

## Station 3 – Lichtschranke

### Unsichtbares Licht

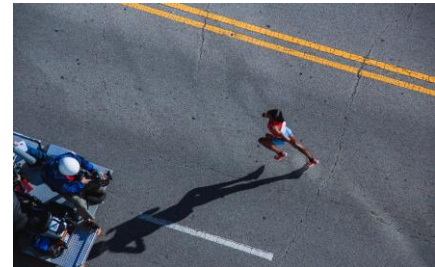
Lichtschranken begegnen euch überall im Alltag, z. B. in Aufzügen oder bei automatischen Türklingeln in Geschäften. Auch wenn ihr ihnen ständig über den Weg lauft, merkt ihr es häufig nicht, denn sie arbeiten mit unsichtbarem Licht: dem **Infrarot-Licht (kurz IR-Licht)**.



[1]



[2]



[3]

### Der Aufbau der Schaltung

Die IR-Lichtschranke besteht aus zwei Elementen: einem **Sender** und einem **Empfänger**. Der Sender, hier eine IR-LED, sendet Licht aus, und der Empfänger, hier eine IR-Photodiode, „sieht“ genau dieses Licht. Wenn die IR-Photodiode IR-Licht registriert, leitet sie Strom. Ist die Sichtlinie unterbrochen, fließt kein Strom. Ihr braucht folgende Bauteile (neben Steckbrett und Arduino):

- 1x 220  $\Omega$  Widerstand
- 1x 100 k $\Omega$  Widerstand
- 1x IR-Photodiode
- 1x IR-LED
- 1 gelbes, 2 blaue und 2 rote lange Steckkabel



[4]



[5]

Prüft, ob ihr alle Bauteile habt. Falls euch ein Bauteil fehlt, meldet euch bei den Betreuern!

#### IR-LEDs und IR-Photodioden

**IR-LEDs** funktionieren im Grunde wie ganz normale LEDs. Allerdings haben sie einen kleinen Nachteil: Mit bloßem Auge sieht man nicht, ob sie an oder aus sind.

Eine **IR-Photodiode** ist so aufgebaut, dass sie nur in **einer** Richtung Strom leitet, wenn sie mit IR-Licht beschienen wird. In der anderen Richtung leitet sie **immer** Strom.

Sowohl IR-LED als auch IR-Photodiode können nur in einem sehr kleinen Winkel Licht ausstrahlen bzw. aufnehmen.

## Station 3 – Lichtschranke

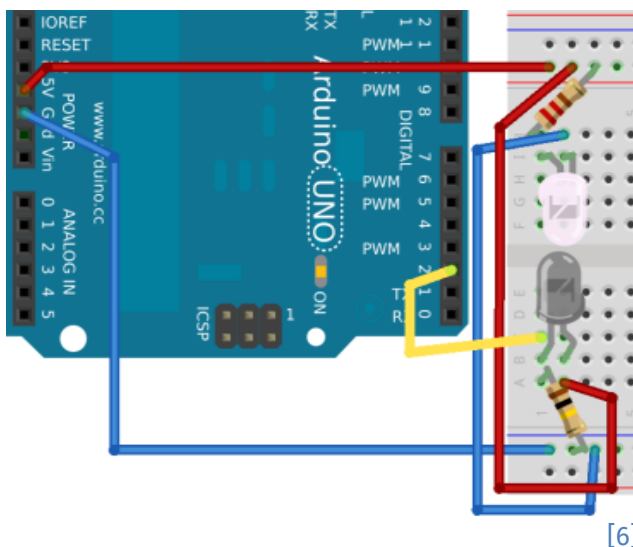
**IR-LED**

Bei der IR-LED sind die Beinchen vertauscht. Deshalb müsst ihr darauf achten, dass diesmal das **kurze** Beinchen an die **Plus**-Leiste angeschlossen ist.

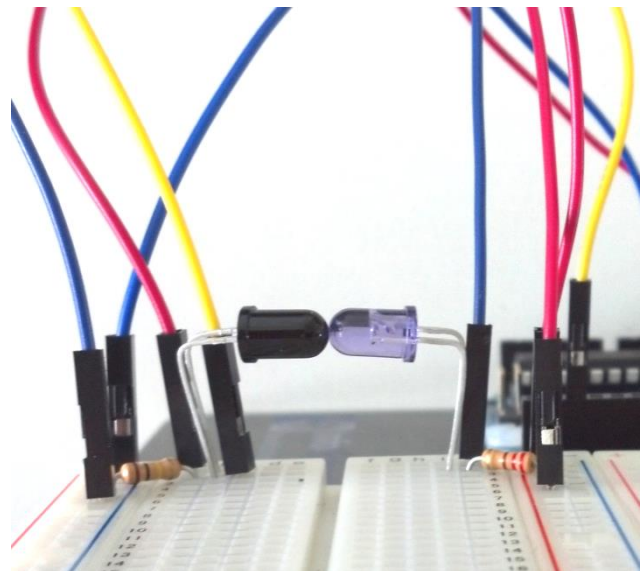


Jetzt könnt ihr mit dem Zusammenbau starten! Auf den Bildern unten seht ihr, wie die Schaltung am Ende aussehen soll.

