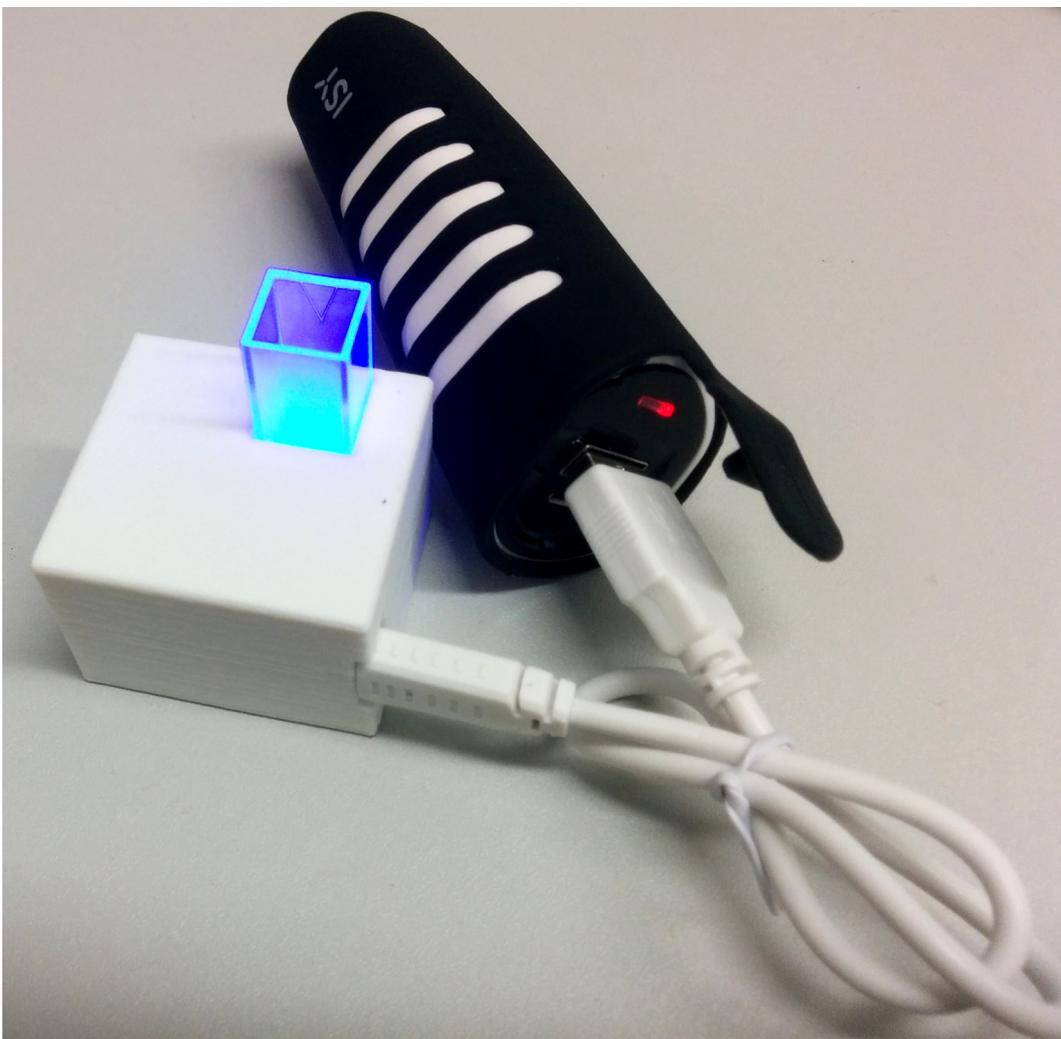


Konzeption und Aufbau eines mobilen Fotometers mit WLAN/ WIFI - Schnittstelle



HAW Hamburg, Labor für Bioprozessautomatisierung

Dipl.-Ing. Ulrich Scheffler

11.11.2020

Ein Projekt von:

Dipl.-Ing. Ulrich Scheffler
HAW Hamburg, Labor für Bioprozessautomatisierung
Ulmenliet 20
21033 Hamburg
email: ulrich.scheffler@haw-hamburg.de

in Kooperation mit dem Schullabor mobile Analytik der HAW Hamburg
email: olaf.elsholz@haw-hamburg.de

Weblink zur Seite mit dieser Anleitung / diesem Projekt:

<https://www.haw-hamburg.de/hochschule/life-sciences/forschung/projekte-schullabore/schullabor-mobile-analytik/smartphone-photometer/>



Sollte ein Nachbau oder eine Variation dieses Projektes erfolgen, freuen wir uns über Rückmeldungen und Hinweise.

Inhaltsverzeichnis

| | |
|--|----|
| 1 Vorwort..... | 1 |
| 2 Konzeptidee..... | 1 |
| 2.1 Anforderungen..... | 3 |
| 3 Design-Entscheidungen..... | 3 |
| 3.1 Steuercomputer..... | 3 |
| 3.2 Lichtsensor..... | 4 |
| 3.3 Energieversorgung..... | 5 |
| 3.4 Lichtquelle..... | 5 |
| 3.5 Schaltungsaufbau..... | 6 |
| 4 Software Entwicklung..... | 6 |
| 5 Bauteilübersicht und -kosten..... | 10 |
| 6 Zusammenbau..... | 11 |
| 6.1 Löten..... | 11 |
| 6.2 Zusammenstecken..... | 16 |
| 6.3 Software einspielen..... | 16 |
| 7 Gehäuse(ein)bau..... | 18 |
| 8 Inbetriebnahme..... | 19 |
| 9 Benutzung..... | 19 |
| 10 Programmcode..... | 21 |
| 10.1 Benötigte Bibliotheken..... | 22 |
| 11 Anhang..... | 24 |
| 11.1 Aktuelle Anmerkungen und Ergänzungen..... | 24 |
| 11.2 Lambert-Beer'sches Gesetz..... | 25 |
| 11.3 Gehäusemodell-Code für OpenSCAD..... | 27 |

Lizenz:



Dieses Material steht unter der Creative-Commons-Lizenz Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International. Um eine Kopie dieser Lizenz zu sehen, besuchen Sie <http://creativecommons.org/licenses/by-nc-sa/4.0/>.

1 Vorwort

Die MINT¹-Wissenschaften gelten als Kernwissenschaften für den Standort Deutschland da sie zu den innovativsten und wirtschaftlich wichtigsten Bereichen gehören. Ungünstiger Weise finden sich immer weniger junge Menschen, die sich für diese Themen interessieren lassen – Stichwort: Fachkräftemangel. Hier setzt MINT-Förderung an. Eine Idee dazu: Wenn es gelingt, Jugendliche schon früh in ihrer Schullaufbahn in Kontakt zu bringen mit Fragestellungen, die ihr Interesse an MINT Themen weckt, kann es gelingen, mehr Jugendliche zur Aufnahme einer Ausbildung oder eines Studiums im MINT Fächerkanon zu bewegen und somit dem drohenden Fachkräftemangel entgegenzuwirken.

Hier setzt das im Folgenden vorgestellte Projekt an. Der Selbstbau des Fotometers verknüpft Themen aus den Bereichen.

- Physik Licht, Optik, LED
- Chemie Konzentration, Farbreaktionen, (Bouguer)-Lambert-Beer'sches Gesetz
- Elektrotechnik Strom, Spannung, Bauelemente, Schaltplan, Löten
- Informatik Programmieren, WIFI, Webserver
- Mathematik Gleichungen, Wertetabellen
- Kunst 3D-Modellerstellung und Druck, Webseiten Gestaltung

Bewusst wurde Wert darauf gelegt, ein technisch anspruchsvolles Projekt mit aktueller „High-Tech“ - Ausstattung so zu gestalten, dass sich die Projektinhalte trotzdem in kurzer Zeit an Jugendliche vermitteln lassen. Und es wurde nicht das technisch Mögliche aus dem Design heraus geholt, sondern extra Raum gelassen, um die Jugendlichen zu eigenen kreativen Modifikationen anzuregen.

2 Konzeptidee

Im Rahmen dieses Projektes soll ein Fotometer entstehen, dass für geringe Kosten auch von Personen ohne große Vorkenntnisse im Elektronik-, Computer und Softwarebereich selbst herstellbar ist.

Zu den grundlegenden Komponenten eines Fotometers gehören:

- eine Lichtquelle
- ein Monochromator
- eine Messküvette/Messgefäß
- ein Lichtsensor
- ein die Messdaten aufbereitendes System
- etwas zum Bedienen und zum Visualisieren von Messdaten (Benutzer Interface)
- sowie ein Halter für das Messgefäß, um reproduzierbare Messungen zu ermöglichen

Eine wesentliche Eigenschaft eines Fotometers ist die Messung der Lichtabschwächung (Lichtabsorption) bei einer bestimmten Wellenlänge des Lichts. Viele käufliche Fotometer erlauben die Vorauswahl dieser Wellenlänge. Technisch wird dies häufig durch Auffächern eines breitbandigen Lichtstrahls und Auswahl eines kleinen schmalbandigen Bereichs oder durch ein Farbfilter erwirkt. Diese optische Baugruppe – der sogenannte Monochromator – ist ein erheblicher Kostenfaktor. In diesem Projekt soll daher durch Einsatz von LEDs ein Monochromator unnötig werden. LEDs haben ein mehr oder weniger schmalbandiges Lichtspektrum. Durch die Auswahl der LED Farbe kann somit ein für die Messauf-

¹ MINT ist eine Abkürzung und wird aus den Anfangsbuchstaben der Wissenschaftsbereiche **M**athematik, **I**nformatik, **N**aturwissenschaften und **T**echnik zusammengesetzt.

gabe „passendes“ Licht gewählt werden. Die 1 zeigt schematisch die Lichtintensität in Abhängigkeit von der Wellenlänge für verschieden farbige LEDs.

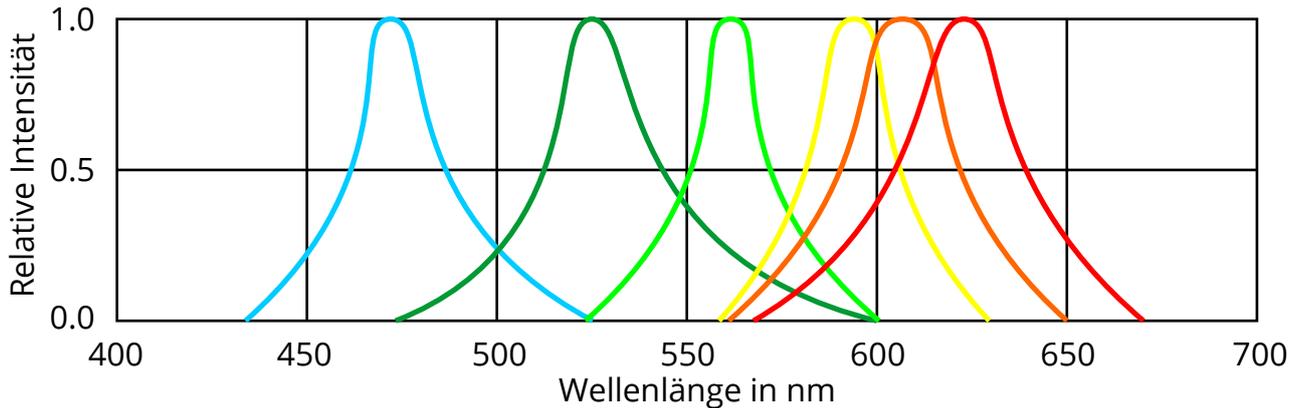


Abbildung 1: Lichtintensität verschieden farbiger LEDs

Die Breite des Spektrums der jeweiligen LED variiert von Hersteller zu Hersteller. Einerseits wäre ein schmaleres Band wünschenswert, um eine hohe Wellenlängenselektivität zu erreichen. Andererseits ermöglicht die Überschneidung der einzelnen Bänder auch die Beschränkung auf einige wenige LEDs, um einen breiten Wellenlängenbereich abdecken zu können.

Für die Auswahl eines geeigneten Lichtsensors sind folgende Kriterien wünschenswert:

- großer Wellenlängen Messbereich, um Messungen von ultraviolettem bis zu infrarotem Licht möglich zu machen
- hohe Auflösung, um Lichtintensitäten fein abgestuft unterscheiden zu können
- wenig Querempfindlichkeiten gegen Umwelteinflüsse (z.B. Temperatur)
- geringer elektrotechnischer Schaltungsaufwand, um an die Messergebnisse zu gelangen

Insbesondere der letzte Punkt legt die Verwendung eines Sensors mit integrierter digitaler Schnittstelle nahe. Signalverstärkende Elektronik und der damit verbundene Aufwand wird damit überflüssig.

Für die Steuerung der LED und die Verarbeitung der Daten des Lichtsensors drängt sich die Verwendung eines kleinen Computers auf. Um die Messdaten anzuzeigen und um Messungen starten zu können, werden noch eine Anzeigemöglichkeit und Bedienknöpfe benötigt. Gelingt es einen kleinen Computer auszuwählen, der seine Daten über eine von Tablet oder Smartphone nutzbare Datenschnittstelle anbieten kann, kann auf eine separate Anzeige und die Bedienknöpfe verzichtet werden.

Um aufseiten des Smartphones auf ein zusätzlich zu erstellendes Programm verzichten zu können, ist es wünschenswert, die Daten und die Möglichkeit Messungen zu starten in Form einer Webseite anzubieten, die von jedem internetfähigen Gerät, also auch von Smartphones, abgerufen werden kann. Die Einbindung eines Smartphones in das Konzept spart Kosten für Display und Bedienelemente und ist geeignet die Akzeptanz des Projekts bei Jugendlichen zu erhöhen. Der benötigte Computer sollte also am besten über eine WLAN/WIFI-Schnittstelle verfügen und die Möglichkeit haben einen kleinen Webserver zu realisieren.

Die einzelnen Bauteile des Fotometers sollen auch von Anfängern im Elektronikbereich handhabbar sein. Lötarbeiten sollen auch von „Erstlöttern“ durchführbar und von kurz eingewiesenem Projektbegleitern begutachtbar und korrigierbar sein. Lötarbeiten an empfindlichen oder sehr kleinen Bauteilen kommen somit nicht infrage.

2.1 Anforderungen

Netzwerkschnittstelle

Das Fotometer sollte **nicht** mit dem Internet verbunden sein, um Sicherheitsbedenken hinsichtlich der Netzwerksicherheit Rechnung zu tragen. Die Absicherung von Geräten, die mit dem Internet verbunden sind, bedarf deutlich erweiterter Computerkenntnis und unterliegt der Bedingung, dass auch in der Folgezeit stetig Aktualisierungen eingebracht werden, um eben diese Netzwerksicherheit auch weiterhin gewährleisten zu können.

Nicht überall und erst recht nicht in vielen Schulgebäuden gibt es Zugang zu einem WLAN/WIFI – Netzwerk. Daher wäre es wünschenswert, wenn das Fotometer ein eigenes WLAN/WIFI – Netzwerk aufbaut und damit auch sicherstellt, dass keine direkte Verbindung zum Internet besteht.

