

Setalight

Technische Dokumentation

25. Juni 2019

Tristan Harders (Medientechnik, 2292908)

Juliane Kuhnle (Medientechnik, 2300095)

Paul Giese (Media Systems, 2270913)

Das Ziel unseres Projektes ist es, eine Anwendung zu entwickeln, die mit Hilfe eines Helligkeitssensors und einer Kamera ein Objekt oder eine Person automatisch gleichmäßig beleuchtet. Zu diesem Zweck dient eine weiße Styroporkugel als Lichtdouble, die nach Lichtsetzung entfernt wird. Des weiteren kann man alle Scheinwerfer auch einzeln ansteuern, um die Kugel nach eigenen Wünschen zu beleuchten.

1. Aufbau und Schaltpläne

Der Aufbau gestaltet sich folgendermaßen:

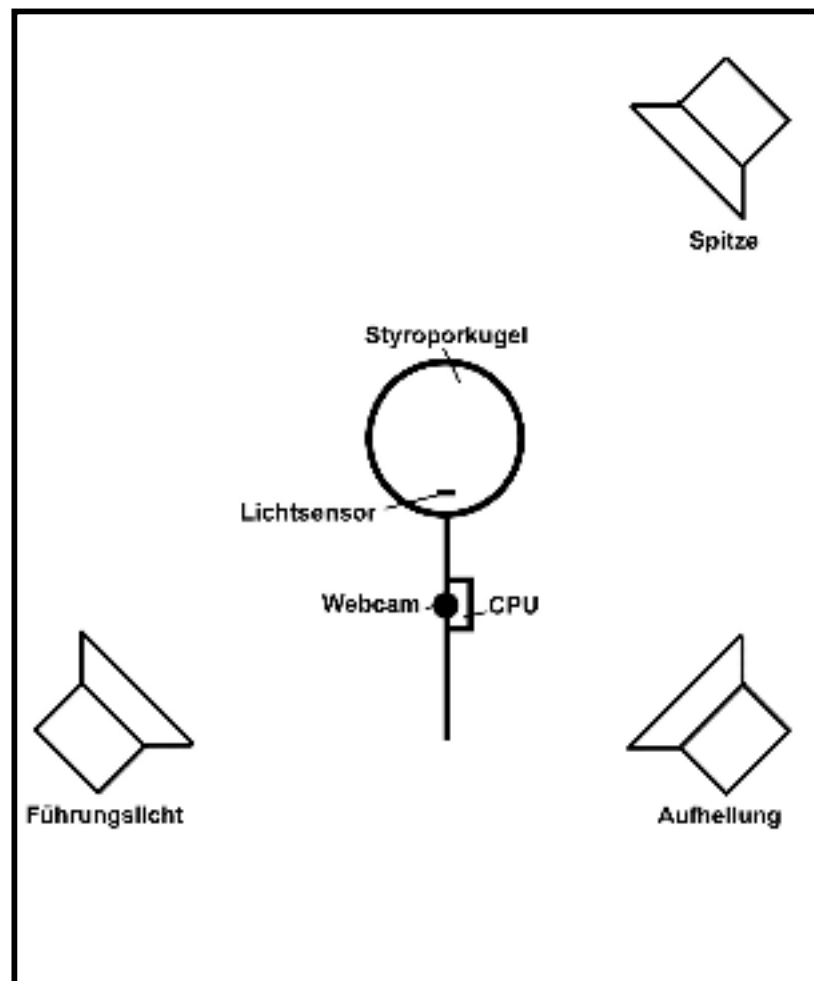


Abb. 1: Aufbau

Die Styroporkugel ist auf einem Stativ zusammen mit der Box und einer Webcam angebracht. Um die Kugel herum sind 3 Scheinwerfer in einer klassischen 3-Punkte-Beleuchtung aufgebaut (Führung, Aufhellung und Spitze). Über das Smartphone kann zwischen Automatik und manuellem Modus entschieden werden.

