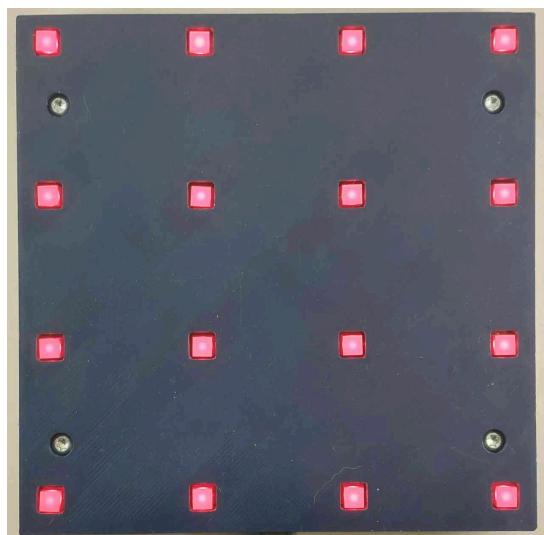


# **Entwicklung eines automatischen Timing-Messgeräts für Kameras**

**Pho21 Bachelorthesis der Fachhochschule Graubünden  
in Zusammenarbeit mit Astrivis Technologies Ltd**

**15. Oktober 2024**



Autor: Raphael Seitz  
E-Mail: r.seitz2001@gmail.com

Referent: Prof. Dr. Philipp Roebrock, FH Graubünden,  
Institut für Photonics und Robotics

Korreferent: Olivier Saurer, Astrivis Technologies Ltd

Studiengang: BSc Photonics, Institut für Photonics und Robotics, FH Graubünden

Start der Thesis: 19.02.2024

Abgabe der Thesis: 09.08.2024

## Zusammenfassung

Viele Computervision-Anwendungen sind auf ein korrektes Timing der aufgenommenen Bilder angewiesen. Es wurde ein Timing-Messgerät entwickelt, mit welchem die Bildwiederholrate und Belichtung über die Zeit detektiert werden können. Dies geschieht über das Analysieren der Änderung vom Wandern lassen einer LED in einer 4x4 Anordnung. Das Gerät lässt sich über ein USB-Kabel mit einem Computer verbinden und über die serielle Schnittstelle konfigurieren. Während dem Messvideo muss die Frequenz des Messgerätes leicht von der Kamera abweichen und die Position und die Lichtverhältnisse so unveränderlich wie möglich bleiben. Die Auswertung geschieht im Nachhinein durch eine Software geschrieben in Python. Darin wird automatisch die Position des Gerätes gefunden und die Helligkeiten der LEDs kalibriert, um robuste Resultate zu erhalten.

Dies funktioniert erst für ein Messsystem bestehend aus je einer Kamera mit Global Shutter und einem Messgerät. Eine Kommunikation zwischen Messgeräten ist gegeben aber noch nicht implementiert. Die Auswertung kann und darf derzeit nur ein Messgerät detektieren. Im Fall, wenn zwei Kameras dasselbe Messgerät zur gleichen Zeit filmen, geschieht während der Auswertung kein Austausch der Messwerte zwischen den einzelnen Videos.

## Inhaltsverzeichnis

<b>1 Abkürzungsverzeichnis</b>	<b>5</b>
<b>2 Einleitung</b>	<b>6</b>
2.1 Begründung des Themas . . . . .	6
2.2 Aufgabenstellung . . . . .	6
2.3 Ziele . . . . .	6
2.4 Systemabgrenzung . . . . .	6
2.5 Relevante Arbeiten . . . . .	7
2.6 Fachliche Grundlagen . . . . .	7
2.6.1 Pulsweitenmodulation . . . . .	7
2.6.2 Timer im Mikrokontroller . . . . .	7
2.6.3 Rolling und Global Shutter . . . . .	8
<b>3 Hauptteil</b>	<b>9</b>
3.1 Material und Methoden . . . . .	9
3.1.1 Technische Ausrüstung & Verwendete Tools . . . . .	9
3.1.2 Entwicklungsprozess der Hardware . . . . .	9
3.1.3 Entwicklungsprozess der Software . . . . .	9
3.1.4 Entwicklungsprozess des Gehäuses . . . . .	10
3.2 Kalibrationsgerät . . . . .	10
3.2.1 Blockschaltbild . . . . .	10
3.2.2 Hard- & Software . . . . .	11
3.2.3 Anordnung der Leuchtdioden . . . . .	11
3.2.4 Kodierung der Zeit . . . . .	12
3.2.5 Infobefehle . . . . .	12
3.2.6 Konfigurationsmöglichkeiten . . . . .	12
3.2.7 Modi . . . . .	12
3.3 Rechenmethoden . . . . .	13
3.4 Messmethoden . . . . .	13
3.4.1 Machbarkeitsanalyse der Kalibration . . . . .	13
3.4.2 Überprüfung von Frequenzen . . . . .	14
3.4.3 Aufnahme Video . . . . .	14
3.5 Auswerteverfahren . . . . .	15
3.5.1 Reihenfolge der Modi . . . . .	15
3.5.2 Detektion & Ausrichtung des Orientierungsmuster . . . . .	16
3.5.3 Übermitteln von Einstellungen & Kalibration der Leuchtdioden-Intensität . . . . .	17
3.5.4 Zentrum der laufenden Leuchtdiode . . . . .	19
3.5.5 Breite der laufenden Leuchtdiode . . . . .	19
3.6 Ergebnisse . . . . .	19
3.6.1 Frequenzlimitationen . . . . .	19
3.6.2 Frequenzzuverlässigkeit . . . . .	19
3.6.3 Gut eingestellt . . . . .	20
3.6.4 Schlecht eingestellt . . . . .	21
3.6.5 Einfluss der Bildqualität . . . . .	21
3.6.6 Einfluss der Kalibration . . . . .	22
<b>4 Schlussteil</b>	<b>23</b>
4.1 Zusammenfassung der Ergebnisse . . . . .	23

4.2 Interpretation der Resultate . . . . .	23
4.3 Kritische Beurteilung . . . . .	23
4.4 Offene Fragen . . . . .	23
4.5 Ausblick . . . . .	23
<b>5 Quellenverzeichnis</b>	<b>25</b>
<b>6 Abbildungs- &amp; Tabellenverzeichnis</b>	<b>27</b>
<b>7 Anhang</b>	<b>28</b>

## 1 Abkürzungsverzeichnis

USB	Universal Serial Bus
LED	Leuchtdiode
PC	Personal Computer
PWM	Pulsweitenmodulation
PCB	Printed Circuit Board (Leiterplatine)
QR-Code	quick-response code
FPS	Bilder pro Sekunde (Frequenz)
LSB	Least Significant Bit (Niederwertigstes Bit)

Tabelle 1: Abkürzungsverzeichnis

## 2 Einleitung

### 2.1 Begründung des Themas

Viele Computervision-Anwendungen, welche Einzel- und Mehrkamerasysteme verwenden, sind auf ein korrektes Timing der aufgenommenen Bilder angewiesen. Dabei gehen die Bilder von der Kamerahardware über den Übertragungsweg zum Auswertesystem (USB (Universal Serial Bus), Ethernet, WiFi, ...) und durch verschiedene Treiberschichten bis in die Anwendung einen langen Weg. Zwar bieten die Hersteller manchmal einen Zeitstempel für jedes aufgenommene Bild. Trotzdem wäre ein automatisches Messgerät praktisch, mit dem das Zeitverhalten der Aufnahmen (z.B. Bilder pro Sekunde, Jitter) oder die Synchronität von Mehrkamerasystemen bestimmt werden können.

### 2.2 Aufgabenstellung

Das Messsystem wird als Elektronik ausgelegt, welche mehrere LEDs (Leuchtdioden) zur Anzeige hat. Die Kamera schaut auf die LEDs und nimmt diese auf. Das Messsystem verändert die LEDs zeitabhängig und kodiert einen Zeitstempel in die Zustände der LEDs. Mittels einer Software sollte man anhand der von der Kamera aufgenommenen Bilder die Zeitstempel rekonstruieren können und mit diesen das Zeitverhalten der Kamera analysieren können.

Das Messsystem sollte sich per USB mit einem PC (Personal Computer) verbinden lassen und von diesem per serielle Schnittstelle konfigurieren lassen (z.B. Messfrequenz). Es wäre auch gut, wenn das Messsystem einen Triggerausgang und Triggereingang hätte mit dem sich Kameras oder das Messsystem selbst triggern lassen.

### 2.3 Ziele

- Mit welcher Anordnung, bzw. Ansteuerung der LEDs kann geschickt Zeit kodiert und detektiert werden?
- Welche Kamerafehler und Aufnahmeeigenschaften können detektiert werden?
- Wie grob weichen die vom Hersteller versprochenen Aufnahmeeigenschaften ab?
- Welche Einflüsse hat eine variierende Bildqualität?