Lichtsensor

Auf dem Markt sind mehrere Modelle mit großem Messbereich und hoher Auflösung verfügbar. Diese Hightech-Sensoren liegen jedoch häufig nur als wenige Millimeter große SMD-Bauteile vor und sind somit für Anfänger im Elektronikbereich eher ungeeignet. Es gibt jedoch auch einige dieser Sensorchips, die schon auf sogenannte „Breakout-Boards“ montiert sind. Diese nur unwesentlich teureren Varianten sind auch für „ungeübte Elektroniker“ einfach zu verwenden, weil sie die Lötarbeiten an kleinen und empfindlichen Bauteilen vermeiden.

Steuercomputer

Der Steuercomputer soll preiswert bis billig sein. Es gibt eine große Anzahl verschiedener vom technischen Aspekt her geeigneter Systeme z. B. ein „Raspberry Pi“ oder „Arduino“-System bzw. eine entsprechende Variante. Das benötigte Entwicklungssystem für die Programmierung der Fotometer-Software sollte keine hohen Zusatzkosten verursachen und muss auch für Programmieranfänger geeignet sowie auch in Schul-Computerräumen, die nicht auf dem neusten technischen Stand sind, nutzbar sein.

3 Design-Entscheidungen

3.1 Steuercomputer

Als Steuercomputer (Mikrocontroller) wird ein Arduino kompatibles Computersystem ausgewählt, weil eine kostenlose Entwicklungsumgebung und viele günstige Modelle verfügbar sind. Die Modellvarianten mit WIFI-Erweiterung sind jedoch vergleichsweise gesehen teuer.

Die Steuercomputer mit Espressif ESP8266 System bringen eine WLAN/WIFI-Schnittstelle jedoch gleich von Haus aus mit, sind sehr günstig und lassen sich als Arduino – Variante in die Software-Entwicklungsumgebung einpassen. Die Anzahl der digitalen und analogen Ein- und Ausgabe Anschlüsse ist zwar geringer als bei den „originalen“ Arduinos, sind aber für dieses Projekt mehr als ausreichend. Nachteilig ist jedoch, dass es nur wenige Anbieter auf dem deutschen Markt gibt.

Standardmäßig haben die Espressif ESP8266 Systeme keine Programmierschnittstelle über einen USB-Anschluss. Dies wäre jedoch wünschenswert, um eine einfache Handhabung auch für Anfänger

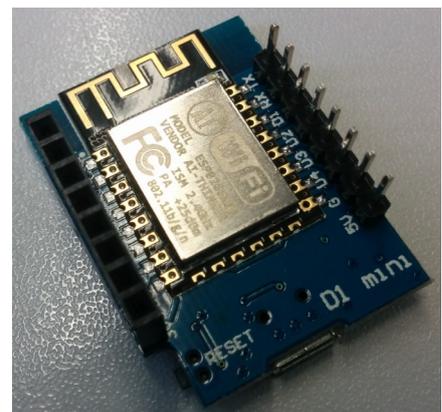


Abbildung 2: „D1 mini“

zu ermöglichen. Glücklicherweise sind auch Mikrocontroller-Boards mit Espressif ESP8266 System und mit einer USB-Anschluss-Baugruppe gemeinsam auf einem Board verfügbar. Sie sind etwas teurer, als die Varianten ohne USB-Anschluss sind dafür dann aber sehr einfach in der Anwendung.

Für dieses Projekt wird die Variante mit USB entsprechend einem Design der Firma WeMos bzw. Lolin, der „D1 mini“, ausgewählt. Vermarktet wird das Modul Als „D1 mini“ und wird von mehreren Firmen hergestellt.

3.2 Lichtsensor

Als Lichtsensor soll der TSL2561 – Chip der Firma TAOS oder der BH1750 - Chip der Firma Rohm Semiconductor jeweils auf einem Breakout-Board eingesetzt werden. Diese Sensoren haben eine hohe Auflösung und Empfindlichkeit im sichtbaren Lichtspektrum - der TSL2561 ermöglicht auch Messungen im Infrarot-Bereich. Die Messergebnisse können nach einem vom Hersteller verifiziertem Algorithmus in LUX berechnet werden und es gibt mehrere Versionen der Breakout-Boards auf dem Weltmarkt, die auch bei Distributoren im Inland verfügbar sind. Auch in einem Fotometrie-Projekt² von Oliver Happel hat sich dieser TSL2561 - Chip bewährt.



Abbildung 3: Lichtsensor TSL2561

Module mit dem TSL2561 sind die bevorzugte Variante. Sie sind jedoch teilweise schwer zu bekommen und auch teurer als Module mit dem BH1750. Der BH1750 liefert im sichtbaren Bereich sehr gute Ergebnisse und ist deutlich preiswerter. Die Module der Sensoren tragen zum Teil noch weitere Bezeichnungen. Auf den von uns verwendeten Modulen haben wir GY-2561 (TSL2561) oder GY-302 (BH1750) vorgefunden.

Einige Kenndaten des TSL 2561:

- Empfindlichkeit an die des menschlichen Auges angeglichen
- Automatische Unterdrückung des 50/60Hz Flackerns durch Einstrahlung von elektrischen Licht in Innenräumen
- Sehr geringer Energiebedarf (Minimum: 0,75 mW)
- Hohe dynamische Auflösung von 1.000.000 zu 1 durch programmierbare Signalverstärkung und Integrationszeit
- Baugröße (L · B · H): 3,8 mm · 2,6 mm · 1,35 mm



Abbildung 4: TSL2561 Chip

Spektrale Empfindlichkeit

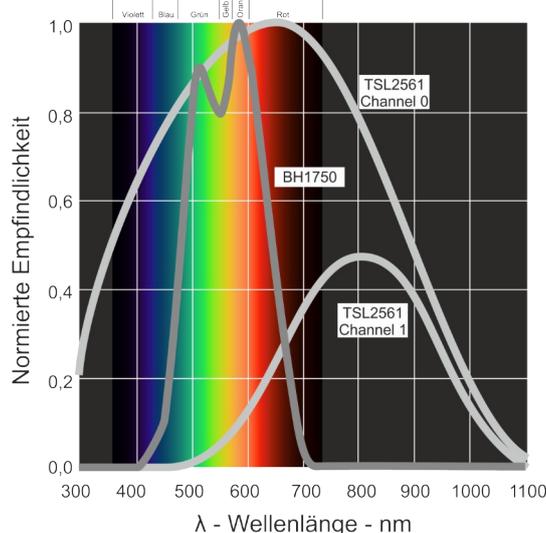


Abbildung 5: Spektrale Empfindlichkeit der Chips

² Oliver Happel (2015) Analytische Methoden mit dem LED Photometer. AATIS-Praxisheft 25

3.3 Energieversorgung

Die Energieversorgung des Gesamtsystems soll über den USB-Anschluss auf dem „D1 mini“ Modul mit einem USB-Steckernetzteil oder einer USB-Powerbank erfolgen. Bei der Nutzung einer Powerbank wird auch ein mobiler Betrieb des Fotometers möglich. Da die Stromaufnahme der gesamten Schaltung nur gering sein wird, ist auch eine leistungsarme Powerbank ab ca 2000 mAh geeignet.

3.4 Lichtquelle

Für das Fotometer stehen zwei Ausbauvarianten zur Verfügung: eine mit einer einfarbigen LED und eine weitere mit einer dreifarbigem LED. In den folgenden Texten wird ggf. auf die Unterschiede der jeweiligen Variante hingewiesen.

LEDs benötigen zum Betrieb einen den Strom begrenzenden Widerstand. Die Höhe des maximalen Stroms und die zum Betrieb benötigte Spannung lässt sich üblicherweise aus dem Datenblatt entnehmen. In diesem Projekt wird beabsichtigt die LED direkt am Ausgangspin des Mikrocontrollers zu betreiben. In diesem Fall ist es somit auch wichtig den maximal möglichen Strom am Ausgang des Mikrocontrollers zu berücksichtigen, um ein Überlasten des Mikrocontrollers zu vermeiden. Der ESP8266 Chip kann am Digitalausgang maximal 12 mA bei einer Ausgangsspannung von 3,3 Volt zur Verfügung stellen. Die Betriebsspannung einer LEDs wird oft bei einem Strom von 20 mA angegeben. Bei der hier verwendeten LED soll der Strom jedoch nur 12 mA betragen und es fällt dann auch eine kleinere Spannung als im Datenblatt für 20 mA angegeben über der LED ab. Aus dem Datenblatt der LED lässt sich oft aus einer Kennlinie auch ein Strom bei 12 mA ablesen (z. B. 2 Volt). Über dem Vorwiderstand der LED müssen somit ca 1,3 Volt abfallen (Zusammen 3,3 Volt). Über das Ohmsche Gesetz lässt sich somit der Widerstandswert berechnen: $R = \frac{U}{I}$. In

unserem Fall also: $R = \frac{1,3V}{0,012A} = 108,33\Omega$.

Einen Widerstand mit 108,33 Ohm gibt es nicht zu kaufen, daher wird in diesem Fall der nächst größere kaufbare Wert – 120 Ohm – gewählt.

Wird eine LED mit einem anderen Spannungsabfall bei maximal 12 mA gewählt, ist die Spannung am Vorwiderstand eine andere und somit ist der Widerstand neu zu berechnen. Bei RGB-LEDs ist der Spannungsabfall bei jeder LED-Farbe anders und muss für jede Farbe einzeln dem Datenblatt entnommen werden. Für die in diesem Projekt mehrfach verwendete RGB-LED „LL-509RGBC2E-006“ (Reichelt Artikel-Nr.: LED LL 5-8000RGB) haben sich folgende Werte für die Vorwiderstände bewährt:

blaue LED: R = 0 Ohm, grüne LED: R = 62 Ohm, rote LED: R = 182 Ohm

Liegt kein Datenblatt vor, sollte der Spannungsabfall experimentell durch Variation des Vorwiderstands ermittelt werden.

Soll nur jeweils eine LED Farbe verwendet werden, lassen sich viele LEDs mit unterschiedlichen Leuchtfarben finden die das Kriterium 12 mA bei 2 Volt erfüllen und dann bei Bedarf gegeneinander einfach austauschbar sind. Dies Ermöglicht auch Messungen bei speziellen Wellenlängen.

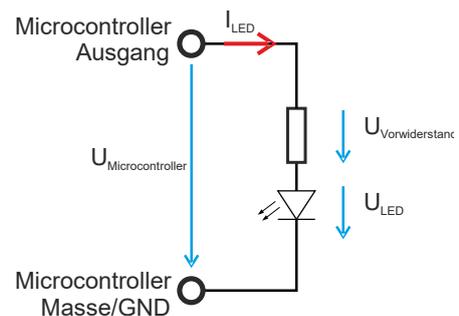
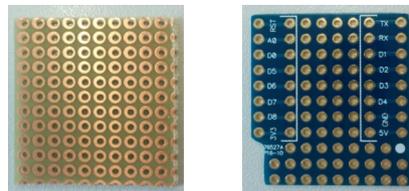


Abbildung 6: Anschluss der LED mit Strom und Spannungen an LED und Vorwiderstand

3.5 Schaltungsaufbau

Die elektrische Verbindung der einzelnen Baugruppen soll auf einer Platine, die auch die räumliche Ausrichtung der einzelnen Bauteile zueinander bestimmt, erfolgen. Eine preiswerte Lösung hierfür sind sogenannte Lochrasterplatinen. Diese Platine stellt die elektrischen Verbindungen zwischen Mikrocontroller Baugruppe, dem Lichtsensor Modul und der LED her. Eine Sandwich-Struktur – unten die Mikrocontroller Baugruppe, mittig die Lochrasterplatine und darauf das Lichtsensor-Modul und die LED – ermöglicht einen besonders platzsparenden Aufbau. Da von beiden Seiten Bauteile auf der Platine befestigt werden müssen, ist eine doppelseitige Kupferbelegung sinnvoll. Benötigt wird eine Lochrasterplatine mit 10 mal 11 Löchern. Üblicherweise werden Lochrasterplatinen in größeren Bau- größen angeboten, sie lassen sich jedoch leicht in einzelne Teile unterteilen. Hierzu wird die Platine zersägt oder mit einem Messer so angeritzt, dass einzelnen Löcher genau mittig unterteilt werden, und dann über einer Kante gebrochen. Da somit an den Rändern halbe Lochreihen übrig bleiben wird pro Platine ein Bereich von 11 mal 12 Löchern benötigt. Aus einer handelsüblichen Lochrasterplatine in den Maßen von 100 mm mal 160 mm lassen sich so 15 kleine Platinen gewinnen.

Die einfachere Lösung besteht darin eine entsprechende Platine im gewünschten Format zu kaufen. Passend zum „D1 mini“ – Modul gibt es so eine Platine („D1 mini“- Proto Board).



4 Software Entwicklung

Die Entwicklung und Änderung der Software kann mit der kostenlos erhältlichen Arduino IDE (integrated development environment / integrierte Entwicklungsumgebung) erfolgen. Diese Software ist auch für Programmierereinsteiger konzipiert und ermöglicht einen einfachen Einstieg in die Welt der Programmierung von Mikrocontrollern. Mit der Arduino IDE kann neuer Programmcode erstellt oder bereits bestehender bearbeitet werden und danach einfach über einen USB Verbindung auf das Microcontroller Board geladen werden. Alles ist im Vergleich zu professionellen Entwicklungsumgebungen etwas einfacher gehalten. So gibt es z. B. Syntax-Highlighting – das ist die farbige Hervorhebung von bekannten Schlüsselworten im Programmcode – aber eine automatische Code-Vervollständigung gibt es (noch) nicht. Dennoch bietet die IDE aber auch erfahrenen Anwendern die meisten Möglichkeiten aktueller Softwareentwicklungssysteme.

Was ist Arduino?

- Unter Arduino versteht man eine aus Soft- und Hardware bestehende Computer Plattform
- Die Hardware besteht aus einem Mikrocontroller mit analogen und digitalen Ein- und Ausgängen
- Die Entwicklungsumgebung (IDE) soll auch technisch weniger Versierten den Zugang zur Programmierung und zu Mikrocontrollern erleichtern
- Es gibt sehr viele Varianten mit und ohne Extras (Preisspanne 2€ bis 100€) und auch diverse Hersteller kompatibler Ausführungen

Die Arduino IDE kann von der Webseite der Arduino Initiative unter „<https://www.arduino.cc/>“ heruntergeladen werden.