Die Kommunikation und Verschaltung der Steuerungseinheit sieht folgendermaßen aus:

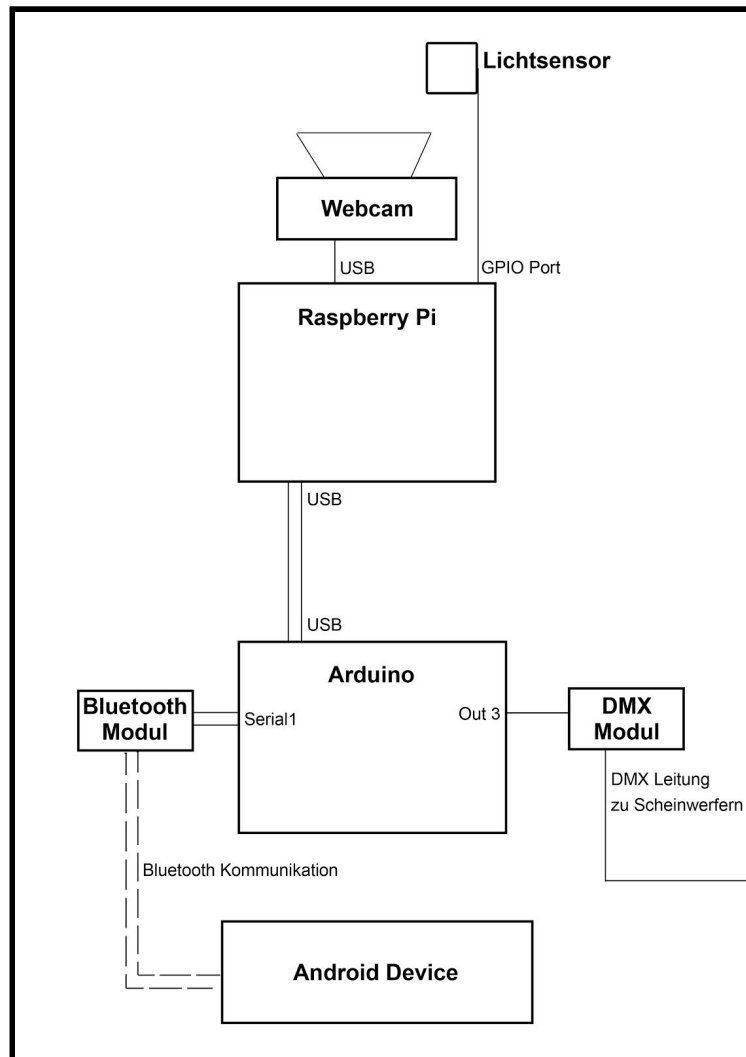


Abb. 2: Schaltplan

Vom Smartphone aus wird ein Befehl gesendet, um die Bildverarbeitung und Messung durch den Raspberry Pi zu starten.

Mit einer einfachen Webcam wird ein Foto gemacht und die Helligkeitswerte ausgelesen.

Da die Webcam automatisch die Helligkeit des Bildes anpasst, müssen wir zusätzlich die tatsächliche Helligkeit am Objekt mittels eines Lichtsensors messen und vergleichen. Die daraus berechneten realen Werte werden über den Arduino und das Bluetooth-Modul an das Smartphone übertragen und von der App verarbeitet.

Das Smartphone gibt daraufhin über Bluetooth den Steuerbefehl mit den verrechneten Werten zurück an den Arduino, der wiederum die Änderung über DMX veranlasst.

2. Programmierung Raspberry Pi und Arduino

Zunächst haben wir uns mit der Bildverarbeitung beschäftigt. Dafür verwenden wir *OpenCV*, dessen Installation sehr viel mehr Zeit in Anspruch genommen hat als geplant. Über *fswebcam* wird der Webcam der Befehl gegeben, ein Foto zu machen. Die Funktion *HoughCircles* dient zur Kreiserkennung. Auf dem Kreis haben wir drei Messpunkte, entsprechend der Scheinwerferpositionen, definiert. Die Punkte werden in RGB-Werten ausgegeben, deshalb müssen sie zunächst mit Hilfe der Formel $Y=0,3*R+0,59*G+0,11*B$ in Helligkeitswerte umgerechnet werden.