1. Verbindet zunächst die **Plus**-Leiste mit dem **5V**-Anschluss, die **Minus**-Leiste mit einem **GND**-Pin.
2. Steckt die **IR-LED** und die **IR-Photodiode** ins Steckbrett (wie in der Abbildung unten).
3. Verbindet die IR-LED also mit dem **220  $\Omega$  Widerstand** und je einem Kabel mit der **Plus**- und **Minus**-Leiste.
4. Testet die IR-LED wie oben beschrieben.
5. Die IR-Photodiode wird fast genauso angeschlossen. Das kurze Beinchen wird mit der **Plus**-Leiste verbunden. Das lange Beinchen wird mit dem **digitalen Pin 2** auf dem Arduino verbunden. Zusätzlich wird dieses Beinchen mit dem **100 k $\Omega$  Widerstand** mit der **Minus**-Leiste verbunden.
6. Biegt beide Bauteile so, dass sie sich gegenseitig „angucken“ (wie in der Abbildung).



[6]

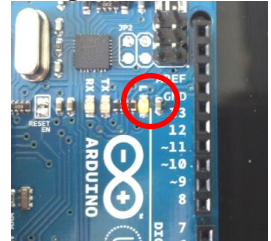


[7]

## Station 3 – Lichtschranke

## Der erste Test

Gut, eure Schaltung steht. Nun geht es an die Programmierung! Ihr habt bisher zwar keine Möglichkeit etwas anzuzeigen, aber hier haben die Arduino-Macher zum Glück mitgedacht. Der **Pin 13** ist mit einer **LED** auf dem Arduino intern verbunden, wird Pin 13 eingeschaltet (also auf **HIGH** gesetzt), leuchtet die LED.



Eure Aufgabe ist jetzt, die LED genau dann einzuschalten, wenn die Lichtschranke unterbrochen wird.

[8]



1. Erstellt ein neues Projekt in ArduBlock und speichert es unter einem sinnvollen Namen.
2. Als Erstes bringt ihr die Arduino-LED zum Blinken. Das kennt ihr bereits aus dem Einstiegsprojekt. Dazu braucht ihr die beiden Bausteine **digitalWrite** und **warteMillisekunden**. Denkt daran, dass der Wert eurer LED-Variablen **HIGH** ist, wenn die LED leuchtet und sonst **LOW**.
3. Testet euer Programm. Blinkt die LED? Super! Im nächsten Schritt lernt ihr, wie ihr die Lichtschranke testen könnt.

## Eingangs-Pin lesen

Einen Eingangs-Pin kann man mit dem Befehl **digitalRead** auslesen. Hinter die Raute (#) setzt man dabei den Pin der IR-Photodiode. So wird ausgelesen, ob Strom durch die Diode fließt oder nicht.

Jetzt wird die Lichtschranke getestet. Ihr wollt die LED natürlich nicht immer einschalten, sondern nur, wenn die Lichtschranke unterbrochen ist. Dafür braucht ihr einen **falls/sonst**-Block, der für euch ja nichts Neues ist.



1. Ihr könnt **digitalRead** direkt in **teste** benutzen, um zu überprüfen, ob an der IR-Photodiode Strom ankommt (**HIGH**) oder nicht (**LOW**):



[9]



[10]

In die Lücke fügt ihr dann den Block mit der passenden Zahl für den digitalen Pin ein, an dem ihr die IR-Photodiode angeschlossen habt. Wenn ihr euch an die Abbildung auf der letzten Seite gehalten habt, dann wäre das also die 2.

*Weiter geht's auf der nächsten Seite...*

## Station 3 – Lichtschranke



2. Wenn der digitale Eingang ein **LOW** ausspuckt, wird die LED eingeschaltet. In den **falls/sonst**-Blöcken könnt ihr festlegen, was passieren soll. Dazu müsst ihr nun eure Blöcke, die die Arduino-LED blinken lassen, an die richtigen Stellen verschieben.
3. Entfernt alle **warteMillisekunden**, die noch übrig sind. Diese benötigt ihr jetzt nicht mehr.
4. Es ist an der Zeit, euer Programm zu testen. Benutzt dafür einen schmalen Gegenstand (z. B. ein Blatt Papier).

*Super! Jetzt wird euch über die LED angezeigt, ob eure Lichtschranke unterbrochen wurde.*




#### Quellenverzeichnis:

**Abb. 1** – Quelle: pxhere.com (<https://pxhere.com/de/photo/981592>), CC0 1.0 Universal (CC0 1.) Public Domain Dedication (<https://creativecommons.org/publicdomain/zero/1.0/>), abgerufen am: 06.07.2022.

**Abb. 2** – Quelle: pxhere.com (<https://pxhere.com/de/photo/927145>), CC0 1.0 Universal (CC0 1.) Public Domain Dedication (<https://creativecommons.org/publicdomain/zero/1.0/>), abgerufen am: 06.07.2022.

**Abb. 3** – Quelle: pxhere.com (<https://pxhere.com/de/photo/53576>), CC0 1.0 Universal (CC0 1.) Public Domain Dedication (<https://creativecommons.org/publicdomain/zero/1.0/>), abgerufen am: 06.07.2022.

**Abb. 4, 6** – Quelle: Screenshots der Fritzing-Software (<http://fritzing.org>), CC-BY-SA (<https://creativecommons.org/licenses/by-sa/3.0/>), erstellt am 31.01.2023.

**Abb. 5, 7, 8,**  – Quelle: InfoSphere-Team, CC BY-SA 4.0 Attribution-ShareAlike 4.0 International (<https://creativecommons.org/licenses/by-sa/4.0/>).

**Abb. 9, 10** – Quelle: Screenshot der Software ArduBlock, GNU General Public License (<https://github.com/taweili/ardublock/blob/master/LICENSE.txt>), erstellt am 31.01.2023.