### 2.4 Systemabgrenzung

Als erstes wurde eine Systemabgrenzung erstellt, welche in Abbildung 1 zu sehen ist.

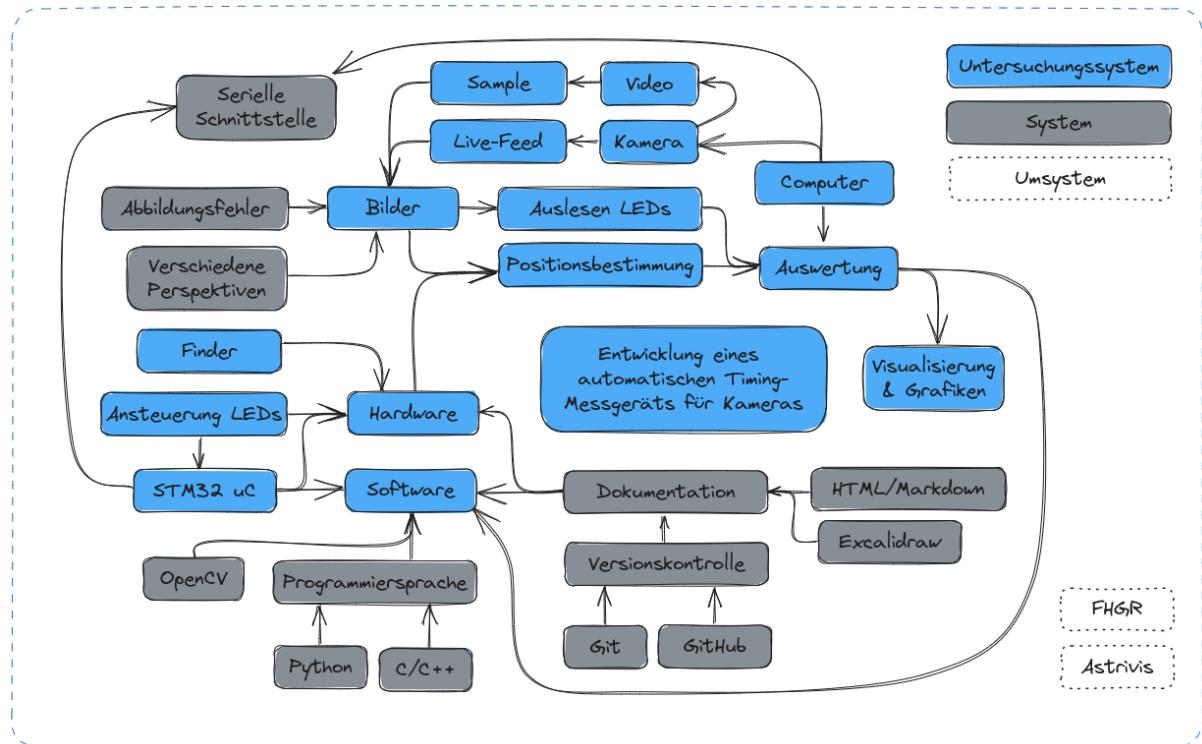


Abbildung 1: Systemabgrenzung

## 2.5 Relevante Arbeiten

Zeitstempel können als Gray-Codes codiert werden. Zur Decodierung und Analyse einer Kamera wird jedoch eine zusätzliche externe Kamera verwendet, die das System von Außen beobachtet [1]. Die Position, inspiriert durch Data Matrizen [2], wird anhand der Anordnung der LEDs bestimmt.

## 2.6 Fachliche Grundlagen

### 2.6.1 Pulsweitenmodulation

Die Helligkeit einer LED kann durch Variation der Eingangsleistung verändert werden. Oft kann eine LED jedoch nur ein- oder ausgeschalten werden, welches eine Änderung der Helligkeit durch ändern der Stromstärke schwierig gestaltet. Die Eingangsleistung wird demnach über die Zeit durch PWM (Pulsweitenmodulation) des digitalen Ansteuerungssignales kontrolliert reguliert.

Ein Beispiel: Wenn eine LED wiederholend innert kurzer Zeit 25% an und den Rest der Zeit aus ist, sieht es für einen Betrachter so aus, als leuchte die LED mit einem viertel der vollen Intensität. Dies nennt sich auch 25% Duty Cycle.

### 2.6.2 Timer im Mikrocontroller

Die Aufgabe eines Timers in einem Mikrocontroller ist das Zählen auf eine bestimmte Zahl. Vorgegeben ist auch die Eingangsgeschwindigkeit und ein Prescaler, welcher die Eingangsgeschwindigkeit dividiert und so die Zählergeschwindigkeit vorgibt. Durch zurücksetzen des Zählers beim Erreichen des Zielwertes wird fortlaufend gezählt.

Der Zeitpunkt beim Erreichen der vorgegebenen Zahl kann somit als variable Frequenzgeneration eingesetzt werden. In (1) sind  $z_{ARR}, z_{PSC} \in \mathbb{N}$  die Werte vom Auto-Reload- & Prescaler-Register. Die Eingangs frequenz ist durch  $f_{Eingang}$  gegeben und die resultierende variable Ausgangsfrequenz ist  $f_{Timer}$ .

$$f_{Timer} = \frac{f_{Eingang}}{(z_{ARR} + 1) \cdot (z_{PSC} + 1)} \quad (1)$$

### 2.6.3 Rolling und Global Shutter

Das Auslesen der Pixel einer Kamera wird unterschieden zwischen Rolling und Global Shutter. Beim Global Shutter werden alle Pixelwerte gleichzeitig ausgelesen. Anders ist dies beim Rolling Shutter, wo zeilenweise Pixelwerte ausgelesen werden [3].

### 3 Hauptteil

#### 3.1 Material und Methoden

##### 3.1.1 Technische Ausrüstung & Verwendete Tools

Alles was verwendet wurde ist in Tabelle 2 aufgelistet.

<b>Programme</b>	
KiCAD [4]	Schema- & Leiterplatten-Designer
STM32CubeIDE [5]	Programmierumgebung von ST
Fusion 360 [6]	3D-Konstruktions-Software
Prusa Slicer [7]	Preparation von Objekten zum 3D-Druck
LibreOffice [8]	Präsentationen (Exposé)
MobaXTerm [9]	Kommunikation über serielle Schnittstellen
Excalidraw [10]	Diagramme
Python [11]	Programmiersprache
C/C++ [12]	Programmiersprache
<b>Programmier-Bibliotheken</b>	
opencv [13]	Bildverarbeitungs-Bibliothek
numpy [14]	Numerische Rechenbibliothek
matplotlib [15]	Bibliothek zum grafischen darstellen von Daten
scipy [16]	Algorithmen für Wissenschaftliches Rechnen
<b>Allgemeines</b>	
vim [17], neovim [18]	Text- & Code-Editor
git [19], GitHub [20]	Versionierungs-Software & Datenbank
Linux [21], Windows [22]	Betriebssystem
<b>Hardware</b>	
Keysight MSOX3024T [23]	Oszilloskop
Prusa MK4 [24]	3D-Drucker
IDS UI-3250CP-M-GL [25]	Kamera
AOS U750 COLOR [26]	Kamera
Samsung Galaxy A52 [27]	Kamera

Tabelle 2: Technische Ausrüstung & Verwendete Tools

##### 3.1.2 Entwicklungsprozess der Hardware

Die Hardware wurde iterativ entwickelt. Zu Beginn wurde das Schieberegister auf einem Steckbrett ausgetestet. Danach konnte ein Schema und Layout im KiCad gezeichnet werden, welches dann von JLCP-CB [28] hergestellt wurde. Bauteile wurden entweder von der Schule bezogen oder von DigiKey [29] bestellt. In der ersten PCB (Printed Circuit Board) Version war noch kein Kommunikationsport vorhanden, es hatte noch keinen externen Oszillator und noch diverse Speisungsfehler für den Mikrocontroller. In der zweiten Version wurde dies behoben. Dennoch hat es noch einen kleinen Fehler, der Ausgesetzt werden musste, damit die USB-Kommunikation einwandfrei funktioniert. Die im Anhang 5 beigelegte Version enthält alle Verbesserungen.

##### 3.1.3 Entwicklungsprozess der Software

Die Software wurde ebenfalls iterativ entwickelt [30]–[33]. Zuerst entstand die Auswertung für eine LED-Helligkeitskalibration (Anhang 8), der Abschnitt 3.4.1 geht näher darauf ein. Danach entstand eine Versi-

on, in der noch keine Kalibration durchgeführt wurde und die Position des Gerätes von Hand eingezeichnet werden musste (Anhang 11). Darin befindet sich der erste Ansatz zur Auswertung eines Messvideos. In der aktuellen Version der Auswertungssoftware (Anhang 10) wurden die beiden Teile zusammengeführt. Damit erfolgt die Auswertung nun vollautomatisch, denn das Messgerät wird gefunden, und ebenso eine Helligkeitskalibration durchgeführt.

