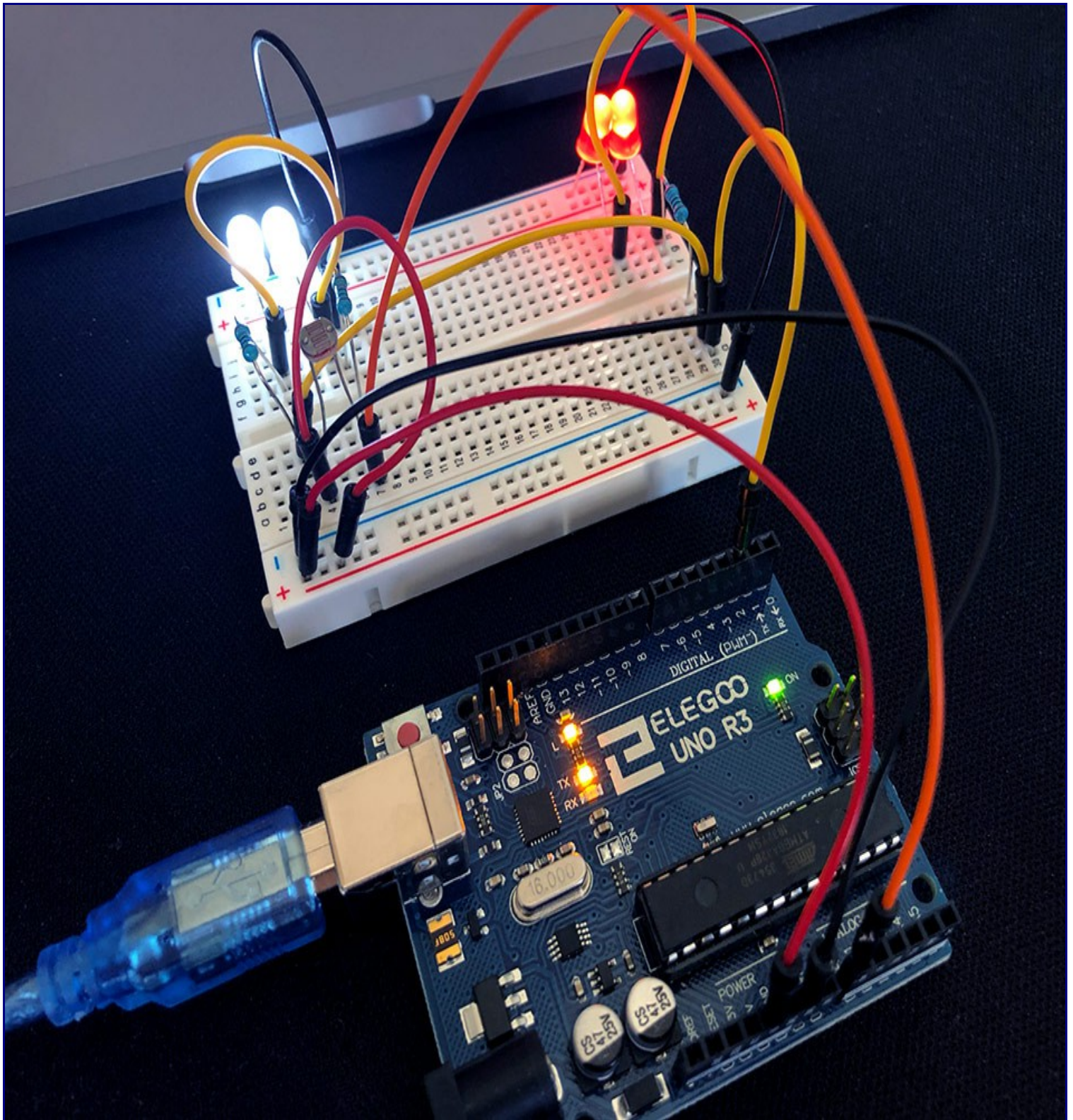


Adaptives Fahrtlicht mit Arduino



Moderne Autos verfügen oft über adaptives Fahrtlicht. Das bedeutet, dass sich das Abblendlicht je nach Lichtverhältnissen automatisch einschaltet. Das lässt sich selbstverständlich gut mit Arduino nachbauen. Dazu benötigt man nur ein paar Teile.