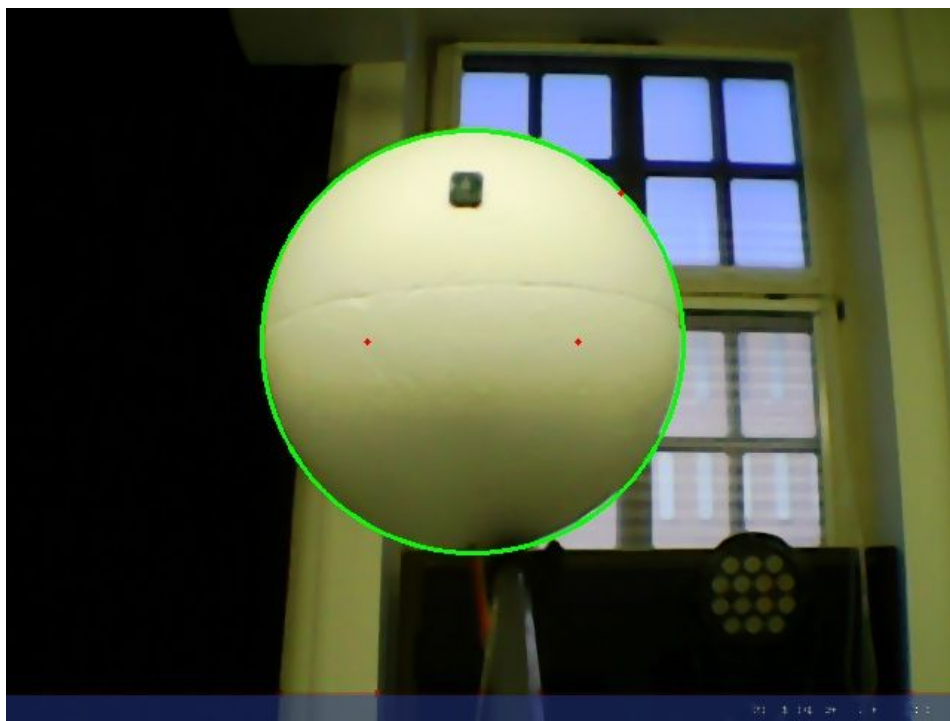


Abb. 3: Webcam-Foto mit erkanntem Kreis (grün) und drei Messpunkte (rot)

Für den Helligkeitssensor *TSL2561* haben wir im Internet ein fertiges Skript gefunden das die sichtbare Lichtstärke in Lux ausgibt. Dieses Skript haben wir übernommen. Aus dem ausgegebenen Wert erstellen wir einen Faktor, mit dem dann die umgerechneten Helligkeitswerte der Kugel automatisch angepasst werden. Die optimale Helligkeit für eine normale Interviewsituation beträgt 500 lux, daher lautet unsere Formel $\text{Faktor} = 500 / (\text{gemessene Lichtstärke})$. Liegt die gemessene Lichtstärke über dem optimalen Wert, werden die Messwerte der Bildpunkte runtergestuft. Liegt sie darunter, werden sie entsprechend hochgestuft.

Der Raspberry Pi schreibt also die Helligkeitswerte hintereinander in einen String, der dann an den Arduino gesendet wird. Der Arduino leitet diesen einfach weiter an das Bluetooth-Modul *HM10*, welches die Daten wiederum ans Smartphone schickt.

Als etwas komplizierter erwies sich der Signalweg in die andere Richtung. Vom Smartphone aus sollte zum einen ein Befehl gesendet werden, der das Skript auf dem Raspberry Pi startet, das die Messungen vornimmt. Zum anderen sollten die Steuerbefehle für die Scheinwerfer auf DMX umgesetzt und an selbige weitergeleitet werden. Wir haben also ein Skript geschrieben, welches erkennt ob es sich bei dem eingehenden String um den Befehl zum Starten des Skriptes oder um Steuerbefehle für die Lampen handelt. Der Startbefehl besteht rein aus Buchstaben (dem Wort „setalight“), während es sich bei den Steuerbefehlen nur um Zahlen handelt. Das Skript erkennt hieran, um was es sich handelt und leitet den Startbefehl direkt über den Seriellen Port an den Raspberry Pi weiter. Die Steuerbefehle hingegen werden über die Library *DMXSimple* in DMX-Kanäle geschrieben und an das DMX-Modul gesendet. Die Steuerbefehle bestehen aus 27-stelligen Zahlenstrings (jeweils 3 Werte pro Lampe, diese in 8 bit).

3. Programmierung Android-App

Die App zur Steuerung der Scheinwerfer hat mehrere Funktionen. Als Erstes soll der User aktive Bluetooth-Geräte in seiner Umgebung scannen und Setalight auswählen. Hierzu klickt sie/ er auf einen Button oben rechts in der Ecke. Danach erscheinen alle verfügbaren Bluetooth-Geräte in der Nähe. Hierbei sucht die App aber nicht nach allen Bluetooth-Geräten, sondern nur nach solchen, die auch eine BLE-Verbindung, also eine “Bluetooth Low Energy”-Verbindung unterstützen. Diese Verbindung hat den Vorteil, dass sie weniger Energie bei dauerhaftem Betrieb verbraucht.