Für den ersten Teil ist die Mikrokontroller-Software im Anhang 7 erstellt worden. Danach wurde eine separate Version kontinuierlich erweitert in Anhang 6. Dies ist gleichzeitig die aktuelle Version.

### 3.1.4 Entwicklungsprozess des Gehäuses

Das Gehäuse wurde, wie auch die Hardware und Software, iterativ entwickelt. Die ersten paar Versionen hatten zu kleine Toleranzen und es schlichen sich Logikfehler ein. Die aktuelle Version im Anhang 4 passte schlussendlich. Der Prusa MK4 wurde verwendet um dies physikalisch zu erstellen.

## 3.2 Kalibrationsgerät

Ein Kalibrationsgerät wurde entwickelt, welches die Auswerteverfahren in Abschnitt 3.5 ermöglicht.

### 3.2.1 Blockschaltbild

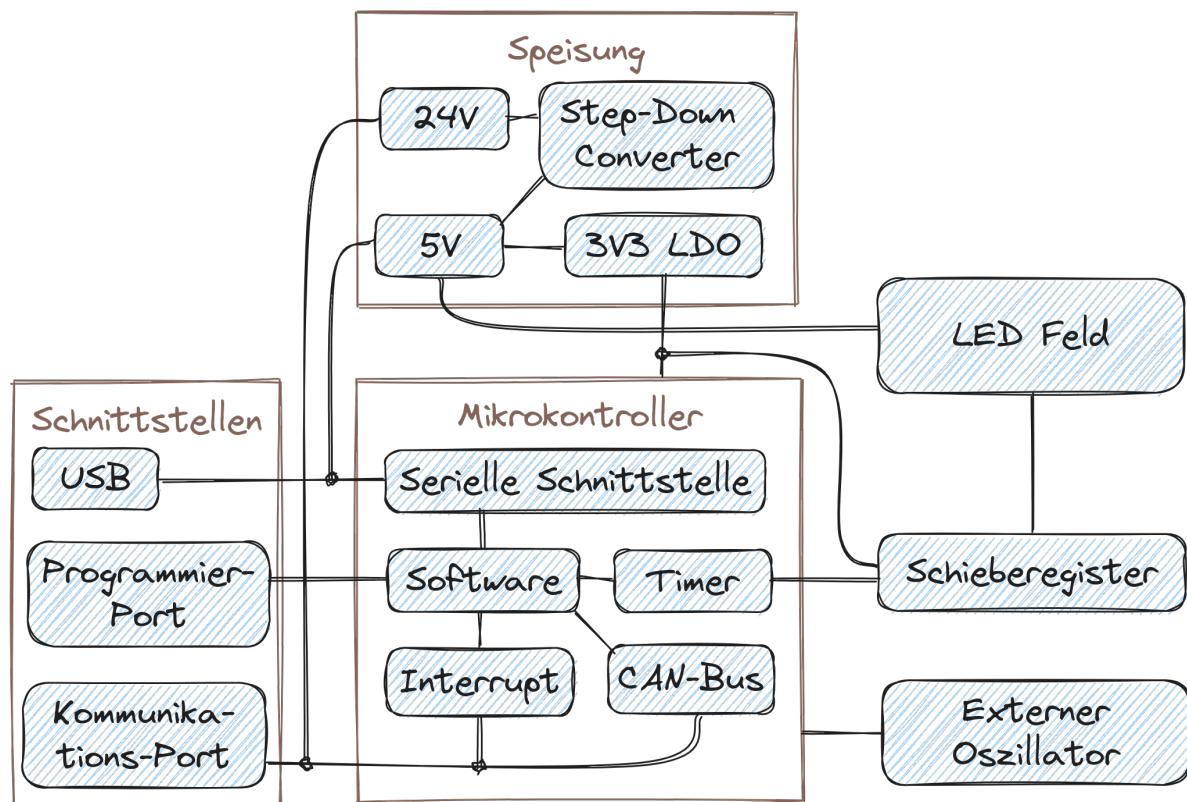


Abbildung 2: Blockschaltbild

Diverse Punkte von Abbildung 2 genauer betrachtet:

- Für den Mikrokontroller ist ein STM32 mit einer Taktfrequenz von 72MHz vorgesehen.
- Das Schieberegister fungiert auch direkt als Stromtreiber für die LEDs. Die maximale Taktfrequenz dessen beträgt 30MHz.
- Kommunikation mit dem PC geschieht über den USB-Port per serieller Schnittstelle.

- Kommunikation mit anderen Kalibrationsgeräten geschieht über den Kommunikations-Port per CAN-Bus (*noch nicht implementiert*). Ein Interrupt soll auf eine konfigurierbare Aktion belegt werden können, zum Beispiel als externer Trigger (*aktuell nur belegbar auf den auto Modus*).
- Es ist möglich das Gerät über USB oder eine 24V-Speisung zu betreiben.
- Um eine möglichst genaue Taktrate zu erzielen wird ein externer Oszillator verwendet.

### 3.2.2 Hard- & Software

Alles bezüglich der Hardware (Schema und Layout) ist im Anhang 5. Das PCB ist in Abbildung 3 gezeigt und beschrieben. Das Gehäuse ist im Anhang 4. Die gesamte Software des Messgeräts ist im Anhang 6.

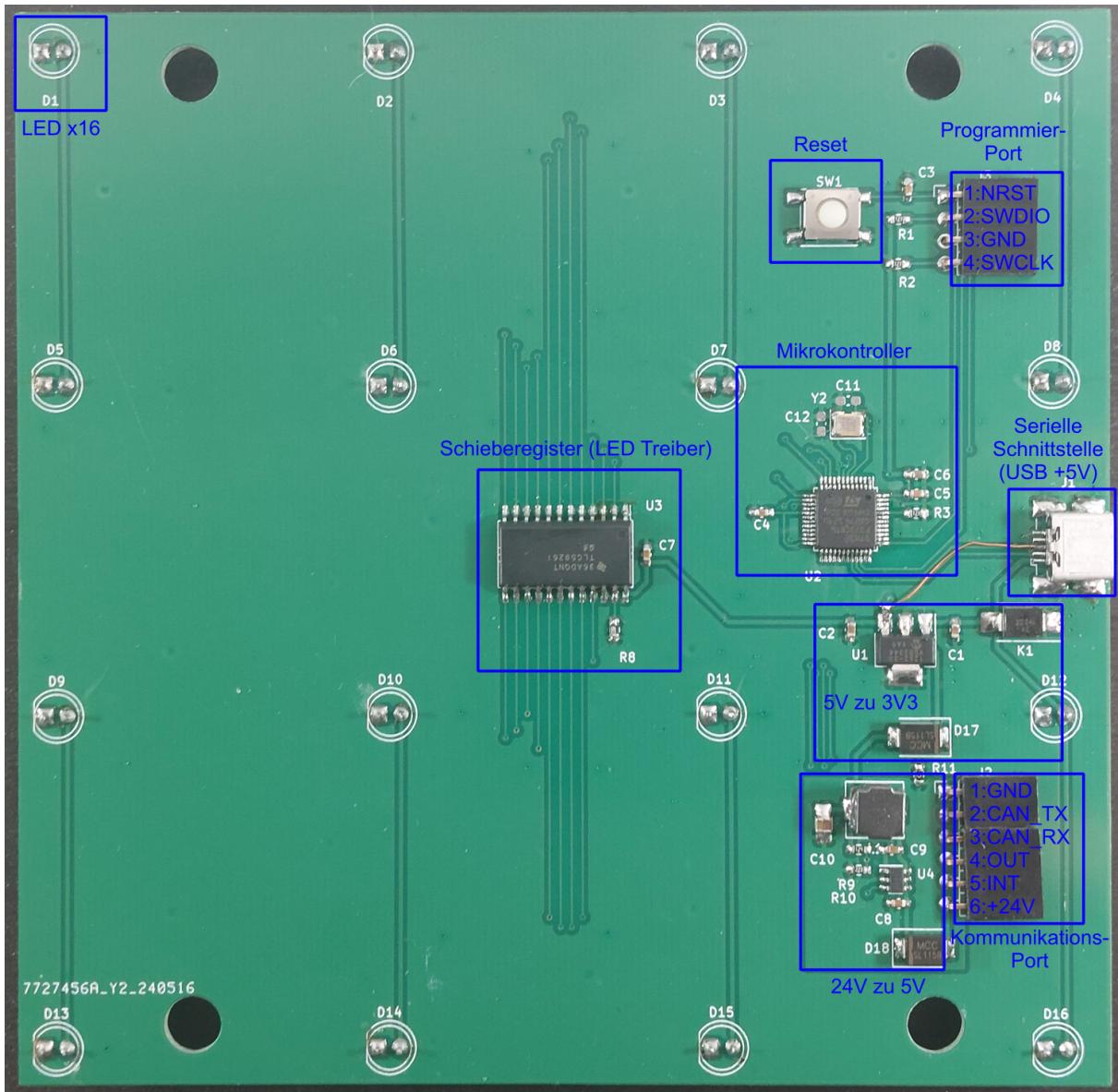


Abbildung 3: PCB

### 3.2.3 Anordnung der Leuchtdioden

Verschiedene größen von quadratischen Anordnungen wurden in Betracht gezogen. Der grösste Einflussende Faktor ist jedoch die Anzahl an Ausgängen vom Schieberegister. Das ausgesuchte Schieberegister besitzt 16 Ausgänge. Somit kann passend ein Feld bestehend aus 4x4, also insgesamt 16 LEDs, aufgebaut werden.