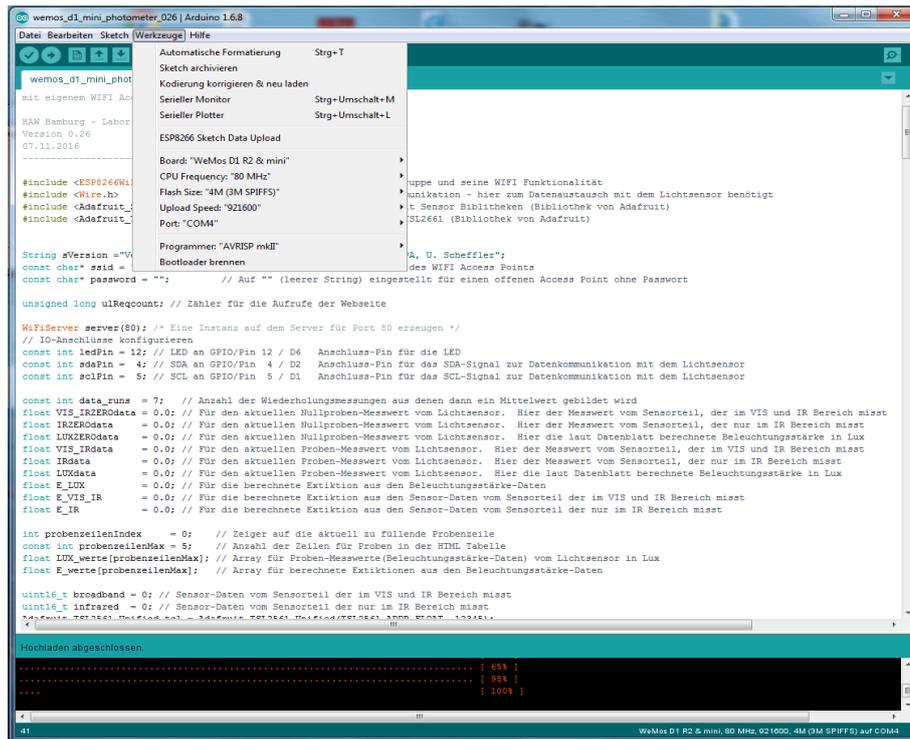


Abbildung 7: Arduino IDE

Für dieses Projekt muss diese Softwareumgebung um die Möglichkeit auch die ESP8266 Systeme programmieren zu können, ergänzt werden. Hierfür eignen sich alle Versionen der Arduino IDE ab Version 1.6.5 (von uns getestet bis Version 1.8.12).

Hilfreiche Weblinks zum Thema Programmierung des ESP8266 mit der Arduino IDE finden sich unter:

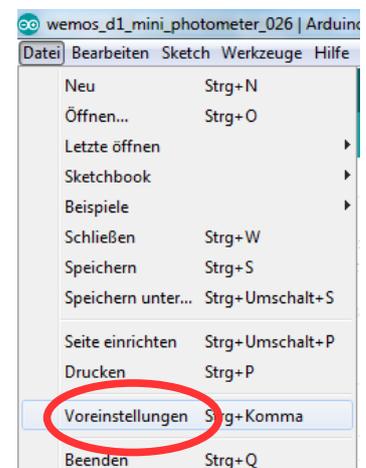
<https://arduino-hannover.de/2015/04/08/arduino-ide-mit-dem-esp8266/>

<https://www.wemos.cc/tutorial/get-started-arduino.html>

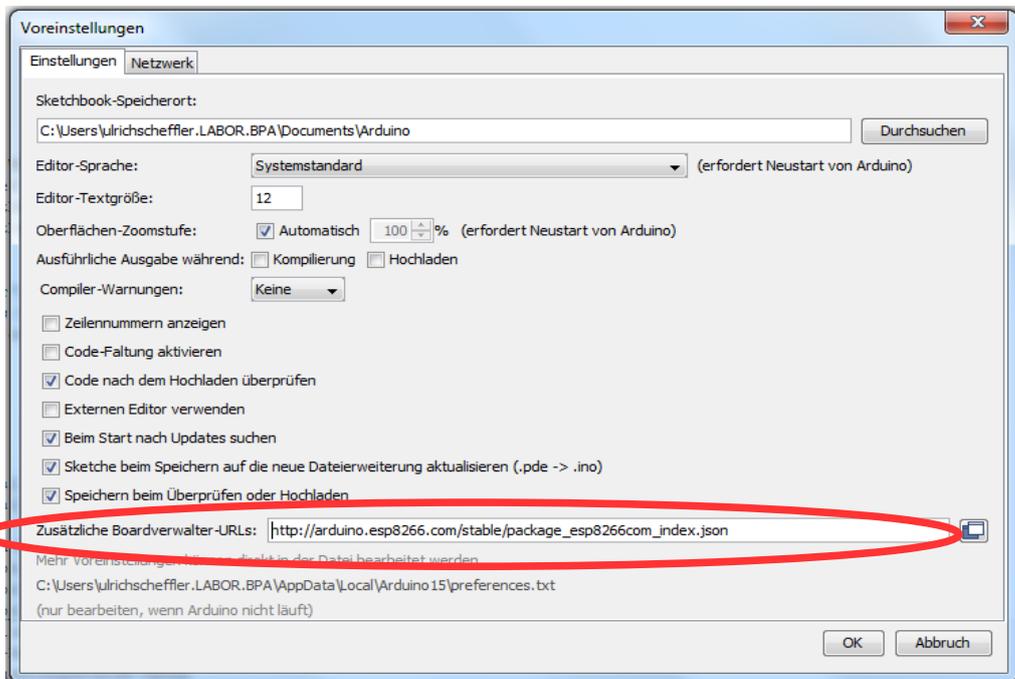
<https://github.com/esp8266/Arduino>

Die Ergänzung der Arduino IDE erfolgt am einfachsten so:

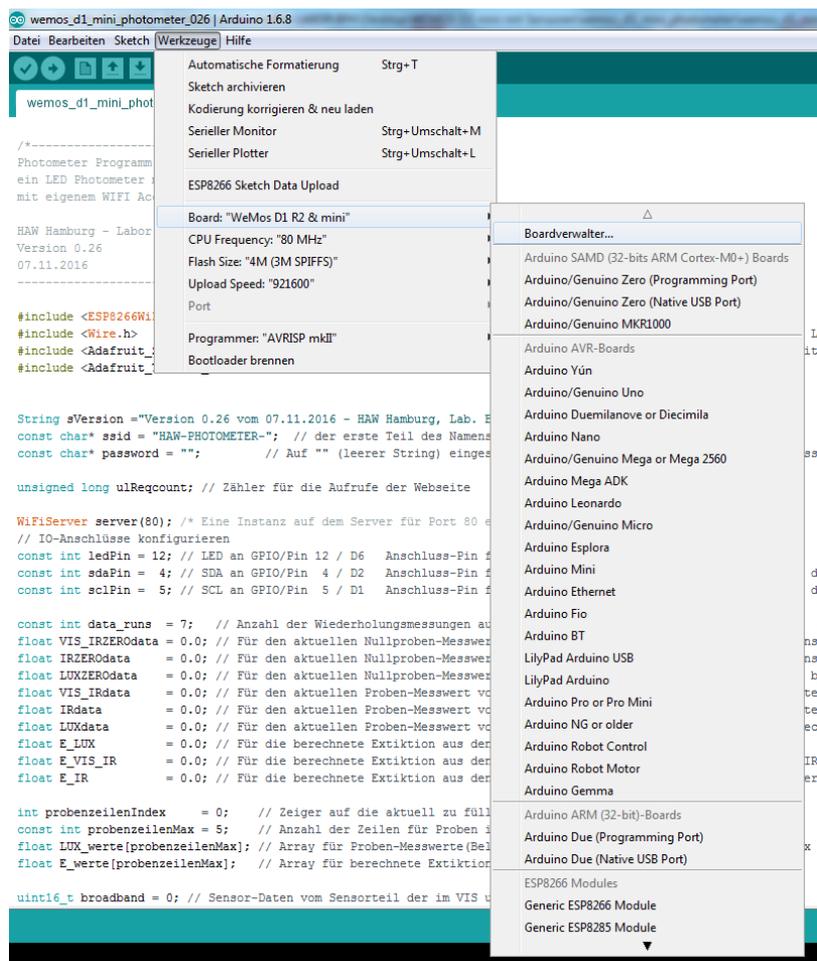
1. Arduino IDE starten.
2. Im Menü "Datei" den Punkt "Voreinstellungen" auswählen.



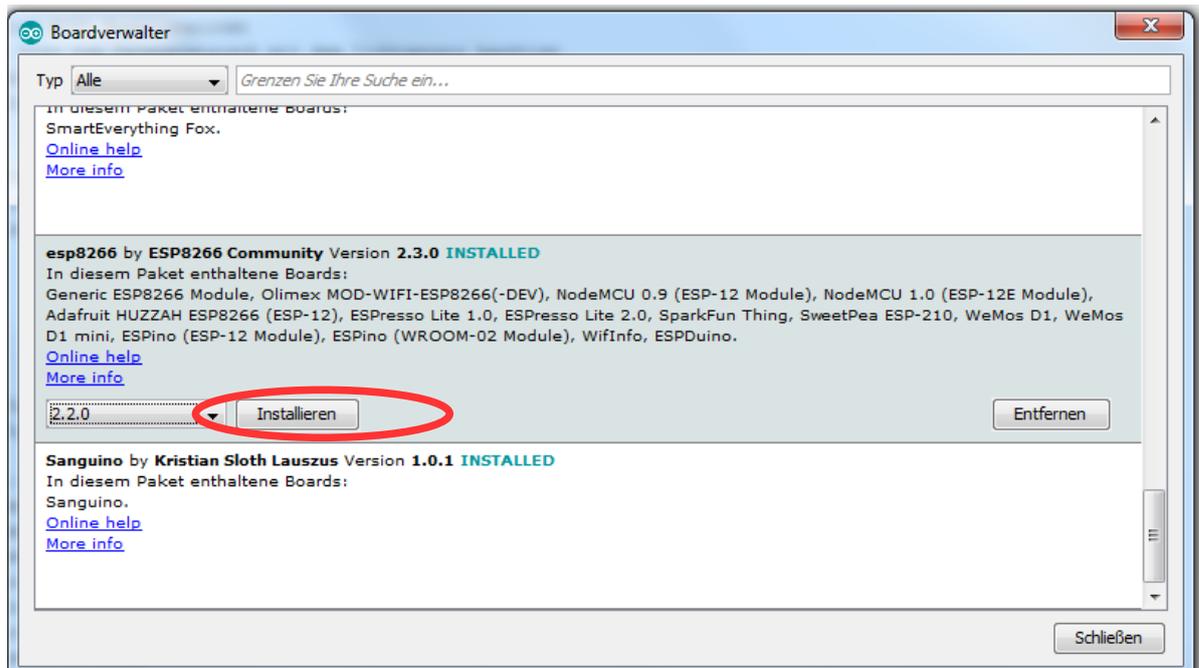
3. Auf der Karteikarte "Einstellungen" unter "Zusätzliche Boardverwalter-URLs" http://arduino.esp8266.com/stable/package_esp8266com_index.json eintragen.



4. Dann im Menü "Werkzeuge" den Punkt "Board: "..."" auswählen (Hinweis: für "... " steht immer das zuletzt ausgewählte Zielsystem. Hier im Screenshot: Board: "WeMos D1 R2 & mini")

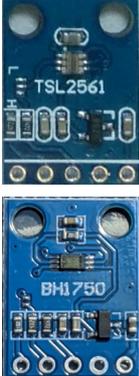
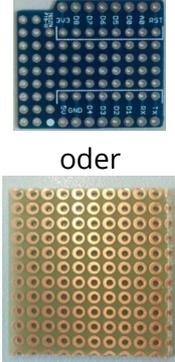


5. Den obersten Punkt "Boardverwalter..." auswählen.



6. Den Eintrag "esp8266 – by ESP8266 community" suchen (meist weit unten) und anklicken und "Installieren" drücken. (Hinweis: Im Screenshot ist die Installation bereits ausgeführt worden.)
7. Arduino IDE beenden und erneut starten.
8. Jetzt kann unter "Werkzeuge" "Board: "..."" der Eintrag "LOLIN(WEMOS) D1 R2 & mini" ausgewählt werden.
9. Ab jetzt sollten Programme für das Zielsystem "D1 mini" erzeugt werden können.

5 Bauteilübersicht und -kosten

| Mikro-controller | Lichtsensoren | Lichtquelle | Vorwiderstand für LED | Adapterplatine | Spannungsversorgung |
|---|---|---|---|---|---|
|  |  |  |  |  |  |
| D1 mini | TSL2561 oder BH1750 (Beschriftung auch: GY-2561 oder GY-302) | LED | Widerstand | D1 mini Proto- Board oder Lochraster- platine | Powerbank |

Kostenübersicht

| Nr | Bauteil | Funktion | Bezug in Deutschland | | | Direktimport |
|-----------------|--------------------|-------------------|----------------------|--|---------------|---------------|
| | | | Anbieter | Artikel Nr | Kosten | Kosten |
| 1 | D1 mini | Mikrocontroller | Reichelt | D1 Mini | 4,81 € | 2,35 € |
| 2 | BH1750 | Lichtsensoren | Reichelt | DEBO BH 1750 | 2,16 € | 0,67 € |
| 3 | RGB LED | Lichtquelle | Reichelt | LED LL 5-8000RGB | 0,59 € | 0,15 € |
| 4 | Widerstand (62 Ω) | Vorwiderstand LED | Reichelt | METALL 62,0 | 0,08 € | 0,03 € |
| 5 | Widerstand (182 Ω) | Vorwiderstand LED | Reichelt | METAL 182 | 0,08 € | 0,03 € |
| 6 | D1 mini ProtoBoard | Adapterplatine | Reichelt | D1Z PROTO | 0,95 € | 1,37 € |
| Summe | | | | | 8,67 € | 4,60 € |
| Optionen | | | | | | |
| 2 | TSL2561 | Lichtsensoren | | | | 2,88 € |
| 6 | Lochrasterplatine | Adapterplatine | Reichelt | UP 832EP (reicht für 15 Stück, 4,53 €) | 0,30 € | |
| 7 | Powerbank | Energieversorgung | Reichelt | GOO 71599 | 5,25 € | 3,89 € |

Stand: November 2020

Anbieter in Deutschland

Eckstein GmbH (Versandkosten ca. 3,49 €)
 Reichelt Elektronik (Versandkosten ca. 5,60 €)
 Azando (roboter-bausatz) (Versandkosten ca. 2,99 €)

<https://eckstein-shop.de>

<http://www.reichelt.de>

<https://www.roboter-bausatz.de>

Anbieterplattform für Direktimport

Aliexpress (Versandkosten oft inklusive)

<https://www.aliexpress.com>

6 Zusammenbau

6.1 Löten

Die Mikrocontroller Platine und das Lichtsensor-Modul werden in speziellen elektrisch leitfähigen **Tüten** angeliefert. Dies sollte man als Hinweis auf empfindliche Bauteile deuten. Tatsächlich sollten elektrostatische Entladungen über diese Bauteile vermieden werden. Profis legen bei der Verarbeitung ein mit Schutz Erde verbundenes Armband an und arbeiten nur auf leitfähigen Unterlagen. Meistens ist jedoch auch eine Verarbeitung dieser Bauteile auf einem einfachen Tisch und ein Abbauen evtl. vorhandener elektrostatische Ladung durch vorheriges Anfassen des Tischbeins aus Metall oder der Heizung gut möglich.

Nach dem Auspacken können die benötigten Teile herausgesucht werden. In den Tüten für das Mikrocontroller Board und der Adapterplatine sind einige für dieses Projekt nicht benötigte Stift- und Buchsenleisten zu viel drin.