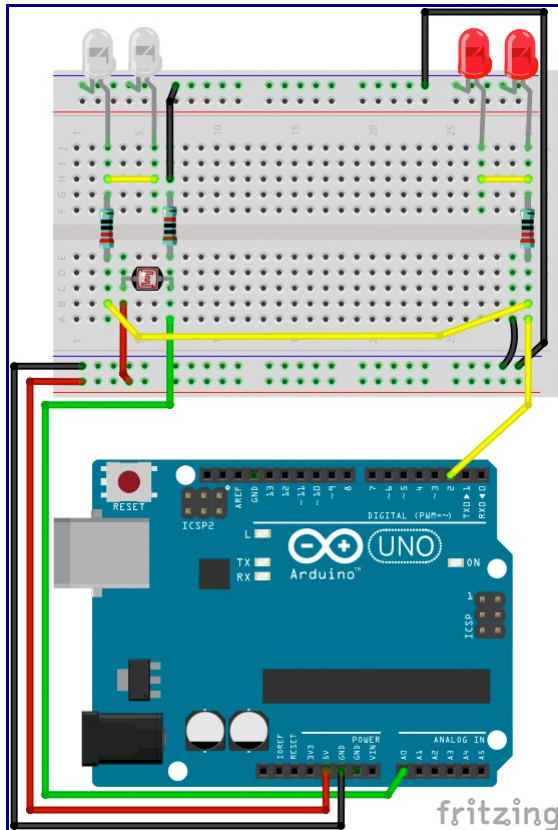
Bauteile

- 1x Arduino-UNO
- 1x LDR (Fotowiderstand)
- 1x 100 kOhm Widerstand

<https://starhardware.org/adaptives-fahrtlicht-mit-arduino/>

- 2x LED rot
- 2x LED weiß
- 1x 220 kOhm Widerstand
- —
- Starterset, das alles enthält: [Elegoo UNO, Breadboard, Kabel*](#)

Schaltplan



Der Schaltplan lässt sich in zwei Schaltkreise unterteilen: der Sensor-Schaltkreis und der Lichtschaltkreis:

Der Lichtschaltkreis besteht aus zwei Parallelschaltungen jeweils gleichfarbiger LEDs (Scheinwerfer und Rücklichter), die jeweils mit einem Widerstand (220 Ohm) in Reihe geschaltet sind. Da immer alle LEDs gleichzeitig an- oder ausgeschaltet werden genügt die Steuerung durch einen digitalen Output vom Arduino. Übrigens benötigen wir diese beiden Parallelschaltungen, da LEDs mit unterschiedlichen Farben unterschiedliche Innenwiderstände besitzen. Würden wir alle vier parallel schalten, würden nur die beiden roten LEDs leuchten. (Ja, das musste ich erst ausprobieren :-))

Der Sensor-Schaltkreis besteht aus einer Spannungsteiler-Schaltung. Hierbei werden zwei Widerstände, nämlich der Fotowiderstand und der 100 kOhm Widerstand in Reihe geschaltet. Eine Seite der Reihenschaltung wird mit dem GND verbunden, die andere mit dem 5V Plus. »Ausgelesen« wird die Schaltung, indem die Verbindung der beiden Widerstände an einen analogen Input-Pin des Arduino-Boards angeschlossen wird.

Erschließung

Sehen wir uns als erstes an, welche Werte der analoge Input-Pin vom Arduino anzeigt. Dazu nutzen wir die serielle Kommunikation.

```
const int ldrPin = A0;
```

<https://starthardware.org/adaptives-fahrtlicht-mit-arduino/>

```

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

void loop() {
  Serial.println(analogRead(ldrPin));
  delay(10);
}

```

Im Seriellen Monitor (*Arduino>Werkzeuge>Seriellen Monitor*) werden die Werte vom Fotowiderstand ausgegeben. Je dunkler es wird, desto kleiner wird der Wert. Nun legen wir einen Wert fest, ab dem das Licht eingeschaltet werden soll. Im Beispiel wird das 900 sein. Dieser Wert wird als Variable *threshold* (engl. Schwelle) gespeichert. Nun lässt sich im Code darauf prüfen.

```

const int ldrPin = A0;
const int ledPin = 2;
const int threshold = 900;

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

void loop() {
  Serial.println(analogRead(ldrPin));
  if (analogRead(ldrPin)<threshold){
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
  delay(10);
}

```

Nun wäre es ziemlich unpraktisch, wenn sich die Lampen so schnell nach Lichtsituation ändern würden. Deshalb bleiben sie beim Auto nach dem Einschalten eine Weile an, auch wenn man z. B. aus einem Tunnel fährt in die pralle Sonne fährt. In unserem Beispiel müssen wir nun also einen Timer einbauen. Das geht so:

```

const int ldrPin = A0;
const int ledPin = 2;
const int threshold = 900;

int einschaltTimer = 0;
int einschaltDauer = 1000;

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

void loop() {
  Serial.println(analogRead(ldrPin));
  if (analogRead(ldrPin)<threshold){
    digitalWrite(ledPin, HIGH);
    einschaltTimer = millis();
  } else {
    if (einschaltTimer+einschaltDauer<millis()){
      digitalWrite(ledPin, LOW);
    }
  }
}

```

<https://starthardware.org/adaptives-fahrtlicht-mit-arduino/>

```
    delay(10);  
}
```

Wenn der Fotowiderstandswert kleiner als die Schaltschwelle (threshold) wird, wird eine »Stopuhr« (timer) gestartet. Das passiert bei jedem Programmdurchlauf erneut, bis die Schaltschwelle größer als der Fotowiderstandswert ist. Nun wird im else-Zweig geprüft, ob der Stopuhr-Start plus der Wartezeit (einschaltDauer) kleiner als die aktuelle Zeit (millis()) ist. Ist das der Fall, wird der ledPin auf LOW gesetzt, was die LEDs abschaltet.