### 3.2.4 Kodierung der Zeit

Die erste Idee war, dass ein Zeitstempel, ähnlich wie bei QR-Codes (quick-response code) oder Data-Matrizen, angezeigt wird. Dies müsste dann mit der Frequenz abgestimmt werden und es hätte in jedem Bild einen unverwechselbaren Zeitstempel. Kompliziert wird es jedoch, wenn die Kamera durch die Belichtungszeit mehrere Zeitstempel ineinander verwischen lässt.

Der zweite und ausgewählte Ansatz ist das "wandern" lassen einer LED. Es soll also zu jedem Zeitpunkt immer nur eine LED leuchten. Die Geschwindigkeit der "Wanderung" wird abgestimmt auf die Aufnahmefrequenz, so dass einmal das gesamte Feld innerhalb von einer zur nächsten Bildaufnahme besucht wird (siehe Abschnitt 3.6.3). Damit lässt sich in der Video-Analyse anhand von Änderungen in der Position der LEDs die Stabilität ausfindig machen. Durch die Anzahl an leuchtenden LEDs im Messvideo lässt sich die Belichtungszeit ermitteln.

### 3.2.5 Infobefehle

Über die serielle Schnittstelle sind die Infobefehle aus Tabelle 3 zur Verfügung gestellt.

Befehl	Zweck
info	Auflistung aller Infobefehle ( <i>dies ist als Hilfestellung gegeben</i> )
info version	Ausgabe der aktuellen Version
info arr	Ausgabe der arr Konfiguration
info psc	Ausgabe der psc Konfiguration
info fps	Ausgabe der fps Konfiguration
info out	Ausgabe der out Konfiguration
info int	Ausgabe der int Konfiguration
info all	Informationen von allem anzeigen

Tabelle 3: Liste der Infobefehle

### 3.2.6 Konfigurationsmöglichkeiten

Über die serielle Schnittstelle sind die Konfigurationsbefehle aus Tabelle 4 zur Verfügung gestellt.

Befehl	Zweck
config	Auflistung der Konfigurationsbefehle ( <i>dies ist als Hilfestellung gegeben</i> )
config fps=ZAHL	Setzen der Schaltfrequenz einer einzelnen LED im rider Modus
config arr=ZAHL	Manuelles einstellen vom Auto-Reload des Timers
config psc=ZAHL	Manuelles einstellen vom Prescaler des Timers
config int=on/off	Ein- oder ausschalten des GPIO Interrupts zum Start vom auto Modus
config out=on/off	Ein- oder ausschalten des GPIO Ausgangs beim Start vom auto Modus
config reset	Reset auslösen

Tabelle 4: Ansteuerung der Konfiguration

- Die fps Konfiguration versucht möglichst nahe an die gewünschte FPS (Bilder pro Sekunde) heranzukommen und setzt den Auto-Reload und Prescaler automatisch.
- Wird arr oder psc verändert, aktualisiert dies die fps gemäss (1).

### 3.2.7 Modi

Das Gerät kann in verschiedene Modi gesetzt werden, die alle einen bestimmten Zweck erfüllen, welche in Tabelle 5 aufgelistet sind.

Befehl	Zweck
mode	Auflistung der Modi ( <i>dies ist als Hilfestellung gegeben</i> )
mode none	Alle LEDs aus
mode pwm=ZAHL	Alle LEDs über PWM ansteuern
mode display=BINÄRZAHL	Eine Zahl anzeigen
mode rider	Eine LED wandern lassen
mode auto	Automatisches abspielen der Modi (siehe Abschnitt 3.5.1)

Tabelle 5: Ansteuerung der Modi

- Im `pwm` Modus kann eine Zahl zwischen 0 bis und mit 16 angegeben werden. Somit gibt dies 17 Stufen (siehe Abschnitt 3.6.2.2).
- Im `display` Modus muss mindestens eine 16-stellige Binärzahl angegeben werden, bei der jedes Bit einer LED entspricht. Alles nach 16 Bits wird ignoriert. Beispiel: `mode display=1010010111110000`.

### 3.3 Rechenmethoden

Aus den aktiven Leuchtdioden können zwei grundlegende Informationen extrahiert werden:

1. Bildwiederholrate Zentrum der aktiven Leuchtdioden, siehe Abschnitt 3.5.4
2. Belichtungszeit Breite der aktiven Leuchtdioden, siehe Abschnitt 3.5.5

Da die Auswertung über mehrere Bilder geschieht, kann über die Stabilität der Bildwiederholrate und der Belichtungszeit beurteilt werden (*Jitter*).

### 3.4 Messmethoden

#### 3.4.1 Machbarkeitsanalyse der Kalibration

Zu Beginn war unbekannt, ob es Sinn machen würde, die Helligkeiten der LEDs zu kalibrieren, und ob dies überhaupt funktioniert. Deshalb wurden auf einer LED eines STM32 Nucleo Boards verschiedene Helligkeitsstufen mittels PWM eingestellt. Eine Kamera nahm Bilder davon auf und die Helligkeiten wurden ausgewertet.

In Abbildung 4 sind die Verläufe, nebst einer Ideallinie, eingezeichnet. Es ist ersichtlich, dass eine Helligkeits-, bzw. PWM-Kalibration potentiellen nichtlinearitäten entgegenwirken könnte. Es kommt ebenso darauf an, wie gross die auszuwertende Fläche in Pixel ist, welche in Abbildung 5 dargestellt ist. Die weissen Pixel in der Mitte sind Teil der leuchtende LED.

Der Source-Code für den Mikrokontroller ist im Anhang 7 und die Auswertesoftware im Anhang 8 zu finden.

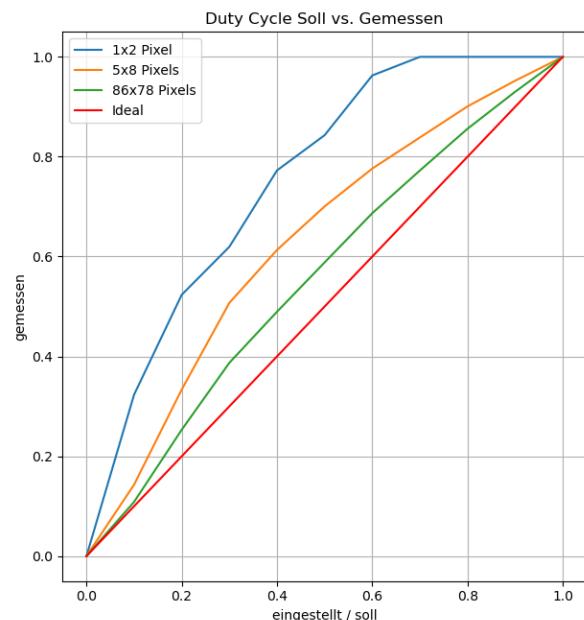


Abbildung 4: PWM Machbarkeitsanalyse

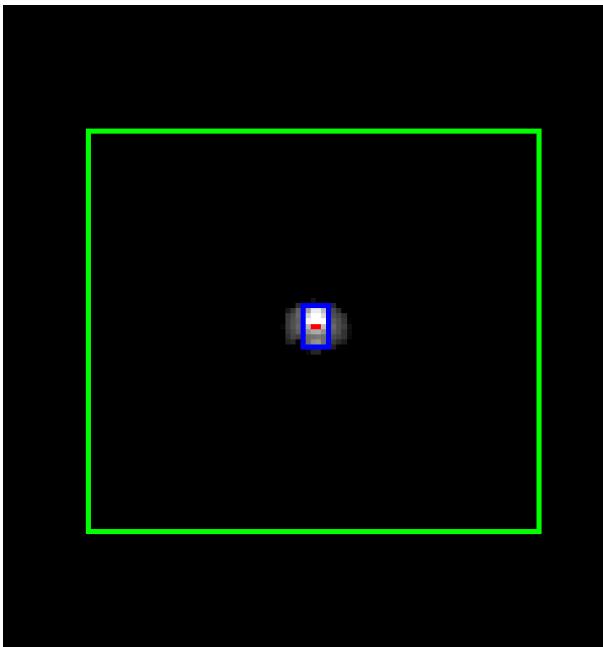


Abbildung 5: PWM Machbarkeitsanalyse (Ausgewertete Flächen)

### **3.4.2 Überprüfung von Frequenzen**

Um die Frequenzen des Messgerätes zu überprüfen wurde ein Oszilloskop verwendet. Eine Sonde misst direkt an einer LED wie in Abbildung 6 gezeigt, dessen Ground wie in Abbildung 7 verbunden ist.