Benötigt werden:

| Anzahl | Stift-/Buchsenleiste |
|--------|--|
| 2 | 8-polige Stiftleiste |
| 2 | 8-polige Buchsenleiste mit kurzen Beinchen |
| 1 | 4-polige Stiftleiste (Abgetrennt von einer 8-poligen Stiftleiste) für die Version mit einer dreifarbigen LED. 2 polige Stiftleiste für die Version mit nur einer LED Farbe. |
| 1 | 4-polige Buchsenleiste (Abgetrennt von einer 8-poligen Buchsenleiste) für die Version mit einer dreifarbigen LED. 2 polige Buchsenleiste für die Version mit nur einer LED Farbe. |
| 1 | 4-polige Buchsenleiste mit um 90° abgewinkelten Beinchen (Abgetrennt von einer 8-poligen Buchsenleiste und danach Beinchen auf der Tischfläche umgebogen). |
| 1 | 4-polige Stiftleiste (Abgetrennt von einer 8-poligen Stiftleiste). |



Abbildung 8: Bauteile vor dem Zusammenbau

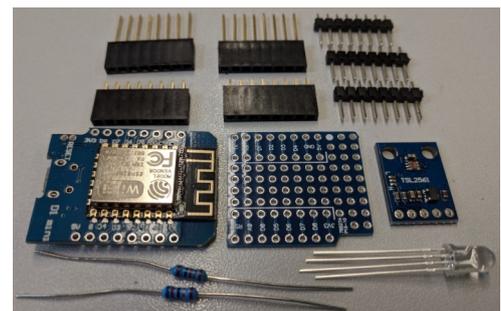


Abbildung 9: Benötigte Bauteile

Auf der Mikrocontroller Platine soll eine 8-polige Stiftleiste und eine 8-polige Buchsenleiste verlötet werden. Das Bild in 10 zeigt auf welcher Seite und welche Leiste wo bestückt werden soll. Bewährt hat sich zunächst erstmal jeweils nur einen Pin anzulöten, dann die Ausrichtung zu überprüfen und ggf. Nachjustieren – d. h. die Lötstelle wieder zu erwärmen und das Bauteil ausrichten – und dann erst die restlichen Anschlusspins zu verlöten.

Die Herstellung der Adapterplatine macht am meisten Mühe. In der Zeichnung in 14 auf Seite 14 ist ein Bestückungsplan abgebildet. Empfehlenswert ist es zuerst die Widerstände bzw. den Widerstand einzubauen. Die Drahtenden des Widerstands können auf der Lötseite der Platine umgebogen, verlötet und abgeschnitten werden. Die abgeschnittenen Drahtenden können benutzt werden, um die Drahtbrücken auf der Lötseite herzustellen. Dann kann die abgewinkelte 4-polige Buchsenleiste bestückt und verlötet werden. Danach die 4- bzw. 2-polige Buchsenleiste einbauen und verlöten – dabei auf einen rechtwinkligen Einbau achten. Jetzt fehlen noch die beiden 8-poligen Leisten. Auch hierbei gibt das Foto in 11 einen Hinweis für den richtigen Einbau.

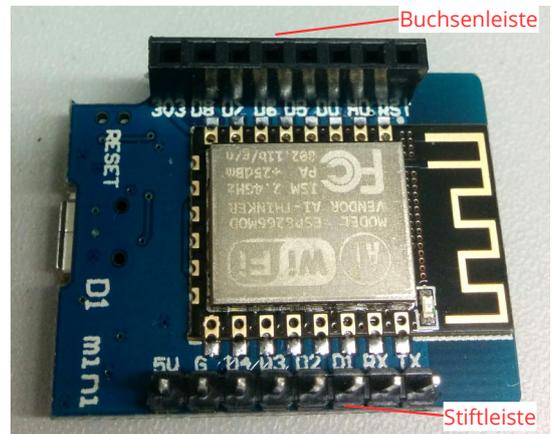
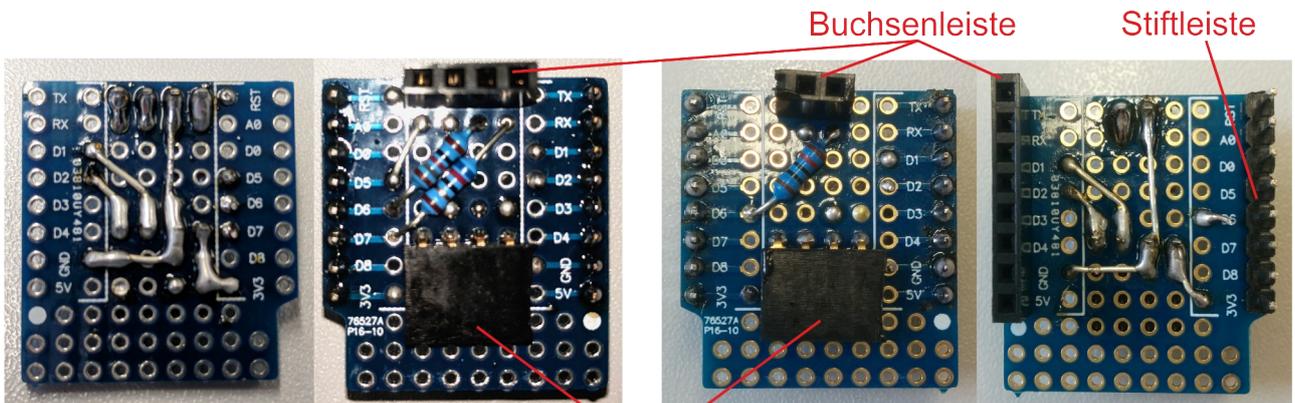


Abbildung 10: D1 mini mit Stift- und Buchsenleiste



Variante für dreifarbige LED

Buchsenleiste abgewinkelt

Variante für einfarbige LED

Abbildung 11: Bestückte Adapterplatine

Am hier genutzten Lichtsensor-Modul muss eine 4-polige Stiftleiste so eingelötet werden, dass die lange Seite der Pins auf der Bauteilseite bleibt (siehe 12). In das Loch mit der Beschriftung „INT“ sollte kein Stift eingebaut werden, er wird nicht benötigt und es kommt später nicht so leicht zu einer falschen Montage. Der eigentliche Lichtsensor-Chip sollte möglichst nicht mit dem Finger angefasst werden, weil sonst ein Fettfilm zurückbleibt, der hinterher im Betrieb störend sein kann.

Anmerkung: Es gibt diverse Varianten des Lichtsensor-Moduls am Markt, die grundsätzlich alle geeignet sind. Wird nicht das hier abgebildete Lichtsensor-Modul (Platinenkennzeichnung auf der Rückseite: GY-2561 oder GY-302) verwendet werden, muss ggf. der Anschlussplan an die abweichende Anordnung der Anschlusspins angepasst werden. Orientierung liefern hierbei die Beschriftungen der Anschlüsse. Verwendet werden: VCC, GND, SCL und SDA. Bei Verwendung eines anderen Lichtsensor Moduls ist auch die geometrische Lage des Lichtsensor Chips zu beachten. Er soll im montiertem Zustand der LED möglichst exakt gegenüber stehen, d. h. weder seitlich noch in der Höhe verschoben. Zusätzlich ist in diesem Fall zu beachten, das auch der Abstand des Lichtsensors zur Adapterplatine nicht zu gering werden sollte, da sonst die üblichen Küvetten, die dann später für die Messungen verwendet werden sollen, nicht mehr mittig durchstrahlt werden.

Die Beinchen der LED können kurz nach der Verdickungsstelle abgetrennt werden, sodass die Beinchen danach noch ca 5 mm lang sind. Die LED soll mit einer Stiftleiste im rechten Winkel verlötet werden. Das Bild in 12 bietet hierfür eine gute Orientierung, wie es hinterher aussehen soll. Aber Achtung! Die LED hat mehrere Anschlüsse, die in diesem Projekt auch nicht vertauscht werden dürfen (sonst funktioniert die LED nicht - aber meistens geht auch nicht gleich was kaputt). Die LED hat im Inneren einen dicken, fetten Teil und mehrere oder einen kleinen, schlanken Teil. Der Anschluss mit dem dicken, fetten Teil im Inneren der LED ist die Kathode (MINUS Seite) der LED.

Folgender Ablauf hat sich bewährt: Stiftleiste mit den langen Pins in die Buchsenleiste der Adapterplatine einstecken. Platine so drehen, dass diese neue Kombination oben ist. Die einfarbige LED kann jetzt so an die Stiftleiste gelötet werden, dass die Kathode (MINUS Seite) der LED links ist - bei der dreifarbige LED ist die Kathode der zweite Pin von links.



Abbildung 12: Sensor Modul (hier TSL2561) und LED mit Stiftleisten



Abbildung 13: LEDs mit Stiftleisten

links dreifarbige LED rechts einfarbige LED

Das Bild in 20 auf Seite 16 lässt erkennen wie es für die einfarbige LED hinterher aussehen soll.