Sobald der User unter den verfügbaren Geräten “Setalight” auswählt, verbinden sich die Geräte und die App zeigt mehrere Einstellungsmöglichkeiten für das sogenannte Standard Preset. Dieses ist ausgelegt für eine einfache 3-Punkte Beleuchtung. In späteren Updates soll der User auch verschiedene Presets auswählen können.

Zu der 3-Punkte-Beleuchtung gehören 3 Scheinwerfer, die nun in der App angezeigt und eingestellt werden können. Es lassen sich die Helligkeit, Sättigung und die Farbe der Scheinwerfer manuell einstellen. Zudem gibt es zwei weitere Buttons.

Mit dem Button „Lampen an“ stellen sich die Lampen auf einen einheitlichen Wert. So lässt sich testen, ob alles richtig verbunden ist und die Lampen richtig eingestellt sind. Außerdem ist es empfehlenswert, die Scheinwerfer mit diesem Button einheitlich einzustellen, da so die automatische Einstellung genauer funktioniert.

Der andere Button aktiviert die automatische Einstellung, bei der das Signalwort “setalight” an den Arduino gesendet wird (s.o.). Daraufhin erhält die App einen 9-stelligen String aus Zahlen, die die realen

Helligkeiten im Raum darstellen (3 Zahlen/ Punkt). Diese Zahlen werden mit von uns festgelegten optimalen Werten für die jeweiligen Punkte verglichen und verrechnet. Hieraus ergibt sich ein Faktor, mit dem die derzeitigen Lampeneinstellungen multipliziert werden. Die daraus resultierenden Werte werden nun wieder an den Arduino gesendet und die Scheinwerfer stellen sich ein.

Um hierbei ein optimales Ergebnis zu erzielen, sollten die Scheinwerfer zuvor mit dem anderen Button einheitlich eingestellt sein, da zum Beispiel veränderte Farben die Genauigkeit der Rechnung beeinträchtigt. Die App ist in der Programmiersprache Kotlin entwickelt.

4. Schwierigkeiten und Fehleranalyse

Wir hatten durchgehend mit schwankenden Helligkeitswerten und fehlerhaften Lampeneinstellungen zu kämpfen. Wir erklären uns dies durch die unberechenbare Blendenautomatik der Webcam und mangelnde Qualität des Sensors. Eine Lösung wäre, statt der Messpunkte durch die Webcam mehrere hochwertige Sensoren anzubringen, da diese ein verlässlicheres Ergebnis liefern.

Der ursprüngliche Plan sah für die Jahresausstellung vor, die Lampen auch live einzustellen. Dadurch wäre dem Besucher klarer gewesen, dass etwas passiert und wie sich Änderungen des Umgebungslichts auch im Moment auswirken. Leider war es nicht möglich so viele Bilder auszuwerten und die Latenz ist immens gestiegen. Dies hat eine Live-Analyse wiederum zwecklos gemacht.

Eine weitere große Hürde war die Einrichtung der Bluetooth-Verbindung von Seiten der Anwendung. Wir haben uns leider deutlich länger als geplant mit der Behebung dieses Problems beschäftigen müssen.

Letztlich haben wir frei verfügbaren Code von Kai Morich genutzt, die er auf Github zur Verfügung stellte, um die Verbindung aufzubauen. Daher ist die App nun eine Fusion aus seiner Bluetooth-Verbindungs-Anwendung und unserer App.

Dazu war noch einer unser Programmierer im Laufe des Projektes abgesprungen.

5. Fazit

Trotz der vielen Hürden und demotivierenden Probleme mit denen wir zu kämpfen hatten, sind wir letztlich zufrieden mit unserem Resultat. SetALight ist ein ausbaubarer Prototyp geworden, der durchaus das Interesse der Besucher des Rundgangs geweckt hat und mit mehr Zeit und Energie zu einer praktischen Beleuchtungs-App werden kann. Es gab vermehrt Anstöße, die App nicht nur für Bühnenbeleuchtung, sondern auch für allgemeine Beleuchtung in privaten Räumen zu nutzen um zum Beispiel angenehmes ambientes Licht zu kreieren. Es hat Freude bereitet, zu sehen, dass einige Leute sich für unsere Anwendung begeistern konnten und Verbesserungsvorschläge unterbreitet haben.