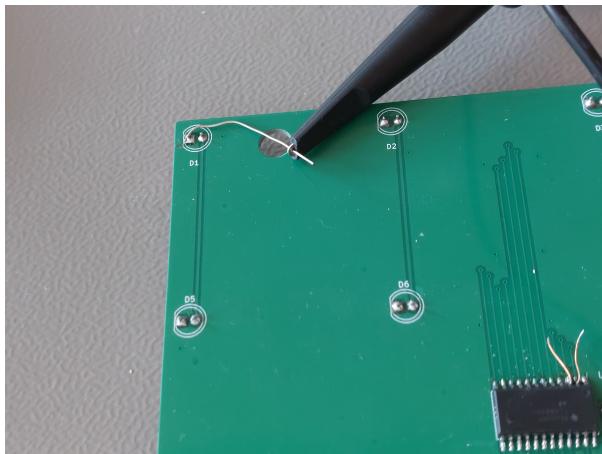


Abbildung 6: Messpunkt LED

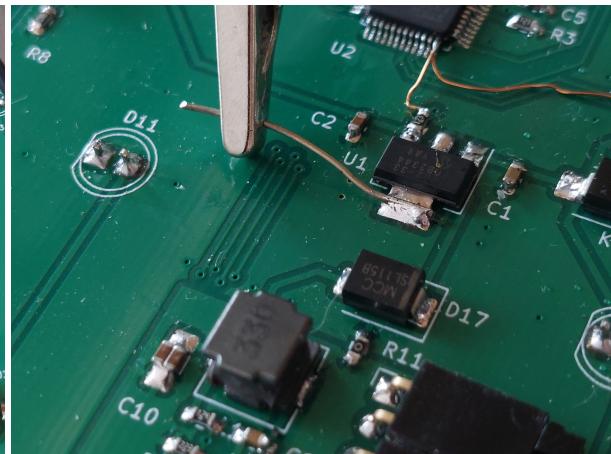


Abbildung 7: Messpunkt Ground

### 3.4.3 Aufnahme Video

Diverse Messvideos in verschiedensten Anordnungen können aufgenommen werden. Bei allen folgenden Messaufbauten ist die automatische Positionserkennung und Auswertung möglich.

Zum Beginn werden alle LEDs ausgeschaltet über den none Modus und die Zielfrequenz des Messgeräts über die Konfiguration fps eingestellt. Danach wird die Aufnahme und der auto Modus gestartet. Die Kamera und das Messgerät werden während der gesamten Aufnahmzeit nicht verschoben. Außerdem sollte sich die Umgebungshelligkeit so wenig wie möglich ändern, damit die Zuverlässigkeit der Ergebnisse erhöht werden kann.

Zur optimalen Objektiveinstellung und Überprüfung, ob das gesamte LED-Feld im Video sichtbar ist, empfiehlt es sich alle LEDs vor der obig beschriebenen Aufnahme leuchten lassen um dies zu kontrollieren.

Der Code in Anhang 9 zeigt, wie dieser gesamte Prozess von Konfiguration zur Aufnahme automatisiert werden kann.

- Die Möglichkeit in Abbildung 8 zeigt die kontrollierte Fixierung der Kamera welche normal auf das Gerät schaut.
- In Abbildung 9 ist dieselbe Kamera ungefähr fünf Meter entfernt beschwert und positioniert worden. Das Gerät ist noch zusätzlich rotiert aufgestellt.
- Bei Abbildung 10 ist die Kamera ungefähr zwei Meter entfernt positioniert und das Gerät ist noch zusätzlich rotiert und schräg aufgestellt.
- Bei Abbildung 11 ist die Kamera so nah wie möglich positioniert, sodass das Gerät gerade noch sichtbar ist im Video. Das Gerät ist noch zusätzlich rotiert und schräg aufgestellt.



Abbildung 8: Aufbau Aufnahme



Abbildung 9: Aufbau (Weit)



Abbildung 10: Aufbau (Weit + Schräg)



Abbildung 11: Aufbau (Nah + Schräg)

### 3.5 Auswerteverfahren

#### 3.5.1 Reihenfolge der Modi

Die Reihenfolge der Modi im auto Modus ist in Tabelle 6 beschrieben.

#	Schritt	Modus	Beschreibung	Zweck
1.	ALL_OFF	none	Alle LEDs aus	Initiale Detektion
2.	ALL_ON	display	Alle LEDs ein	Initiale Detektion
3.	PATTERN	display	Orientierungsmuster	Initiale Detektion
4.	SYS_FREQ	display	Systemfrequenz in MHz	Übermittlung Einstellungen
5.	PSC	display	Prescaler vom Timer	Übermittlung Einstellungen
6.	ARR	display	Auto-Reload-Register vom Timer	Übermittlung Einstellungen
7.	PWM_COUNT	display	Anzahl von PWM Stufen	Kalibration für Rider
8.	PWM_CYCLE	pwm	PWM Stufen	Kalibration für Rider
9.	PWM_END	none	Alle LEDs aus	Kalibration für Rider
10.	RIDER	rider	Eine LED wandert kontinuierlich	Tatsächliche Auswertung

Tabelle 6: Reihenfolge der Modi

### 3.5.2 Detektion & Ausrichtung des Orientierungsmuster

Hierfür werden die ersten drei Schritte ALL\_OFF, ALL\_ON & PATTERN verwendet.

#### 3.5.2.1 Detektion aller LEDs & Finden des Orientierungsmuster

In diesem ersten Schritt werden ein aktuelles Bild mit dem vorangehenden verglichen. Auschlaggebender Indikator für eine Musteränderung ist ein globaler Unterschied der Pixelhelligkeiten. Die zwei Schritte, welche darauf zurückgreifen, sind in Tabelle 7 aufgelistet.

Vorherig	Aktuell	Indikator	Gewonnene Information
ALL_OFF	ALL_ON	Alle LEDs beginnen zu leuchten	Index $i_{\text{ALL\_ON}\perp}$ des Schrittes im Video
ALL_ON	PATTERN	Mehrere LEDs schalten aus	Index $i_{\text{PATTERN}\perp}$ des Schrittes im Video

Tabelle 7: Initiale Detektion

Aus den nun bekannten Indizes können robust  $i_{\text{ALL\_OFF}}$  (2),  $i_{\text{ALL\_ON}}$  (3) und  $i_{\text{PATTERN}}$  (4) berechnet werden, in denen jeweils keine, alle LEDs, oder das Orientierungsmuster zu sehen sind im Bild (5).

$$i_{\text{ALL\_OFF}} = \left\lfloor \frac{3 \cdot i_{\text{ALL\_ON}\perp} - i_{\text{PATTERN}\perp}}{2} \right\rfloor \quad (2)$$

$$i_{\text{ALL\_ON}} = \left\lfloor \frac{i_{\text{PATTERN}\perp} + i_{\text{ALL\_ON}\perp}}{2} \right\rfloor \quad (3)$$

$$i_{\text{PATTERN}} = \left\lfloor \frac{3 \cdot i_{\text{PATTERN}\perp} - i_{\text{ALL\_ON}\perp}}{2} \right\rfloor \quad (4)$$

$$M_i = \text{Bild Nummer } "i" \text{ im Video} \quad (5)$$

#### 3.5.2.2 Gültige Positionen Filtern

Hierfür werden die beiden Bilder “Dunkel” und “Hell”, bei jeweils den indizes (2) und (3), verwendet. Nacheinander kommen folgende Schritte zur Anwendung, um möglichst robust die gültige Position  $p$  des Gerätes zu finden:

1. Berechnung von “Unterschied” = “Dunkel” – “Hell”
2. Anwenden von OTSU-Filter [34] auf “Unterschied”
3. Konturen finden & deren Zentren berechnen