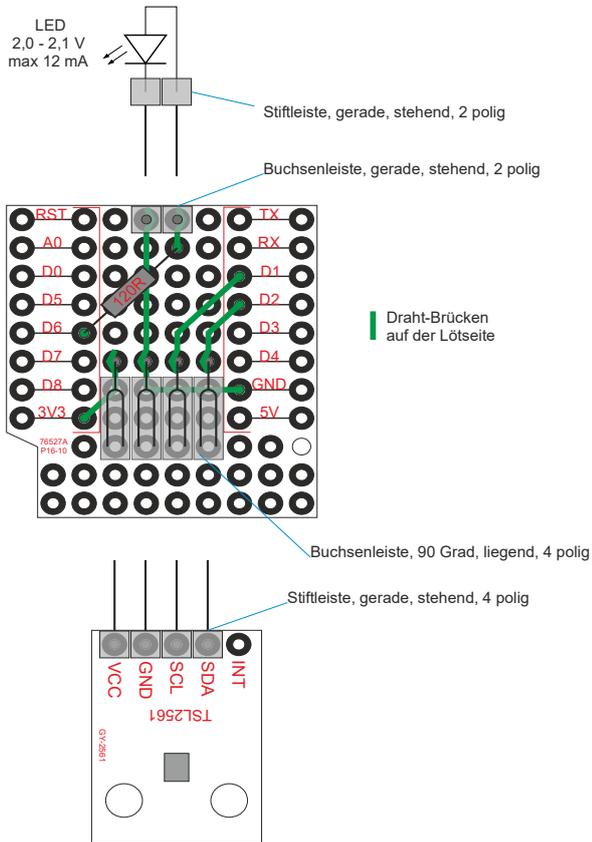


Abbildung 14: Bestückungsplan Adapterplatine für einfarbige LED Variante

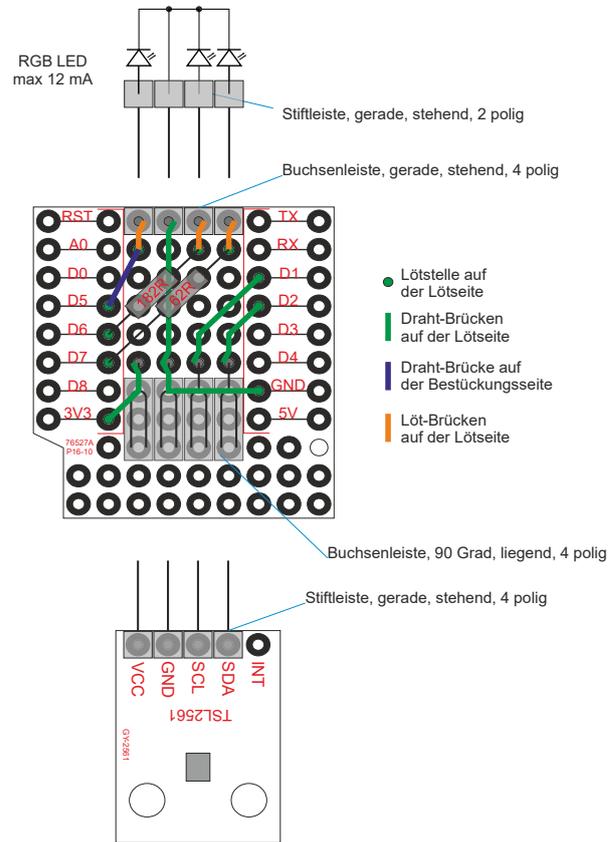


Abbildung 15: Bestückungsplan Adapterplatine für dreifarbige LED Variante

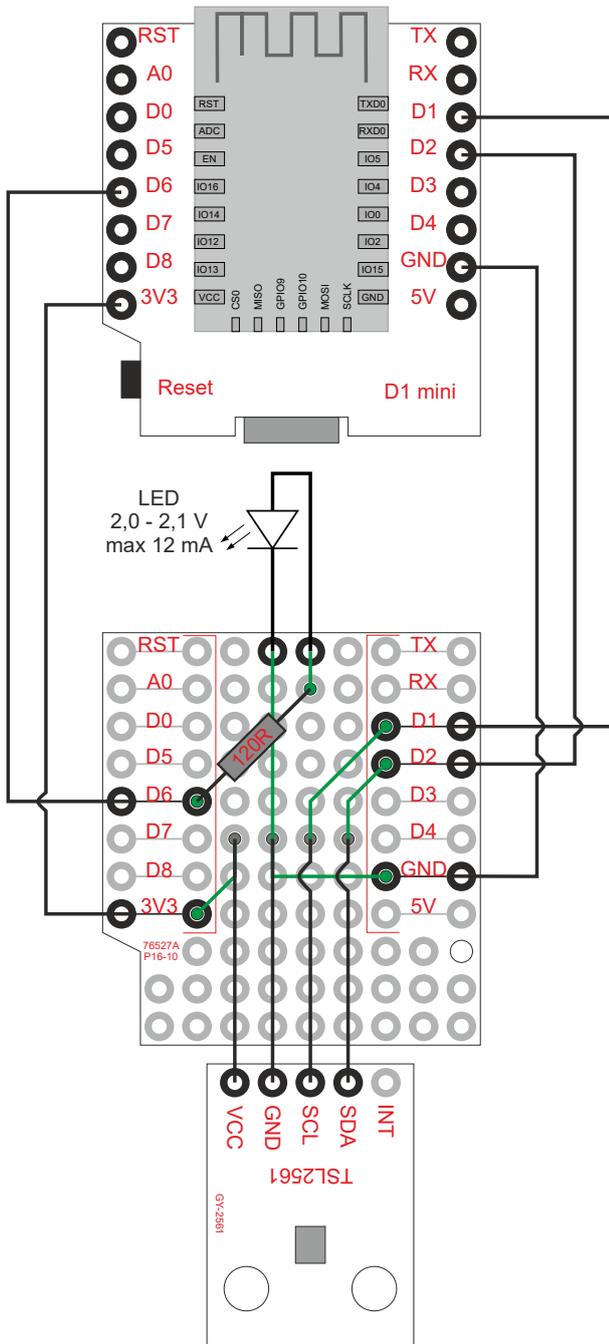


Abbildung 16: Schaltplan des Fotometers für einfarbige LED Variante

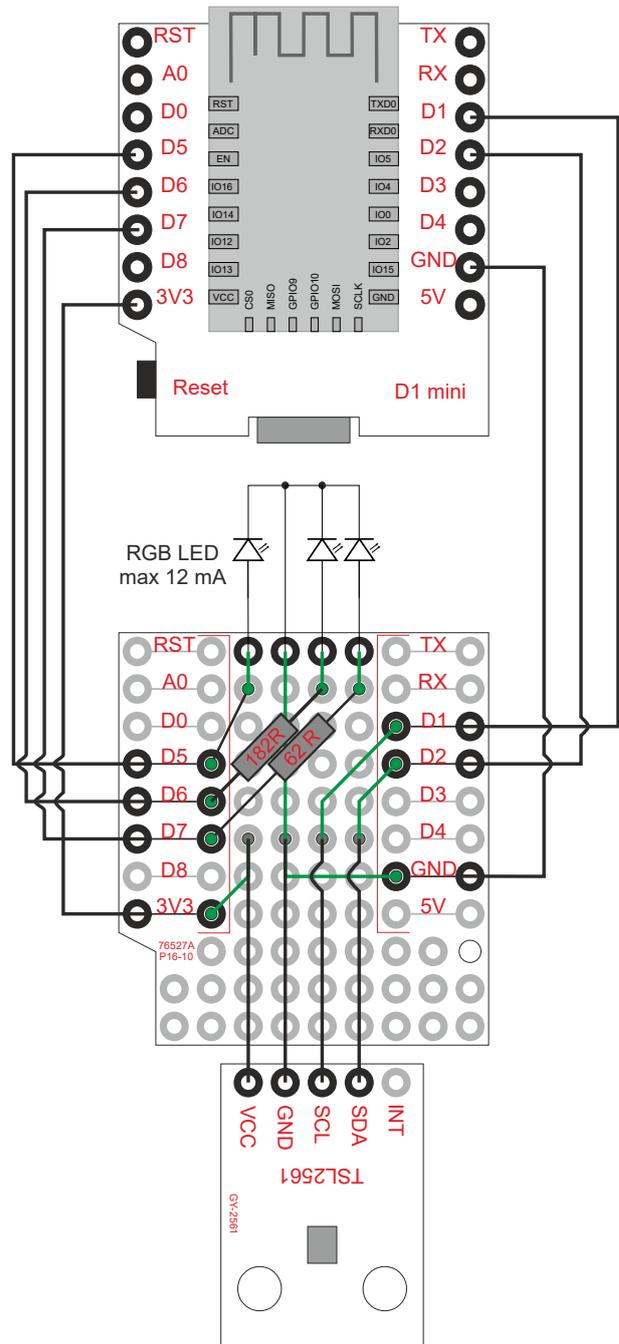


Abbildung 17: Schaltplan des Fotometers für dreifarbige LED Variante

6.2 Zusammenstecken

Wenn alles richtig verlötet wurde, sollte sich die Adapterplatine nun so auf die Mikrocontroller-Platine stecken lassen, dass beide Platinen die kleine Aussparung an der einen Ecke genau übereinander liegend haben. Siehe hierzu auch die Fotos in 19 und 20.

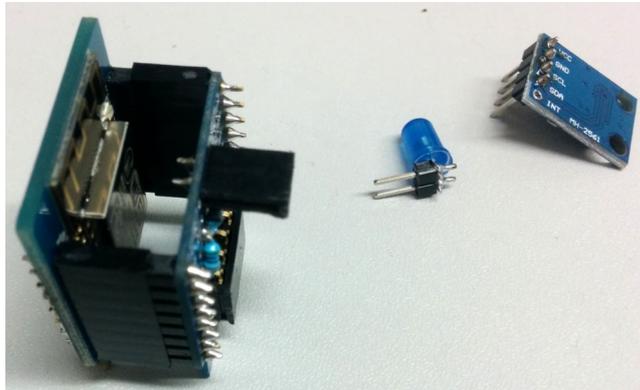


Abbildung 18: Mikrocontroller und Adapterplatine zusammenstecken

Das Lichtsensor Modul und die LED lassen sich nun einfach einstecken.

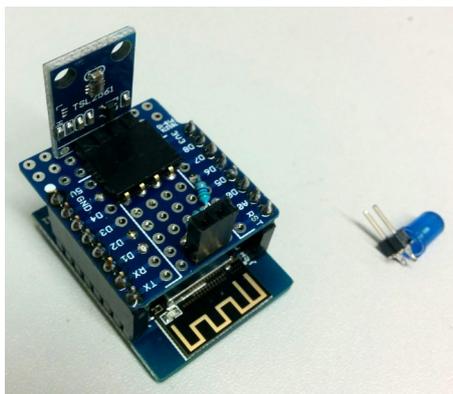


Abbildung 19: Lichtsensor einstecken



Abbildung 20: Alle Bauteile zusammengesteckt

6.3 Software einspielen

Zum Einspielen der Software muss der D1 mini über seine USB Schnittstelle mit dem PC verbunden werden. Evtl. muss jetzt noch einmalig eine Treibersoftware für den USB Chip des D1 mini installiert werden. Der passende Treiber für die CH340G Chip Serie ist z. B. unter:

https://www.wemos.cc/en/latest/ch340_driver.html

zu finden. Die Treibersoftware stellt die Simulation einer seriellen Schnittstelle – ein sogenannter COM Port – in Windows zur Verfügung. Windows identifiziert ggf. mehrere vorhandene COM Ports anhand einer nachgestellten Nummer z. B. COM4. Diese Bezeichnung ist wichtig für die Verwendung in der Arduino IDE, da der IDE Software mitgeteilt werden muss, wo unser zu programmierendes Mikrocontroller-System angeschlossen ist.

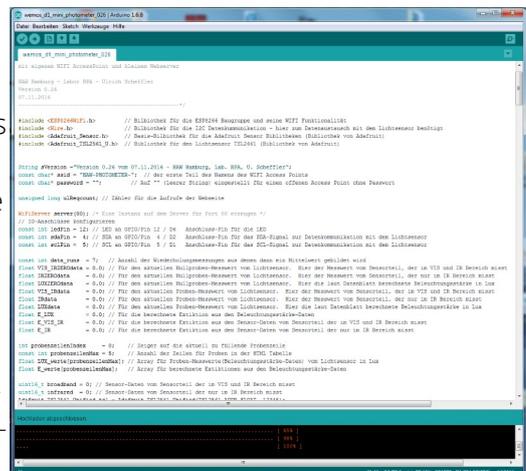


Abbildung 21: Programm einspielen

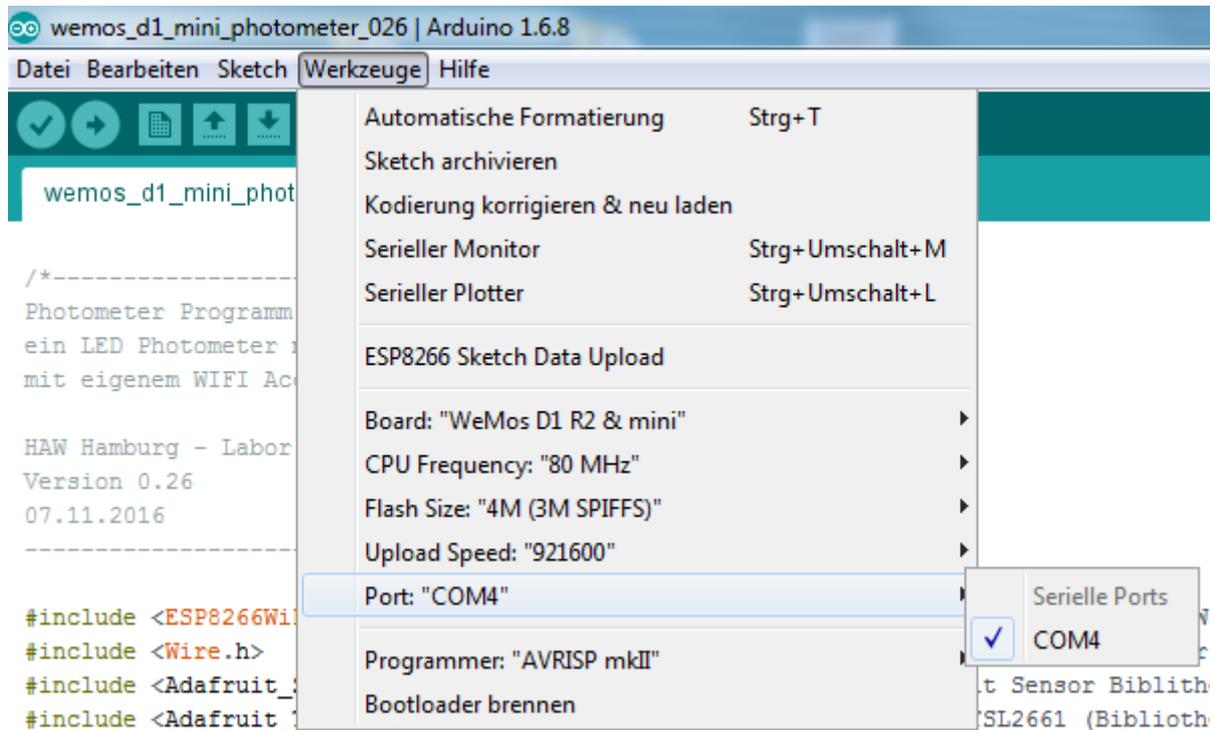


Abbildung 22: Auswahl des Programmieranschlusses

Hierzu im Werkzeug – Menü im Untermenü „Port“ den gewünschten (richtigen) Anschluss auswählen.

Um das Programm zum Mikrocontroller übertragen zu können, muss es zunächst übersetzt werden, d. h. in eine für den Mikrocontroller geeignete Form gebracht werden. Durch drücken des runden Pfeil-Knopfes – in 23 die runde weiße Fläche mit dem Pfeil unterhalb des „B“ vom Bearbeiten Menü – wird das Programm übersetzt und im Anschluss daran an den Mikrocontroller gesendet.



Abbildung 23: Transfer des Programms zum Microcontroller starten

7 Gehäuse(ein)bau

Grundsätzlich lässt sich das Fotometer auch ohne Gehäuse nutzen. Da eine Messung von Licht beabsichtigt ist, kommt es jedoch durch das Umgebungslicht zu Störungen, die die Messung von verlässlichen Ergebnissen deutlich erschweren. Es sind einfache Gehäuse-Varianten aus Papier (nicht so haltbar, wenn Flüssigkeiten im Spiel sind), Holz oder aus nicht mehr benötigten Kunststoffverpackungen (z. B. runde Kaugummi-Dose) einsetzbar.

Für dieses Projekt steht jedoch auch ein Gehäuseentwurf als STL-Datei für den Ausdruck auf einem 3D-Drucker bereit. Dieser Gehäuseentwurf wurde mit dem 3D-Zeichenprogramm OpenSCAD erstellt. Dieses Programm ist freie Software und steht unter der GPL Version 2 Lizenz kostenfrei zum Download unter <http://www.openscad.org> zur Verfügung. Der Entwurf für dieses Gehäuse ist in Form einer Textdatei, die in OpenSCAD eingelesen werden kann, im Anhang unter 11.3 Gehäusemodell-Code für OpenSCAD ab Seite 27 als Text zu finden oder kann beim Autor via email angefragt werden. Somit sind eigene Anpassungen leicht möglich. OpenSCAD kann STL-Datei erzeugen. Diese STL Dateien können dann mit einem Slicer – Programm (es gibt auch hierfür diverse freie Software) für den 3D-Drucker weiter verarbeitet werden, um die Steuerdateien für den 3D-Drucker zu erstellen. Für einen schnellen Einstieg in OpenSCAD bieten sich die folgenden beide deutschsprachigen Einführungen an:



Abbildung 24: Ins Gehäuse einbauen

<https://www.tuebix.org/downloads/tuebix.2015.knopper-openscad.pdf>

<http://campis-fab.blogspot.de/2011/09/ein-kleines-openscad-tutorial.html>

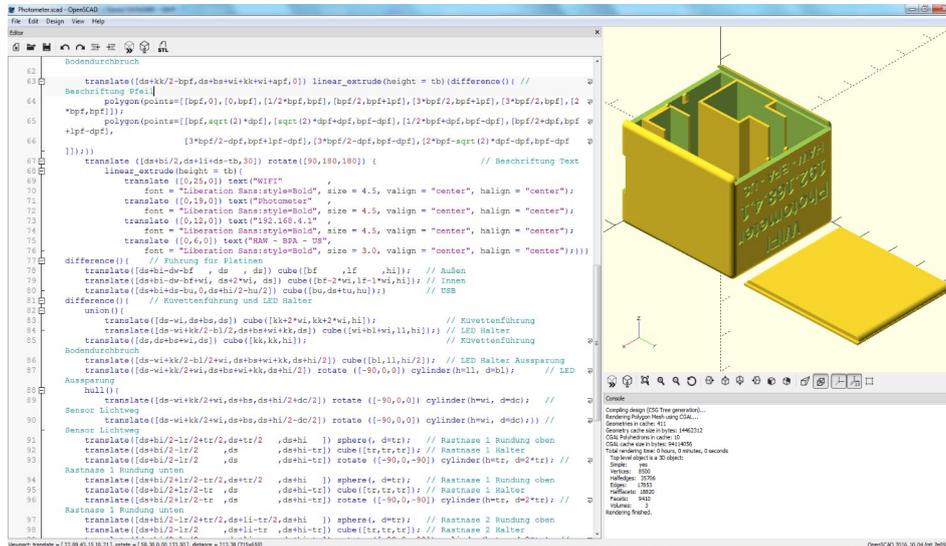


Abbildung 25: Gehäuseentwurf in Open SCAD

Die in den Abbildungen gezeigten Gehäuse wurden aus dem Kunststoff PLA mit einer Schichthöhe von 0.2 mm gedruckt. PLA (Polylactide, umgangssprachlich auch Polymilchsäure genannt) ist sicherlich nicht der beste Kunststoff für den Einsatz in einem Umfeld, in dem Chemikalien eine Rolle spielen können. PLA ist aber sehr einfach zu verarbeiten, gilt als biologisch abbaubarer Kunststoff, kann bis ca 50 °C Umgebungstemperatur verwendet werden und erzeugt auch beim Druckvorgang keine gesundheitsgefährdenden Dämpfe.

8 Inbetriebnahme

Für einen ersten Test muss die Powerbank oder das Netzteil an den USB-Anschluss des Fotometers angeschlossen werden. Die eingebaute LED sollte jetzt für einige Sekunden leuchten und danach wieder ausgehen.



Abbildung 26: Fertig zum Ausprobieren

9 Benutzung

Das Fotometer baut ein eigenes WLAN/WIFI Netzwerk auf. Um sich mit diesem Netzwerk zu verbinden, muss am Smartphone zunächst eine Liste der verfügbaren WLAN/WIFI-Netzwerke aufgerufen werden. Nach Antippen des Eintrags „HAW-Photometer-...“ (statt der Punkte steht die Seriennummer des Fotometers) verbindet sich das Smartphone mit dem Fotometer. Ein Passwort wird nicht abgefragt.

