist und Fotos auf diese Website lädst, solltest du vermeiden, Fotos mit einem EXIF-GPS-Standort hochzuladen. Besucher dieser Website könnten Fotos, die auf dieser Website gespeichert sind, herunterladen und deren Standort-Informationen extrahieren.

Kontaktformulare

Cookies

Wenn du einen Kommentar auf unserer Website schreibst, kann das eine Einwilligung sein, deinen Namen, E-Mail-Adresse und Website in Cookies zu speichern. Dies ist eine Komfortfunktion, damit du nicht, wenn du einen weiteren Kommentar schreibst, all diese Daten erneut eingeben musst. Diese Cookies werden ein Jahr lang gespeichert.

Falls du ein Konto hast und dich auf dieser Website anmeldest, werden wir ein temporäres Cookie setzen, um festzustellen, ob dein Browser Cookies akzeptiert. Dieses Cookie enthält keine personenbezogenen Daten und wird verworfen, wenn du deinen Browser schließt.

Wenn du dich anmeldest, werden wir einige Cookies einrichten, um deine Anmeldeinformationen und Anzeigeoptionen zu speichern. Anmelde-Cookies verfallen nach zwei Tagen und Cookies für die Anzeigeoptionen nach einem Jahr. Falls du bei der Anmeldung „Angemeldet bleiben“ auswählst, wird deine Anmeldung zwei Wochen lang aufrechterhalten. Mit der Abmeldung aus deinem Konto werden die Anmelde-Cookies gelöscht.

Wenn du einen Artikel bearbeitest oder veröffentlichtest, wird ein zusätzlicher Cookie in deinem Browser gespeichert. Dieser Cookie enthält keine personenbezogenen Daten und verweist nur auf die Beitrags-ID des Artikels, den du gerade bearbeitet hast. Der Cookie verfällt nach einem Tag.

Eingebettete Inhalte von anderen Websites

Beiträge auf dieser Website können eingebettete Inhalte beinhalten (z. B. Videos, Bilder, Beiträge etc.). Eingebettete Inhalte von anderen Websites verhalten sich exakt so, als ob der Besucher die andere Website besucht hätte.

Diese Websites können Daten über dich sammeln, Cookies benutzen, zusätzliche Tracking-Dienste von Dritten einbetten und deine Interaktion mit diesem eingebetteten Inhalt aufzeichnen, inklusive deiner Interaktion mit dem eingebetteten Inhalt, falls du ein Konto hast und auf dieser Website angemeldet bist.

Analysedienste

Mit wem wir deine Daten teilen

Wenn du eine Zurücksetzung des Passworts beantragst, wird deine IP-Adresse in der E-Mail zur Zurücksetzung enthalten sein.

Wie lange wir deine Daten speichern

Wenn du einen Kommentar schreibst, wird dieser inklusive Metadaten zeitlich unbegrenzt gespeichert. Auf diese Art können wir Folgekommentare automatisch erkennen und freigeben, anstatt sie in einer Moderations-Warteschlange festzuhalten.

Für Benutzer, die sich auf unserer Website registrieren, speichern wir zusätzlich die persönlichen Informationen, die sie in ihren Benutzerprofilen angeben. Alle Benutzer können jederzeit ihre persönlichen Informationen einsehen, verändern oder löschen (der Benutzername kann nicht verändert werden). Administratoren der Website können diese Informationen ebenfalls einsehen und verändern.

Welche Rechte du an deinen Daten hast

Wenn du ein Konto auf dieser Website besitzt oder Kommentare geschrieben hast, kannst du einen Export deiner personenbezogenen Daten bei uns anfordern, inklusive aller Daten, die du uns mitgeteilt hast. Darüber hinaus kannst du die Löschung aller personenbezogenen Daten, die wir von dir gespeichert haben, anfordern. Dies umfasst nicht die Daten, die wir aufgrund administrativer, rechtlicher oder sicherheitsrelevanter Notwendigkeiten aufbewahren müssen.

Wohin wir deine Daten senden

Besucher-Kommentare könnten von einem automatisierten Dienst zur Spam-Erkennung untersucht werden.

Deine Kontakt-Informationen

Weitere Informationen

Wie wir deine Daten schützen

Welche Maßnahmen wir bei Datenschutzverletzungen anbieten

Von welchen Drittanbietern wir Daten erhalten

Welche automatisierte Entscheidungsfindung und/oder Profilerstellung wir mit Benutzerdaten durchführen

Vorausgesetzte Offenlegungspflichten der Industrie

arduino.langheiter.at

arduino lernen

PDF generated December 29, 2021 at 11:54 AM by Kalin's PDF Creation Station WordPress plugin