4. Alle Zentren verbinden & deren Längen berechnen
5. Winkel der Verbindungen berechnen und ähnlich orientierte Verbindungen gruppieren
6. Gruppierung mit den meisten Winkeln ergibt die Hauptrichtung
7. Gruppierung mit genügend grosser Winkeldifferenz zur Hauptrichtung ergibt die zweite Richtung
8. Verbindungen ziehen entlang der ersten und zweiten Richtung
9. Die kürzesten acht Verbindungen verwerfen in beiden Richtungen (um zum Beispiel Spiegelungen an einem Tisch herauszufiltern)
10. Lange Verbindungen verwerfen in beiden Richtungen
11. Verbleibenden Verbindungen welche nicht auf jeweils mindestens zwei Konturen fallen verwerfen
12. Finale Verbindungen gehen zurück auf die Zentren der Konturen, dies sind die Positionen
13. Finale Positionen in X- und Y-Richtung sortieren

### 3.5.2.3 Interpretation des Orientierungsmuster

Das zu untersuchende Bild, welches das Orientierungsmuster anzeigt, ist durch (4) gegeben. Außerdem wird, ähnlich zum vorherigen Schritt ein "Dunkel"-Bild (2) abgezogen vom zu untersuchenden Bild.

$$\widehat{M}_{\text{PATTERN}} = \frac{M_{\text{PATTERN}} - M_{\text{ALL\_OFF}}}{M_{\text{ALL\_ON}} - M_{\text{ALL\_OFF}}} \quad (6)$$

Die normierten Zustände der LEDs werden in (6) bei den vorherig ermittelten Positionen bestimmt. In einem vom Benutzer beeinflussten Radius wird für jede Position der Mittelwert gebildet. Durch binäres Thresholding um 0.5 werden die Bit-Zustände berechnet.

Das Gerät zeigt die Hexadezimalzahl 0x842F als Orientierungsmuster an. Auf einem 4x4 Bitfeld resultiert dies in einem "Pfeil", welcher auf das LSB (Least significant Bit) zeigt, wie in Tabelle 8 zu sehen.

Das gezeigte Muster kann gespiegelt oder eine beliebige Richtung gedreht sein. Um das Muster korrekt zu orientieren, wird im ersten Schritt die Reihe gesucht, in der alle vier Bits aktiv sind (rot dargestellt).

Danach kann das Muster um 90° gedreht nach den Reihenfolgen 0b1000 (blau dargestellt), 0b1100, 0b1010 und 0b1001 überprüft werden.

Muster	Gespiegelt
1 0 0 0	0 0 0 1
0 1 0 0	0 0 1 0
0 0 1 0	0 1 0 0
<b>1 1 1 1</b>	<b>1 1 1 1</b>

Tabelle 8: Orientierungsmuster

### 3.5.3 Übermitteln von Einstellungen & Kalibration der Leuchtdioden-Intensität

Es werden die Schritte SYS\_FREQ, PSC, ARR, PWM\_COUNT, PWM\_CYCLE & PWM\_END analysiert.

#### 3.5.3.1 Erkennen aller Binärwerte

Bild für Bild wird die Binärzahl ausgewertet des aktuellen Bildes  $M_{\text{aktuell}}$  bei index  $i_{\text{aktuell}}$  in (7).

$$\widehat{M}_{\text{aktuell}} = \frac{M_{\text{aktuell}} - M_{\text{ALL\_OFF}}}{M_{\text{ALL\_ON}} - M_{\text{ALL\_OFF}}} \quad (7)$$

Die zu erwartenden Werte sind alle unbekannt aber grösser als 0. Einzig der Wert nach dem Anzeigen von allen digitalen Informationen, bei der PWM-Stufe 0% ist bekannt, nämlich 0. Ausschlaggebender Index ist folglich  $i_{\text{PWM\_0\%}}$  wenn eine 0 angezeigt wird. Dazu wird der Unterschied in (8) gebraucht, um die restlichen Indizes der Schritte 4. (9) bis 7. (12) zu ermitteln.

$$\Delta i = \frac{i_{\text{PWM\_0\%}\perp} - i_{\text{PATTERN}\perp}}{5} \quad (8)$$

$$i_{\text{SYS\_FREQ}} = i_{\text{PATTERN}\perp} + \frac{3 \cdot \Delta i}{2} \quad (9)$$

$$i_{\text{ARR}} = i_{\text{PATTERN}\perp} + \frac{5 \cdot \Delta i}{2} \quad (10)$$

$$i_{\text{PSC}} = i_{\text{PATTERN}\perp} + \frac{7 \cdot \Delta i}{2} \quad (11)$$

$$i_{\text{PWM\_COUNT}} = i_{\text{PATTERN}\perp} + \frac{9 \cdot \Delta i}{2} \quad (12)$$

Um die Robustheit sicherzustellen, wird bei jedem Index jeweils um  $\pm \frac{\Delta i}{4}$  erweitert in (13) und die angezeigten Binärzahlen der entsprechenden Bilder ausgewertet.

$$\bar{M}_i(\Delta i) = \left\lfloor \frac{2}{\Delta i} \right\rfloor \sum_{k=\left\lfloor i - \frac{\Delta i}{4} \right\rfloor}^{\left\lfloor i + \frac{\Delta i}{4} \right\rfloor} \widehat{M}_k \quad (13)$$

### 3.5.3.2 Intensitätskalibration

Die Intensitätskalibration (siehe Abschnitt 3.4.1) geschieht über die PWM Ansteuerung der LEDs.

Ähnlich wie im Abschnitt 3.5.3.1 wird zuerst das Ende von PWM-END  $i_{\text{PWM\_END}\perp}$  gesucht, wo alle LEDs ausgeschaltet sind. Die Anzahl an PWM-Stufen in linearen Abständen ist vorgegeben durch (12). Durch die Anzahl an Stufen und den Start- & Stopp-Indizes  $i_{\text{PWM\_0\%}\perp}$  und  $i_{\text{PWM\_END}\perp}$  werden die Helligkeiten bestimmt und für zukünftige Messungen linear interpoliert in (14).

$$\Delta i_{\text{PWM}} = \frac{i_{\text{PWM\_END}\perp} - i_{\text{PWM\_0\%}\perp}}{n_{\text{PWM\_COUNT}} + 1} \quad (14)$$

Die Bilder bei PWM-Stufen  $k \in \{0 \dots n_{\text{PWM\_COUNT}}\}$  in (15) werden mithilfe von (13) über (14) gemittelt. Bei den Positionen  $p$  werden in (16) die Helligkeiten ausgewertet.

$$i_{\text{PWM}}(k) = k \cdot \Delta i_{\text{PWM}} + i_{\text{PWM\_0\%}\perp} + \frac{\Delta i_{\text{PWM}}}{2} \quad (15)$$

$$\bar{M}_{\text{PWM}}(k)(\Delta i_{\text{PWM}}) \quad (16)$$

Angenommen das Gerät ist konfiguriert auf 16 PWM-Stufen, so sind die Helligkeiten von 0% bis 100% in 16 Schritten linear abgebildet. In Abbildung 12 und Abbildung 13 sind die Punkte die gemessenen Helligkeiten und die gestrichelten Linien die rückwärts linear interpolierten Kurven.

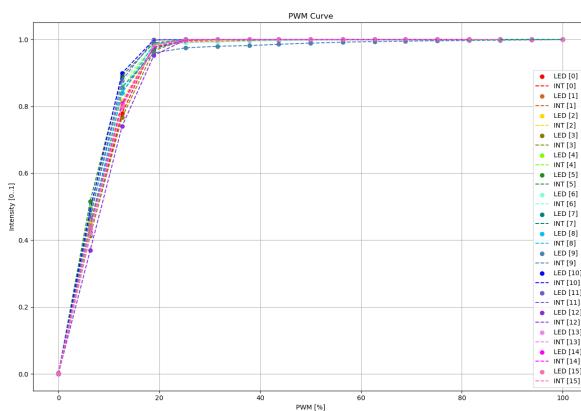


Abbildung 12: PWM Kurven bei offener Blende

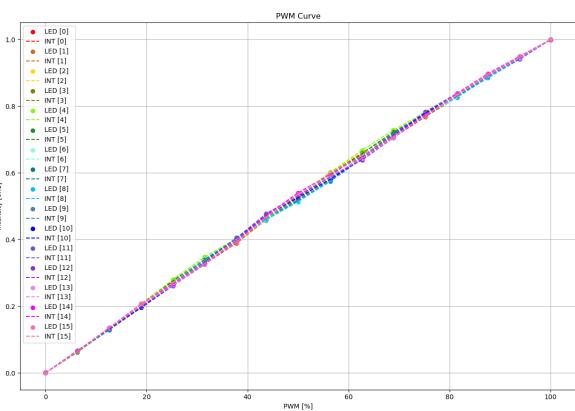


Abbildung 13: PWM Kurven bei mittlerer Blende

### 3.5.4 Zentrum der laufenden Leuchtdiode

Das Zentrum der aktiven Leuchtdioden wird als Kodierung der Stabilität im Video verwendet. Für den Fall, wenn die aktiven LEDs beide Ränder überlappen, muss dies berücksichtigt werden. Der gewählte Ansatz (17) ordnet die LEDs in der Berechnung im Kreis an und berechnet aus den Intensitäten  $I$  den Winkel  $\varphi$ . So ist auch im Fall, wenn eine Wanderung vom einen Rand zum anderen geschieht, klar in welche Richtung sich diese bewegt.