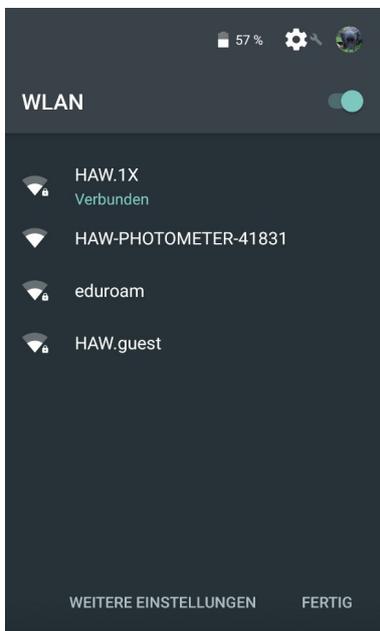


Abbildung 27: Liste der verfügbaren WLAN/WIFI Netzwerke

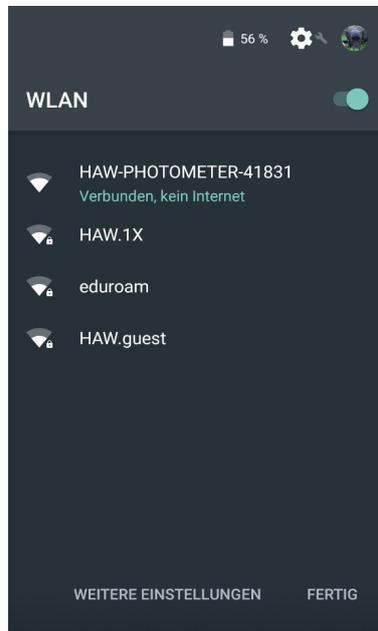


Abbildung 28: WLAN/WIFI Netzwerk auswählen



Abbildung 29: Ggf. Netzwerk ohne Internet zulassen

Einige Smartphones reagieren noch mit einer extra Nachfrage, ob eine Verbindung mit einem Netzwerk ohne Internetverbindung gewünscht ist. Für das Fotometer ist „ohne Internet“ ein Teil des Sicherheitskonzepts und somit ausdrücklich gewünscht.

Im Webbrowser des Smartphones kann jetzt die Webseite des Fotometers aufgerufen werden. Hierzu wird in die Adresszeile des Webbrowsers die IP-Adresse: 192.168.4.1 eingeben und die Webseite abgerufen.

Es wird eine Webseite mit Bedienelementen und einer Tabelle dargestellt.



Abbildung 30: Im Webbrowser Webseite mit IP-Adresse 192.168.4.1 aufrufen



Abbildung 31: Drücken von „Leerprobe“ führt eine Messung aus und trägt die Daten in die Tabelle ein

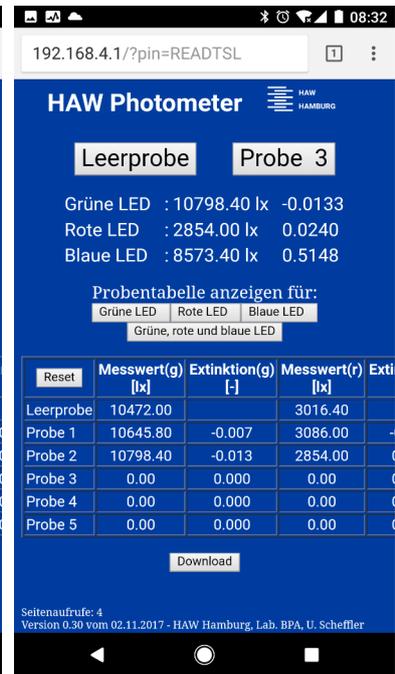


Abbildung 32: Drücken von „Probe“ misst, berechnet die Extinktion und ergänzt die Tabelle



Abbildung 33: Die Tabelle ist breiter und die Anzeige kann seitlich verschoben werden

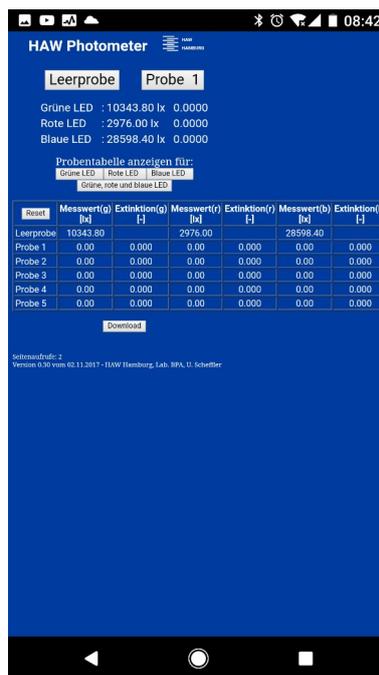


Abbildung 34: Auch Zoomen ist möglich

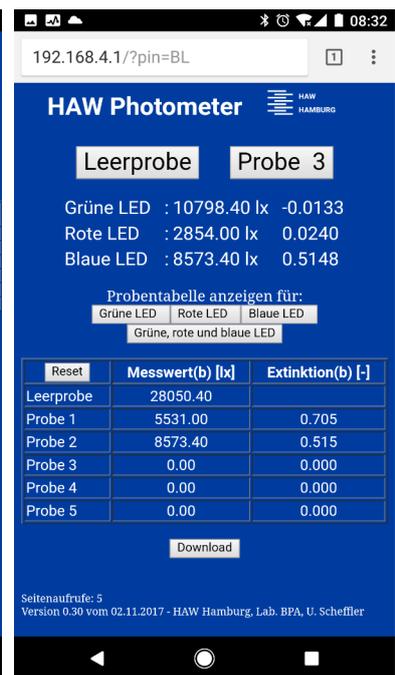


Abbildung 35: Durch Drücken von einem der „... LED „Buttons wird nur der jeweilige Ausschnitt der Tabelle angezeigt

Der Button „Download“ bewirkt ein Herunterladen einer CSV-Datei mit den Daten der Tabelle. Eine CSV-Datei kann in den meisten Tabellenkalkulationsprogrammen einfach weiterverarbeitet werden.

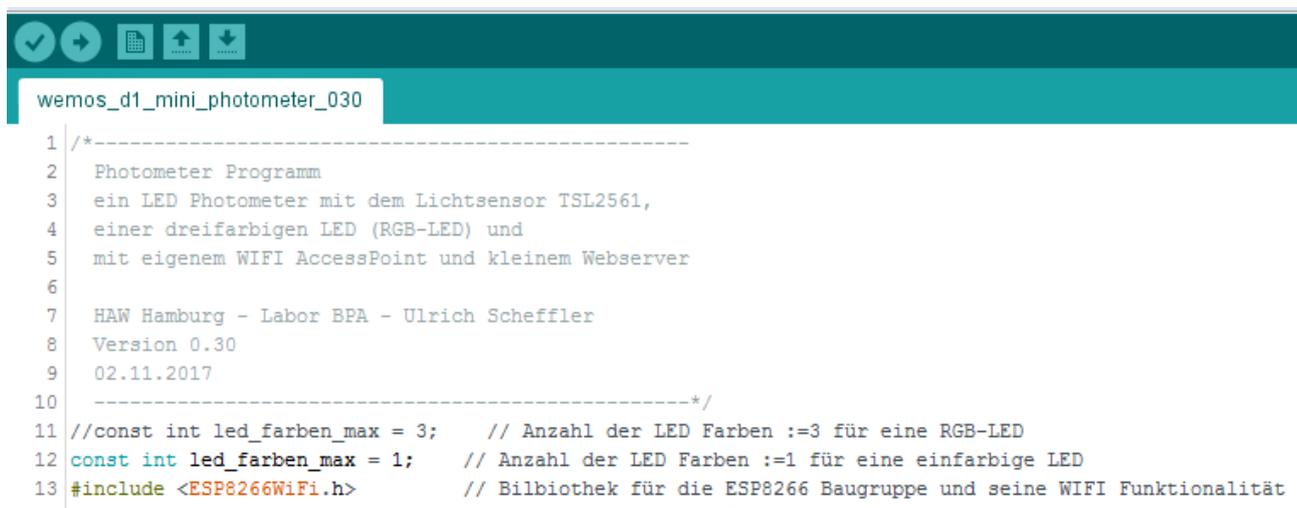
Für die einfarbige Modellvariante fehlen die Auswahlbuttons für die LED Farben.

10 Programmcode

Der Programmcode für das Fotometer mit dem D1 mini wurde in der Arduino IDE erstellt. Die jeweils aktuelle Version wird auf der Projektseite zum Herunterladen angeboten oder kann auch beim Autor via email angefragt werden. Im Folgenden eine kurze Beschreibung, um die Orientierung im Code zu erleichtern und eigene Anpassungen zu erleichtern.

Gleich zu Beginn kann durch Auskommentieren bzw. Kommentieren gewählt werden, ob eine Version für eine einfarbige LED oder eine dreifarbige LED erzeugt werden soll.

Soll eine dreifarbige LED verwendet werden müssen in Zeile 11 die beiden Schrägstriche („//“) am Anfang der Zeile entfernt werden und in Zeile 12 müssen am beginn der Zeile zwei Schrägstriche eingefügt werden. Die beiden Schrägstriche bedeuten, dass die entsprechende Zeile nur Kommentare enthält und nicht als Programmcode gelesen werden sollen.



```
wemos_d1_mini_photometer_030
1  /*-----
2  Photometer Programm
3  ein LED Photometer mit dem Lichtsensor TSL2561,
4  einer dreifarbigen LED (RGB-LED) und
5  mit eigenem WIFI AccessPoint und kleinem Webserver
6
7  HAW Hamburg - Labor BPA - Ulrich Scheffler
8  Version 0.30
9  02.11.2017
10 -----*/
11 //const int led_farben_max = 3;    // Anzahl der LED Farben :=3 für eine RGB-LED
12 const int led_farben_max = 1;    // Anzahl der LED Farben :=1 für eine einfarbige LED
13 #include <ESP8266WiFi.h>          // Bilbiothek für die ESP8266 Baugruppe und seine WIFI Funktionalität
```

Abbildung 36: Programmcode Ausschnitt: Ein- oder dreifarbige LED

Arduino typisch gliedert der Programmcode sich in einen Setup-Teil und einen Loop-Teil.

Der Setup-Teil „void setup()“ wird einmalig beim Programmstart ausgeführt. In ihm werden der WIFI-Accesspoint und der Web Server mit Startparametern versorgt und initialisiert. Zudem wird die LED für einige Sekunden eingeschaltet, um dem Benutzer die Betriebsbereitschaft zu signalisieren. Das Programm erkennt den verwendeten Lichtsensor (TSL2561 oder BH1750) automatisch.

Der Hauptteil des Programms ist die Endlosschleife „void loop()“. Hier wird zunächst geprüft, ob ein Teilnehmer via WIFI verbunden ist und ob eine gültige Anfrage – also eine Anforderung durch den Webbrowser des Benutzers – an den Webserver gestellt wurde. Ist dies der Fall, wird die Anfrage genauer untersucht und in Abhängigkeit dieser Untersuchung begonnen eine Antwort – also eine neue Webseite – des Webserver zusammen zu stellen.

Wurde eine ungültige Anfrage an den Webserver gestellt, wird eine Fehlerseite als Antwort erzeugt.

War die Anfrage gültig, wird noch unterschieden, ob der Bedienknopf „Leerprobe“ oder „Probe“ gedrückt wurde. In beiden Fällen wird eine Messung durchgeführt. Das bedeutet es wird die LED eingeschaltet und Messdaten vom Sensor abgerufen. Die Messung wird siebenmal wiederholt, um ggf. Messfehler und Ausreißer zu unterdrücken – der kleinste und der größte Messwert werden aussortiert und aus den verbleibenden fünf Messwerten wird ein Mittelwert gebildet. Wurde eine Probe vermessen und es stand schon eine Messung einer Leerprobe zur Verfügung wird sodann die Extinktion berechnet.

Alle gemessenen und berechneten Werte werden dann in Textstrings eingebunden aus denen der Antwort-String – im Programmcode „sResponse“ – des Webservers zusammengesetzt wird. Danach wird die Antwort des Webservers d. h. die komplette Beschreibung der Webseite „ausgeliefert“ und die Endlosschleife kehrt an ihren Anfang zur Bearbeitung einer möglichen nächsten Anfrage zurück.

10.1 Benötigte Bibliotheken

Um erweiterte Funktionalitäten einzubinden, werden im Programmcode vier zusätzliche Bibliotheken verwendet. „Wire“ ist Bestandteil der Arduino IDE und wird für die digitale Kommunikationsverbindung mit Zusatzkomponenten – hier der Lichtsensor – benötigt. „ESP8266WiFi“ stammt aus dem installierten ESP8266-Zusatzpaket für die Arduino IDE und stellt die WLAN/WIFI Funktionalitäten zur Verfügung. Die Lizenzbedingungen, auch die der Teilkomponenten, finden sich unter:

<https://github.com/esp8266/Arduino>.

Die beiden Bibliotheken „Adafruit_Sensor“ und „Adafruit_TSL2561“ stammen von der Firma Adafruit und stellen die Funktionalität für den Lichtsensor TSL2561 zur Verfügung. Ihre Lizenzbedingungen finden sich in den Headerfiles.

Hier gibt es Informationen zu den beiden Bibliotheken (auf Englisch):

https://github.com/adafruit/Adafruit_TSL2561/

https://github.com/adafruit/Adafruit_Sensor/

Hier können sie heruntergeladen werden:

https://github.com/adafruit/Adafruit_TSL2561/archive/master.zip

https://github.com/adafruit/Adafruit_Sensor/archive/master.zip

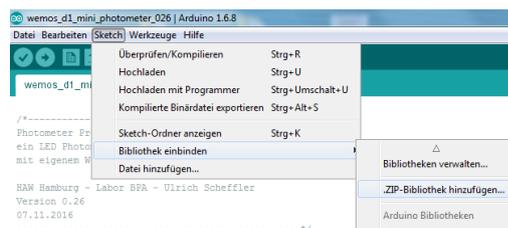
Die Bibliothek „hp_BH1750“ stellt die Funktionalität für den Lichtsensor BH1750 zur Verfügung. Ihre Lizenzbedingungen finden sich in den Headerfiles.

Hier gibt es Informationen zu der Bibliotheken (auf Englisch) und hier kann sie auch heruntergeladen werden (als ZIP File herunterladen!):

https://github.com/Starmbi/hp_BH1750

Um die Bibliotheken für die Arduino IDE verfügbar zu machen, müssen sie an einer bestimmten Stelle im Verzeichnissystem des PCs hinterlegt werden. Am einfachsten gelingt dies folgendermaßen:

1. Die benötigten Bibliotheken als ZIP-Files herunterladen. Achtung: die ZIP-Files **nicht** auspacken, sondern so lassen!
2. In der Arduino IDE im Menü „Sketch“ das Untermenü „Bibliothek einbinden“ anwählen. Im aufklappenden Untermenü den Punkt „ZIP-Bibliothek hinzufügen...“ auswählen.



Daraufhin erscheint ein File-Auswahl-Dialog.