$$\varphi = \arctan \left( \sum_{k=0}^{n_{\text{LED}}-1} \frac{\sin \left( \frac{2\pi k I_{k+1}}{n_{\text{LED}}} \right)}{\cos \left( \frac{2\pi k I_{k+1}}{n_{\text{LED}}} \right)} \right) \quad (17)$$

Über den Prescaler und Auto-Reload ist  $f_R$  durch (1) gegeben. Durch diverse Toleranzen der elektronischen Bauteile ist die eingestellte Geschwindigkeit  $f_R$  der LEDs ungleich der reellen  $f_G$ , siehe Abschnitt 3.6.2.1. Es ist wichtig, dass eine korrekte Einstellung der Frequenz vom Messgerät vorgenommen wird, siehe Abschnitt 3.6.3.

Der Unterschied der Eingestellten tatsächlichen Frequenz  $f_G$  zur Kamerafrequenz ist durch (18) gegeben. Daraus und mit (17) lässt sich ein Unterschied in Winkeländerungen  $\Delta\dot{\varphi}$  (19) berechnen. Dieser Unterschied korreliert mit der Abweichung von der eingestellten zur ermittelten Kamerafrequenz  $f$  in (20).

$$\Delta f = f_G - f_K \quad (18)$$

$$\Delta\dot{\varphi} = 2\pi \frac{\Delta f}{f_K} - \dot{\varphi} \quad (19)$$

$$f = \frac{2\pi + \frac{\Delta\dot{\varphi}}{\Delta f}}{2\pi} \cdot f_K \quad (20)$$

### 3.5.5 Breite der laufenden Leuchtdiode

Aus der Anzahl an aktiven Leuchtdioden über die Intensitäten  $I$  und deren maximaler Intensität  $I_{\max}$  wird die Belichtungszeit  $b$  im Video ermittelt (21).

$$b = \frac{1}{f \cdot n_{\text{LED}}} \sum_{k=1}^{n_{\text{LED}}} \frac{I_k}{I_{\max}} \quad (21)$$

## 3.6 Ergebnisse

### 3.6.1 Frequenzlimitationen

Die maximal Einstellbare Geschwindigkeit einer LED im `rider` Modus beträgt 4.5kHz.

Die Taktgeschwindigkeit im `pwm` Modus jeder LED beträgt auch 4.5kHz.

### 3.6.2 Frequenzzuverlässigkeit

#### 3.6.2.1 Rider Modus

Verschiedene Frequenzen wurden an einer LED im `rider` Modus mit einem Oszilloskop gemessen. In Tabelle 9 sind die genauen Messwerte aufgelistet und im Anhang 12 beigelegt. Diese sind in Abbildung 14 relativ dargestellt und es zeigt sich, dass die tatsächliche Frequenz  $f_G$  (gelb) bis zu 0.2% von der Zielfrequenz  $f_R$  abweicht. Die Standardabweichung  $\sigma$  (rot) steigt mit der Frequenz.

$f_R$ [Hz]	$f_G$ [Hz]	$\sigma$ [ $\mu$ Hz]
30.000	30.005	62.566
40.000	40.007	366.58
56.250	56.259	129.31
59.700	59.711	368.78
60.000	60.010	483.37
60.299	60.309	425.85
64.286	60.309	498.19
75.000	75.013	483.33
120.00	120.02	1'120.8
284.99	285.04	3'247.6
752.01	752.16	20'029
4500.0	4499.6	370'780

Tabelle 9: Überprüfte Frequenzen

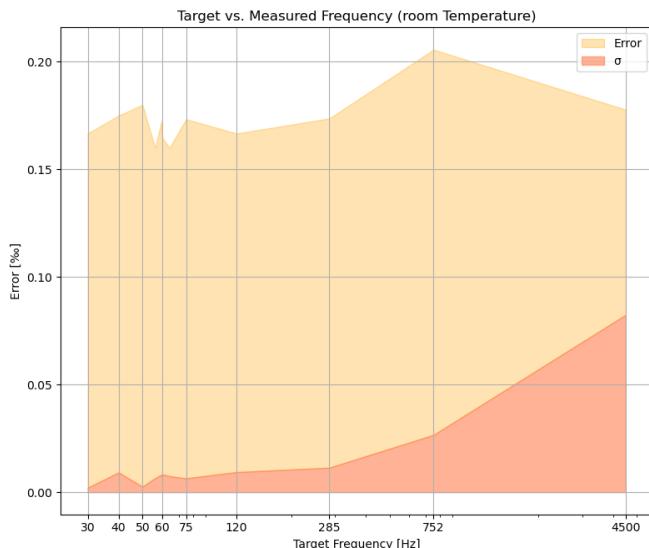


Abbildung 14: Überprüfte Frequenzen

### 3.6.2.2 PWM Modus

$z$	Eingestellt [%]	Gemessen [%]	Fehler [%]
0	0	0	0.00
1	6.25	6.31	0.90
2	12.50	12.61	0.90
3	18.75	18.92	0.90
4	25.00	25.23	0.90
5	31.25	31.53	0.90
6	37.50	37.84	0.90
7	43.75	43.69	-0.13
8	50.00	50.00	0.00
9	56.25	56.40	0.26
10	62.50	62.70	0.32
11	68.75	68.83	0.11
12	75.00	75.14	0.18
13	81.25	81.44	0.24
14	87.50	87.57	0.08
15	93.75	93.87	0.13
16	100.0	100.0	0.00

Tabelle 10: Überprüfte Duty Cycles

Das Messgerät besitzt  $n_{\text{PWM\_COUNT}} = 16$  PWM-Stufen welche grösser sind als 0%. Der konkrete Duty Cycle  $D$  ist gegeben über  $z \in \{0 \dots n_{\text{PWM\_COUNT}}\}$  durch (22).

Diese Stufen wurden einzeln an einer LED gemessen worden mit einem Oszilloskop (siehe Abschnitt 3.4.2) und in Tabelle 10 aufgelistet. Alle Messresultate sind im Anhang 12 beigelegt. Die Abweichungen werden in der Software (Anhang 10) berücksichtigt.

$$D(z) = 100\% \cdot \frac{z}{n_{\text{PWM\_COUNT}}} \quad (22)$$

### 3.6.2.3 Display Modus

Der display Modus ist nicht zeitkritisch und es wurden deshalb keine Frequenzen überprüft. Die Aktualisierungs-, bzw. Verzögerungszeit ist unbekannt, denn diese ist im Moment irrelevant.

### 3.6.3 Gut eingestellt

Die Frequenz des Messgerätes  $f_G$  ist gut eingestellt, wenn sie ein wenig von der Kamerafrequenz  $f_K$  abweicht (zum Beispiel ungefähr 0.5%). Aus der Differenz der Frequenzen kann die im besten Fall maximal detektierbare Anzahl an ausgelassenen Bildern am Stück  $s_{\text{maxskip}}$  berechnet werden gemäss (23).

Resultate verschiedener Konfigurationen sind in Tabelle 11 aufgeführt. Im Anhang 13 sind detaillierte Grafiken über die Zeit zu allen Resultaten zu finden. Die Hersteller [25], [26] geben keine Toleranzen der Kameras an, somit ist kein Vergleich möglich, ob die gemessenen Frequenzen  $\bar{f}$  und Belichtungszeiten  $\bar{b}$  innerhalb einer Toleranz liegen oder nicht. Dennoch sind die Abweichungen kleiner als 0.3%, mit einem Ausreißer bei g).

Kamera	$f_K$ [Hz]	$f_G$ [Hz]	$\bar{f}$ [Hz]	$b_K$ [\mu s]	$\bar{b}$ [\mu s]	Bilder in Anhang 13
IDS	30.00	29.850	30.054	4999	5095	a)
IDS	59.99	58.999	59.959	4999	5123	b)
IDS	59.99	59.701	59.969	4999	5004	c)
IDS	59.99	60.299	59.997	4999	5063	d)
IDS	59.99	61.000	59.978	4999	5040	e)
AOS	60.00	58.999	59.992	5000	5021	f)
AOS	285.0	284.99	281.70	2055	1997	g)
AOS	752.0	752.01	749.86	600	512	h)

Tabelle 11: Auswertung bei gut eingestellten Frequenzen

$$s_{\text{maxskip}} = \frac{f_G}{2 \cdot |f_G - f_K|} \quad (23)$$

### 3.6.4 Schlecht eingestellt

Ist die Kamerafrequenz  $f_K$  gleich oder sehr ähnlich der des Messgerätes  $f_G$ , könnte ein Auslassen eines Bildes im Video eventuell nicht erkannt werden, falls die Aufnahmefrequenz vor und nach dem Auslassen konstant bleibt.