3. Im File-Auswahl-Dialog die gewünschte ZIP-Datei auswählen und „Öffnen“ anklicken.

4. Diese Prozedur (1. bis 3.) für alle gewünschten Bibliotheken wiederholen.

5. Die Arduino IDE beenden und erneut starten.

Die Bibliotheken sind jetzt aus ihren ZIP-Files ausgepackt worden und an der richtigen Stelle im Filesystem für die Verwendung im Fotometer-Programm hinterlegt worden.

11 Anhang

11.1 Aktuelle Anmerkungen und Ergänzungen

Webseite wird nur langsam, unvollständig oder gar nicht dargestellt

Mittlerweile funktionieren auch aktuellere Arduino IDE Versionen (1.8.12 haben wir getestet) mit dem ESP8266-Boards. Aber die zusätzlich benötigten ESP8266 Board-Addons machen mit ihren letzten Revisionen (ab 2.4.1) noch Schwierigkeiten. Die Version 2.4.0 funktioniert aber prima.

$$K = \frac{E_{\lambda}}{c_B}$$

Wie aktiviert man eine bestimmte Version des Board Addons? Das ESP8266-Addon muss wie in der Anleitung beschrieben installiert (ab Seite 7) werden (JSON-String „eintragen“), dann im Boardmanger den Eintrag für den ESP8266 wiederfinden. Jetzt nicht die letzte Version, sondern die Version 2.4.0 auswählen und erst dann „installieren“ auswählen. Hinweis: Das Auswählen einer bestimmten Version ist auch nachträglich möglich.

Aktuell: Mit dem Photometerprogramm ab Version 0.33 funktionieren auch die aktuellen Versionen des ESP8266-Addons!

WiFi Verbindungsaufbau

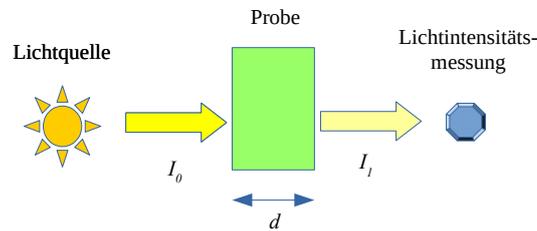
Mit der WiFi Funktionalität gab es bisher keine Schwierigkeiten. Aber da das vom Fotometer aufgebaute WiFi Netz kein Internet anbietet, verhalten sich einige Clients (Handys, Tablets, ...) etwas zickig. Insbesondere Apple Geräte versuchen recht lange doch noch eine Internet-Verbindung herzustellen und erst dann, wenn sie aufgeben, stellen sie dieses Netzwerk auch als VERBUNDEN dem Nutzer zur Verfügung. Dieser Vorgang kann auch gerne mal über eine Minute dauern (halbe Ewigkeit).

3D-Druck des Gehäuses

Tipp für den 3D-Druck des Gehäuses: Vom Lichtsensor-Modul gibt es verschiedene Bauformen. Wenn das Sensormodul bei Aliexpress oder bei „Roboter-Bausatz.de“ gekauft worden ist, ist es vermutlich mit der Typnummer GY-2561 oder GY-302 gekennzeichnet. Diese Module sind sehr gut geeignet, weil sie besonders geringe Abmessungen haben. Aber es gibt zwei verschiedenen Ausführungen: eine mit zwei großen Löchern und eine mit drei großen Löchern in der Platine. Die Löcher sind für das Fotometer nicht entscheidend aber die Position des Lichtsensors auf dem Modul ist leicht unterschiedlich. Im Sourcecode für das Gehäuse sind beide Versionen (2 Löcher und 3 Löcher) zu finden. Die jeweils nicht benötigten Zeilen müssen deaktiviert werden. Auch das Lichtsensor-Modul von Adafruit, das in Deutschland sehr einfach zu bekommen ist, ist geeignet. Es ist jedoch deutlich größer und auch hier muss das Gehäuse entsprechend angepasst werden (siehe Gehäuse-Sourcecode). Extra-Hinweis: Das Adafruit Modul hat auch ein anderes Pin-Layout, sodass auch die selbst zu löttende Adapterplatine etwas anders ausfällt. Auch die Verwendung von Rundküvetten (bzw. Reagenzgläsern) ist möglich. Hierfür sind die benötigten Werte und Programmzeilen im Sourcecode des Gehäusemodells zu finden.

11.2 Lambert-Beer'sches Gesetz

Das Lambert-Beer'sche Gesetz oder Bouguer-Lambert-Beer'sche Gesetz beschreibt die Abschwächung der Intensität des Lichts beim Durchgang durch ein Medium mit einer absorbierenden Substanz unter Berücksichtigung der Konzentration der absorbierenden Substanz und ihrer Schichtdicke.



$$E_{\lambda} = \lg\left(\frac{I_0}{I_1}\right) = \varepsilon_{\lambda} \cdot c \cdot d$$

E_{λ} := Extinktion bei der Wellenlänge λ (Ohne Einheit)

I_0 := Intensität des Lichtes vor der oder ohne die Probe (Einheit: $\text{W}\cdot\text{m}^{-2}$)

I_1 := Intensität des Lichtes nach der Probe (Einheit: $\text{W}\cdot\text{m}^{-2}$)

c := Stoffmengenkonzentration der absorbierenden Substanz in der Flüssigkeit (Einheit: $\text{mol}\cdot\text{l}^{-1}$)

ε_{λ} := dekadischer Extinktionskoeffizient bei der Wellenlänge λ . ε_{λ} ist spezifisch für die absorbierende Substanz. Wird die Konzentration in Mol/l angegeben, so wird ε_{λ} als dekadischer molarer Extinktionskoeffizient notiert (Einheit: $\text{m}^2\cdot\text{mol}^{-1}$).

d := Schichtdicke der Probe Körpers (Einheit: m)

Anwendung in der Fotometrie:

$$E_{\lambda} = \lg\left(\frac{I_0}{I_1}\right) = \varepsilon_{\lambda} \cdot c \cdot d$$

Setzt man voraus, dass sich weder die Schichtdicke d noch die untersuchte Substanz mit ihrem spezifischen Extinktionskoeffizienten ε_{λ} ändert, ergibt sich ein linearer Zusammenhang mit dem Proportionalitätsfaktor K zwischen Konzentration c und der Extinktion E_{λ} .

$$E_{\lambda} = K \cdot c$$

Liegt eine Probe ohne die gesuchte Substanz (Leerprobe) und eine Probe mit bekannter Konzentration c_B vor, kann K über die Gleichung $K = \frac{E_{\lambda}}{c_B}$ berechnet werden. Mit diesem K können nun für weitere unbekannte Proben die Konzentrationen c nach Messung der Extinktion ε_{λ} ermittelt werden.

$$c = \frac{E_{\lambda}}{K}$$

Arbeitsblatt

Name:
LED-Farbe:

Datum:
Fotometer Nr.

Bekannte Proben:

| Konzentration | Testlösung [µl] | Wasser [ml] | Messwert I [LUX] | Extinktion ϵ_λ | K |
|---------------|-----------------|-------------|------------------|-------------------------------|---|
| 0 | 0 | 3,0 | | | |
| 1/30 | 100 | 2,9 | | | |
| 1/10 | 300 | 2,7 | | | |
| 2/5 | 600 | 2,4 | | | |
| 1/3 | 1000 | 2,0 | | | |
| | 3000 | 0,0 | | | |

Unbekannte Proben:

| Proben Nr.: | Messwert I [LUX] | Extinktion ϵ_λ | Konzentration |
|-------------|------------------|-------------------------------|---------------|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

11.3 Gehäusemodell-Code für OpenSCAD

Anmerkung: Das Gehäuse ist passend für das Lichtsensormodul GY-2561 oder GY-302. Wird ein anderes Lichtsensormodul verwendet, muss der Entwurf angepasst werden.

```
// Gehäuse für Photometer mit D1 mini
// U. Scheffler 07.10.2020

$fn=80;
// Die Bauhöhe des Stapels setzt sich aus dem
// D1 mini, der Adapterplatine und dem Lichtsensor zusammen.
// Unter dem D1 Mini sollte etwas Luft bleiben
// Optional mit zweitem Sensor für Fluoreszenzmessung
// LED Lichtaustrittsöffnung anpassbar (15.05.19)
zs = 0.0; // zs=1.0 => mit zweitem Sensor , zs:=0.0 =< ohne zweiten Sensor

hdi = 12.0; // Bauhöhe D1 mini (X)
hap = 3.5; // Bauhöhe der Adapterplatine (Platine und liegende Bauteile) im gestecktem Zustand (X)
//hls = 18.5+zs*4.5; // Bauhöhe des Lichtsensors im gestecktem Zustand (Gy-2561 mit 2 Löchern oder TSL2561 mit 3 Löchern) (X)
hls = 19.5+zs*4.5; // Bauhöhe des Lichtsensors im gestecktem Zustand (Gy-302 / BH1750 mit 2 Löchern) (X)
//hls = 22.5; // Bauhöhe des Lichtsensors im gestecktem Zustand (Adafruit TSL2561 Lux Sensor) (X)
hlu = 1.0; // Luft unter dem D1 mini (X)

//Innenmaße
ds = 1.6; // Dicke Seitenwand (X, Y, Z)
bi = hdi+hap+hls+hlu; // Breite innen (X)
li = 36.0+zs*2.0; // Länge innen (Y)

wi = 0.8; // Wandstärke Innen (X, Y)
hi = 26.0; // Höhe innen (Z)
kr = 1.6; // Kanten Radius
//Deckel
dd = 2.0; // Dicke Deckel (Z)
df = 0.8; // Deckel Führungsnut
fl = 0.2; // Freilauf für Deckelführung
//Rastnasen
tr = 1.2; // Tiefe der Rastnase
lr = 8.0; // Länge der Rastnase
//USB Anschluss
bu = 8.0; // Breite USB (X)
hu = 11.0; // Höhe USB (Z)
tu = 0.4+zs*2.0; // Zusätzliche Tiefe USB (Y)
//Küvette
kmode = 0; // 0:= eckige Küvette , 1:= runde Küvette
// (! Gedruckt ist die Aussparung ca 0.2 mm kleiner als das angegebene Maß!)
kk = 12.6; //13.2; // Kantenlänge bzw Durchmesser für Küvettenaussparung 13.2 eckig (Rohmaß Küvette: 12.3) (X, Y)
//kk = 16.8; // Kantenlänge bzw Durchmesser für Küvettenaussparung 16.8 rund (X, Y)
//Sensor
bs = 5.0+zs*2.0; // Einbaubreite Lichtsensor (Standard = 5.0, Für Adafruit Sensor besser 6.0) (Y)
dc = 3.0; //4.0; // Durchmesser der Öffnung für den Sensor Chip
asc= 12.0; // Abstand der Mitte des Sensorchips von der Unterkante seiner Platine (GY-2561 oder BH1750 mit 2 Löchern) (X)
//asc= 10.0; // Abstand der Mitte des Sensorchips von der Unterkante seiner Platine (TSL2561 mit 3 Löchern) (X)
//asc= 11.2; // Abstand der Mitte des Sensorchips von der Unterkante seiner Platine (Adafruit TSL2561 Lux Sensor) (X)
hsm= hi/2+2.54/2; // Höhe des Sensormittelpunkts (der Sensor ist um einen Halbe Rasterbreite gegenüber dem Baugruppenmittelpunkt verschoben) (Z)
// Führung für Platinen
dw = 5.0; // Dicke für D1 mini (X)
bf = 10.0; // Breite Führungsblock (X)
lf = 8.0+zs*2.0; // Länge Führungsblock (Y)
duw= 3.0; // Dicke unter D1 mini (X)
afu= 30.0+zs*2.0; // Abstand der Führung unter der Platine zur USB Seite (Y)
// LED Führung
bl = 5.5; // Breite des Ausschnitts für die LED (X)
ll = 8.5; // Länge der Auflagefläche für die LED (Y)
lla= 3.0; // Durchmesser Lichtaustrittsöffnung für die LED (X)
// Beschriftungen
tb = 0.4; // Einprägertiefe
bpf= 4.0; // Breite Pfeilschaft
lpf= 10.0; // Länge Pfeilschaft
dpf= 0.8; // Breite der Prägelinie
apf= 1.0; // Abstand Pfeil zum Küvettedurchbruch

// Küvetten Haltefeder
h_khf = ds+hi; // Höhe max (z)
hü_khf = 1.6; // Höhe des an der Küvettenhalterungswand festen Teils (z)
hr_khf = h_khf-hu_khf; // Höhe des runden Teils im mittleren Bereich (z)
ao_khf = 1.0; // Abstand von der Küvettenhalterungswand des geraden Teils im oberen Bereich (x)
t_khf = 2.0; // Tiefe der Feder (Abstand von der Küvettenhalterungswand) (x)
b_khf = 2.0; // Breite der Feder (y)
d_khf = 1.0; // Dicke der Feder (x)
rr_khf = (4* pow(t_khf,2) + pow(hr_khf,2))/(8*t_khf); // Radius des runden Teils im mittleren Bereich

module roundcube(x,y,z,r) { // "cube" mit abgerundeten Kanten und Radius = r
  hull(){
    translate([ r, r, r]) sphere(r=r);
    translate([x-r, r, r]) sphere(r=r);
    translate([ r,y-r, r]) sphere(r=r);
    translate([x-r,y-r, r]) sphere(r=r);
    translate([ r, r,z-r]) sphere(r=r);
    translate([x-r, r,z-r]) sphere(r=r);
    translate([ r,y-r,z-r]) sphere(r=r);
    translate([x-r,y-r,z-r]) sphere(r=r);
  }
}

module haw_logo(h) { //HAW Logo
  b=h/15; // b:=Strichbreite, h:=Logo Höhe
  translate ([ 2*b, 0.5*b,0]) square([11*b,b],center= true);
  translate ([-2*b, 2.5*b,0]) square([11*b,b],center= true);
  translate ([ 2*b, 4.5*b,0]) square([11*b,b],center= true);
  translate ([-2*b, 6.5*b,0]) square([11*b,b],center= true);
  translate ([ 2*b, 8.5*b,0]) square([11*b,b],center= true);
  translate ([-2*b,10.5*b,0]) square([11*b,b],center= true);
  translate ([ 2*b,12.5*b,0]) square([11*b,b],center= true);
  translate ([-2*b,14.5*b,0]) square([11*b,b],center= true);
}
```

```

}

module beschriftungen() { // Beschriftungen an den Gehäuseseiten anbringen
    translate ([ds+bi/2,ds+li+ds-tb,30]) rotate([90,180,180]) { // Hinten - Beschriftung mit HAW Photometer
        linear_extrude(height = tb){
            translate ([-10,19-6.0/2,0]) haw_logo(6);
            translate ([6,19,0]) text("HAW",
                font = "Liberation Sans:style=Bold", size = 6.25, valign = "center", halign = "center");
            translate ([0,12,0]) text("Photometer",
                font = "Liberation Sans:style=Bold", size = 4.5, valign = "center", halign = "center");}}
    translate ([ds+bi/2,tb,30]) rotate([90,180,0]) { // Vorne - Beschriftung mit HAW Logo
        linear_extrude(height = tb){
            translate ([3,15,0]) text("HAW",
                font = "Liberation Sans:style=Bold", size = 7.0, valign = "center", halign = "center");
            translate ([0,6,0]) text("Hamburg",
                font = "Liberation Sans:style=Bold", size = 4.5, valign = "center", halign = "center");
            translate ([0,20,0]) haw_logo(7.0);}}
    translate ([tb,ds+li/2,30]) rotate([270,0,90]) { // Links - In diesem Block ist Platz für eine Gerätenummer oder ein eigenes
Logo
        linear_extrude(height = tb){translate ([0,15,0]) text("",
            font = "Liberation Sans:style=Bold", size = 6.0, valign = "center", halign = "center");}}
    translate ([2*ds+bi-tb,ds+li/2,30]) rotate([90,180,90]) { // Rechts - In diesem Block ist Platz für eine Gerätenummer oder
ein eigenes Logo
        linear_extrude(height = tb){translate ([0,15,0]) text("",
            font = "Liberation Sans:style=Bold", size = 6.0, valign = "center", halign = "center");}}
    }
}

module feder() { // Kuvetten-Haltefeder
    translate ([0,d_khf,0]) rotate([0,0,-90]){
        cube ([d_khf,b_khf,hu_khf]);
        intersection() {
            difference() {
                translate ([t_khf-rr_khf+d_khf,b_khf,hu_khf+hr_khf/2]) rotate([90,0,0]) cylinder (r=rr_khf, h= b_khf);
                translate ([t_khf-rr_khf+d_khf,b_khf,hu_khf+hr_khf/2]) rotate([90,0,0]) cylinder (r=rr_khf-d_khf, h= b_khf);}}
            translate ([0,0,hu_khf]) cube ([t_khf+d_khf,b_khf,hr_khf*5/5]);}}
        hull() {
            translate ([ d_khf,0,0]) cube ([0.1,b_khf,2.85*hu_khf]);
            translate ([-d_khf,0,0]) cube ([d_khf,b_khf, hu_khf]);}}
    }
}

module gehaeuse() { // Gehäuse
    difference() {
        translate ([ 0, 0, 0]) roundcube (ds+bi+ds,ds+li+ds,ds+hi+dd,kr); // Außenhülle
        translate ([ds,ds,ds]) cube ([ bi, li, hi+dd]); // Innenraum
        translate ([ds+bi+ds-bu,0,ds+hi/2-hu/2]) cube ([bu,ds+tu,hu]); // USB Anschluss
        translate ([ds,0,ds+hi]) cube ([ bi, ds+li+ds,dd ]); // Deckelloch
        translate ([ds-df,0,ds+hi]) cube ([df+bi+df,ds+li+ds,dd/2]); // Deckelführung
        if(kmode==0) translate ([ds+(hls-asc-kk/2),ds+bs+wi,0]) cube ([kk,kk,ds]); // eckige Kuvettenführung Bodendurchbruch
        if(kmode==0 && zs>0.0) translate ([ds-wi+(hls-asc-kk/2)-zs*wi/4,ds+bs+wi,0]) cube ([wi/2,kk,hi+2*ds]); // Filteraussparung bei
eckiger Kuvettenführung
        if(kmode==1) translate ([ds+(hls-asc),ds+bs+wi+kk/2,0]) cylinder (d=kk,h=ds); // runde Kuvettenführung Bodendurchbruch
        translate ([ds+kk/2-bpf+(hls-asc-kk/2),ds+bs+wi+kk+apf/2,0]) linear_extrude (height = tb) {difference() { // Beschriftung Pfeil
            polygon (points=[ [bpf,0], [0,bpf], [1/2*bpf,bpf], [bpf/2,bpf+lpf], [3*bpf/2,bpf+lpf], [3*bpf/2,bpf], [2*bpf,bpf] ]);
            polygon (points=[ [bpf,sqrt(2)*dpf], [sqrt(2)*dpf+dpf,bpf-dpf], [1/2*bpf+dpf,bpf-dpf], [bpf/2+dpf,bpf+lpf-dpf],
                [3*bpf/2-dpf,bpf+lpf-dpf], [3*bpf/2-dpf,bpf-dpf], [2*bpf-sqrt(2)*dpf-dpf,bpf-dpf] ]);}}
        beschriftungen();
    }
    difference() { // Führung für Platinen (zwischen den Platinen) mit Aussparung für die USB Buchse
        translate ([ds+bi-dw-bf,ds,ds]) cube ([bf,lf,hi]); // Außen
        translate ([ds+bi-dw-bf+wi,ds,ds]) cube ([bf-2*wi,lf+wi,hi]); // Innen
        translate ([ds+bi+ds-bu,0,ds+hi/2-hu/2]) cube ([bu,ds+tu,hu]); // USB
    }
    difference() { // Kuvettenführung und LED Halter
        union() {
            if(kmode==0) translate ([ds-wi+(hls-asc-kk/2)-zs*wi,ds+bs,ds]) cube ([kk+2*wi+zs*wi,kk+2*wi,hi]); // eckige Kuvet-
tenführung
            if(kmode==1) translate ([ds+(hls-asc),ds+bs+wi+kk/2,ds]) cylinder (d=kk+2*wi,h=hi); // runde Kuvettenführung
            translate ([ds-wi+kk/2-bl/2+(hls-asc-kk/2),ds+bs+0*wi+kk,ds]) cube ([wi+bl+wi,li+wi,hi]); // LED Halter Block
            if(kmode==0) translate ([ds+hls-asc-kk/2-wi,ds+bs+wi+kk,0]) cube ([wi+(kk-bl)/2,4*wi, ds+hi]); // Kuvetten Haltefeder
        }
        Aussparung links
        if(kmode==0) translate ([ds+hls-asc+bl/2,ds+bs+wi+kk,0]) cube ([wi+(kk-bl)/2,4*wi, ds+hi]); // Kuvetten Haltefeder
        Aussparung rechts
        if(kmode==0) translate ([ds+(hls-asc-kk/2),ds+bs+wi,ds]) cube ([kk,kk,hi]); // eckige Kuvettenführung Bodendurchbruch
        if(kmode==1) translate ([ds+(hls-asc),ds+bs+wi+kk/2,ds]) cylinder (d=kk,h=hi); // runde Kuvettenführung Bodendurchbruch
        translate ([ds-wi+kk/2-bl/2+wi+(hls-asc-kk/2),ds+bs+2*wi+kk,ds+hi/2]) cube ([bl,li+2*wi,hi/2]); // LED Halter Aussparung
        translate ([ds-wi+kk/2,wi+(hls-asc-kk/2),ds+bs+(-2)*wi+kk+4*wi,ds+hi/2]) rotate ([-90,0,0]) cylinder (h=li+3*wi, d=bl); //
LED Aussparung
        translate ([ds-wi+kk/2,wi+(hls-asc-kk/2),ds+bs+(-2)*wi+kk+0*wi,ds+hi/2]) rotate ([-90,0,0]) cylinder (h=li+3*wi,
d=lla); // LED Lichtweg Austrittsöffnung
        hull() {
            translate ([ds-wi+kk/2+wi+(hls-asc-kk/2),ds+bs,ds+hsm+dc/2]) rotate ([-90,0,0]) cylinder (h=2*wi, d=dc); // Sensor Licht-
weg
            translate ([ds-wi+kk/2+wi+(hls-asc-kk/2),ds+bs,ds+hsm+dc/2]) rotate ([-90,0,0]) cylinder (h=2*wi, d=dc); // Sensor Licht-
weg
        }
        if(zs>0.0) {
            hull() {
                translate ([ds+(hls-asc)-kk/2-2*wi,ds+bs+wi+kk/2-dc/2,ds+hsm+dc/2]) rotate ([0,90,0]) cylinder (h=3*wi, d=dc); // Zweiter
Sensor Lichtweg
                translate ([ds+(hls-asc)-kk/2-2*wi,ds+bs+wi+kk/2+dc/2,ds+hsm+dc/2]) rotate ([0,90,0]) cylinder (h=3*wi, d=dc); // Zweiter
Sensor Lichtweg
            };
            if(kmode==0) translate ([ds-wi+(hls-asc-kk/2)-zs*wi/4,ds+bs+wi,0]) cube ([wi/2,kk,hi+2*ds]); // Filteraussparung bei eckiger
Kuvettenführung
        };
        if(kmode==0) translate ([ds+hls-asc-kk/2,ds+bs+wi+kk,0]) cube ([ (kk-bl)/2-wi,3*wi, ds+hi+dd]); // Kuvetten Haltefeder Ausspa-
rung links
        if(kmode==0) translate ([ds+hls-asc+bl/2+wi,ds+bs+wi+kk,0]) cube ([ (kk-bl)/2-wi,3*wi, ds+hi+dd]); // Kuvetten Haltefeder Ausspa-
rung rechts
    }
    if(kmode==0) translate ([ds+hls-asc-kk/2+(kk/2-wi-bl/2)/2-b_khf/2,ds+bs+wi+kk+wi,0]) feder(); // Kuvetten Haltefeder links
    if(kmode==0) translate ([ds+hls-asc+kk/2-(kk/2-wi-bl/2)/2-b_khf/2,ds+bs+wi+kk+wi,0]) feder(); // Kuvetten Haltefeder rechts

    translate ([ds+bi-duw,ds+afu,ds]) cube ([duw,wi,hi]); // Platinenführung unter D1 mini
    translate ([ds+bi/2-1r/2+tr/2,ds+tr/2,ds+hi]) sphere (, d=tr); // Rastnase 1 Rundung oben
    translate ([ds+bi/2-1r/2,ds,ds+hi-tr]) cube ([tr,tr,tr]); // Rastnase 1 Halter
    translate ([ds+bi/2-1r/2,ds,ds+hi-tr]) rotate ([-90,0,-90]) cylinder (h=tr, d=2*tr); // Rastnase 1 Rundung unten
    translate ([ds+bi/2+1r/2-tr/2,ds+tr/2,ds+hi]) sphere (, d=tr); // Rastnase 1 Rundung oben
}

```

```

translate([ds+bi/2+lr/2-tr ,ds ,ds+hi-tr]) cube([tr,tr,tr]); // Rastnase 1 Halter
translate([ds+bi/2+lr/2-tr ,ds ,ds+hi-tr]) rotate ([-90,0,-90]) cylinder(h=tr, d=2*tr); // Rastnase 1 Rundung unten
translate([ds+bi/2-lr/2+tr/2,ds+li-tr/2,ds+hi ] ) sphere(, d=tr); // Rastnase 2 Rundung oben
translate([ds+bi/2-lr/2 ,ds+li-tr ,ds+hi-tr]) cube([tr,tr,tr]); // Rastnase 2 Halter
translate([ds+bi/2-lr/2 ,ds+li ,ds+hi-tr]) rotate ([-90,0,-90]) cylinder(h=tr, d=2*tr); // Rastnase 2 Rundung unten
translate([ds+bi/2+lr/2-tr/2,ds+li-tr/2,ds+hi ] ) sphere(, d=tr); // Rastnase 2 Rundung oben
translate([ds+bi/2+lr/2-tr ,ds+li-tr ,ds+hi-tr]) cube([tr,tr,tr]); // Rastnase 2 Halter
translate([ds+bi/2+lr/2-tr ,ds+li ,ds+hi-tr]) rotate ([-90,0,-90]) cylinder(h=tr, d=2*tr); // Rastnase 2 Rundung unten
}

module deckel() { // Deckel
rotate([0,0,90]) translate([0,-(2*ds+li),0]) {
difference() {
union() {
translate([ds+fl,0,0]) cube([bi-2*fl,ds+li+ds,dd-kr ]); // Deckel
difference() { // mit Abrundung an Kante
hull() {
translate([ds+fl,0+kr ,dd-kr]) rotate([0,90,0]) cylinder(h=bi-2*fl, r=kr);
translate([ds+fl,ds+li+ds-kr,dd-kr]) rotate([0,90,0]) cylinder(h=bi-2*fl, r=kr);
translate([ds-df+fl-20,0-20,-20]) cube([df+bi+df-2*fl+40,ds+li+ds+40,dd/2-2*fl+20]);
translate([ds-df+fl,0,0]) cube([df+bi+df-2*fl,ds+li+ds,dd/2-2*fl]); // Deckelführung
translate([ds+bi/2-lr/2-fl,ds ,0]) cube([lr+2*fl,tr,tr]); // Aussparung Rastnase
translate([ds+bi/2-lr/2-fl,ds+li-tr,0]) cube([lr+2*fl,tr,tr]); // Aussparung Rastnase
}
translate ([ds+bi/2-10,4.5+5,dd-tb]) rotate([0,0,-45]) { // Beschriftung Text
linear_extrude(height = tb){
translate ([0,15,0]) text("192.168.4.1" , ,
font = "Liberation Sans:style=Bold", size = 6, valign = "center", halign = "center");}
}
}
}
}

module kappe_eckig() { // Kappe zum Abdecken einer eckigen Küvette
wand = 1.2;
innenkante = 14.0;
innenhoehe = 18.5;
difference() {
cube (size = [innenkante + 2*wand,innenkante + 2*wand,innenhoehe + wand]);
translate ([wand,wand,wand]) cube (size = [innenkante,innenkante,innenhoehe+2*wand]);
translate ([wand+innenkante/2,wand+innenkante/2,-wand]) cylinder(d=wand ,h =innenhoehe*2, $fn=50);
}
}
// *****
anzahlSpalten = 1; // Photometer Anzahl pro Zeile
druckabstand = 2; // Abstand der einzelnen Teile zueinander
kappe=16.4; //Kantenlänge der Kappen

//for (i = [0:anzahlSpalten-1]) {
// translate([i*(ds+bi+ds+druckabstand) ,0*(ds+li+ds+druckabstand),0]) kappe_eckig();
// translate([i*(ds+bi+ds+druckabstand)+ ds+bi+ds-kappe,0*(ds+li+ds+druckabstand),0]) kappe_eckig(); // Zweitete Kappe
// translate([i*(ds+bi+ds+druckabstand) , (1-1)*(ds+li+ds+druckabstand)+kappe+druckabstand,0]) gehaeuse ();
// translate([i*(ds+bi+ds+druckabstand) , (2-1)*(ds+li+ds+druckabstand)+kappe+druckabstand,0]) gehaeuse ();
// translate([i*(ds+bi+ds+druckabstand) , (3-1)*(ds+li+ds+druckabstand)+kappe+druckabstand,0]) deckel ();
// translate([i*(ds+bi+ds+druckabstand) , (3-1)*(ds+li+ds+druckabstand)+(df+bi+df-2*fl)+druckabstand+kappe+druckab-
stand,0]) deckel ();
//
//}

for (i = [0:anzahlSpalten-1]) {
translate([i*(ds+bi+ds+druckabstand),0*(ds+li+ds+druckabstand),0]) gehaeuse ();
translate([i*(ds+bi+ds+druckabstand),1*(ds+li+ds+druckabstand),0]) deckel ();
translate([i*(ds+bi+ds+druckabstand),2*(ds+li+ds+druckabstand),0]) kappe_eckig ();
}
*difference() { // Gehäuse im Anschnitt für grafische Darstellungen
gehaeuse ();
translate([ds+hls-asc,0,0]) cube([ds+bi+ds,ds+li+ds,ds+hi+dd]);
}
}

```