Gemäß (23) können bei bereits 25% abweichen der Frequenz nur noch zwei hintereinander ausgelassene Bilder verlässlich detektiert werden. Dennoch wurden mehrere Videos bei schlechten Frequenzen aufgenommen, um Effekte zu beobachten. Die Resultate sind in Tabelle 11 aufgeführt. In den detaillierteren Grafiken über die Zeit im Anhang 13 ist deutlich, dass bei manchen Messungen Bilder gewonnen wurden, wo keine sind.

Kamera	$f_K$ [Hz]	$f_G$ [Hz]	$\bar{f}$ [Hz]	$b_K$ [\mu s]	$\bar{t}$ [\mu s]	Bilder in Anhang 13
IDS	59.99	40.000	60.684	4999	5011	i)
IDS	59.99	50.000	59.990	4999	5029	j)
IDS	59.99	56.250	59.986	4999	5023	k)
IDS	59.99	60.000	60.176	4999	5025	l)
IDS	59.99	64.286	59.989	4999	4934	m)
IDS	59.99	75.000	59.992	4999	4981	n)
IDS	59.99	120.00	60.989	4999	4894	o)

Tabelle 11: Auswertung bei schlecht eingestellten Frequenzen

### 3.6.5 Einfluss der Bildqualität

Das Bild kann ziemlich verschwommen sein, wie in Abbildung 15 gezeigt, und das Resultat weicht nicht stärker ab als bei idealen Konditionen. Wenn die Kamera einen Rolling Shutter verwendet, zeigt die Auswertung grosse Variation. Zwei solche Messungen wurden durchgeführt und die Resultate sind in Tabelle 13 aufgelistet.

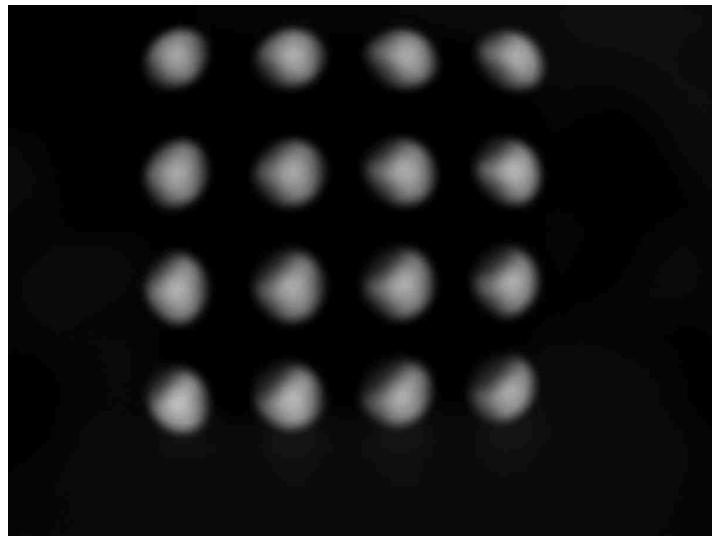


Abbildung 15: Verschwommene LEDs

Kamera	$f_K$ [Hz]	$f_G$ [Hz]	$\bar{f}$ [Hz]	$b_K$ [ $\mu$ s]	$\bar{t}$ [ $\mu$ s]	Bilder in Anhang 13
IDS (verschwommen)	59.99	59.701	59.974	4999	5018	p)
A52 (Rolling Shutter)	60.00	59.701	59.792	ca. 10000	9921	q)

Tabelle 13: Auswertung bei schlechter Bildqualität

### 3.6.6 Einfluss der Kalibration

Aus den Bildern in Anhang 13 erschliesst sich, dass die Kalibration die Messungen zum grössten Teil ein wenig verschärf. Die FPS verschiebt sich im besten Fall sehr wenig (Abbildung 16), die Belichtungszeit wird schmäler (Abbildung 17) und die Belichtungszeit zu Winkel werden ausgeglichen (Abbildung 18). Als Beispiel sind die Bilder aus Anhang 13 d) gezeigt. Die blauen Daten sind korrigiert und die orangenen sind die Rohdaten welche die Kalibration ignorieren.

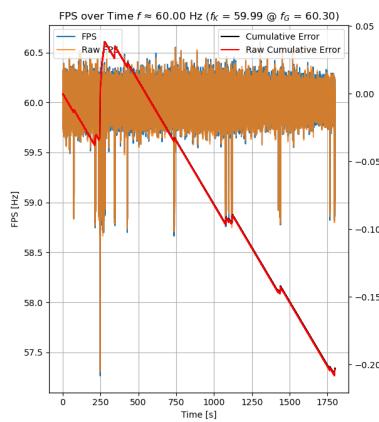


Abbildung 16: FPS über Zeit

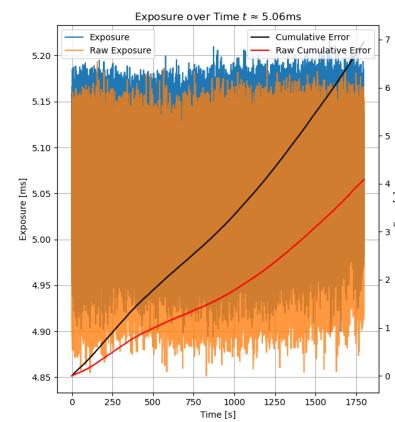


Abbildung 17: Belichtung über Zeit

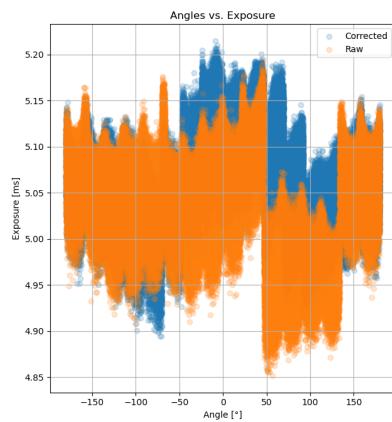


Abbildung 18: Belichtung zu Winkel

## 4 Schlussteil

### 4.1 Zusammenfassung der Ergebnisse

- Die Kodierung der Zeit geschieht über Änderungen in der Position der LEDs (Abschnitt 3.5.4).
- Die Kodierung der Belichtungszeit ist ablesbar aus der Anzahl an aktiven LEDs (Abschnitt 3.5.5).

### 4.2 Interpretation der Resultate

Die in Anhang 13 gezeigten Resultate haben diverse Einflüsse:

- Eine Änderung der Belichtungszeit über die Messdauer spielt vermutlich zusammen mit sich ändernden Lichtverhältnissen, denn dies ist sehr anfällig darüber.
- Die FPS bleiben in der Regel sehr gleichmäßig, bei Ausreisern gingen entweder Bilder verloren (weniger FPS), die Frequenzen waren nicht gut abgestimmt (weniger oder mehr FPS), die Kamera verlor sehr viele Bilder (mehr FPS) oder die Kamera wiederholte gewisse Bilder (mehr FPS). Ein Fall, welcher nicht detektiert werden kann, ist wenn ein Vielfaches von (18) Bildern verloren gehen.
- Wenn die Frequenzen zwischen Messgerät und Kamera gut eingestellt sind, weicht die ermittelte FPS in der Regel weniger als 0.3% von der vorgegebenen ab. Die grösste Abweichung ist sichtbar in den Resultaten g).
- Die Kalibration verbessert in der Regel die Belichtungszeit zu den Winkeln, was die Effektivität dessen darstellt. Je regelmässiger dies ist, desto Wahrheitsgetreuer sind alle Resultate bei einer stabilen Kamera.

### 4.3 Kritische Beurteilung

Verschiedene Punkte, welche mit dem aktuellen Messgerät und der aktuellen Auswertungssoftware nicht funktionieren, sind folgende:

- Bei Videos mit Rolling Shutter liefert die Auswertesoftware, wenn überhaupt das Orientierungsmuster erkannt wird, falsche Resultate.
- Bei 100% Belichtungszeit von Bildern kann keine Position im rider Modus entdeckt werden. Dies kann zum Beispiel bei einer Hochgeschwindigkeitskamera der Fall sein.
- Es ist wichtig, dass die Frequenz des Messgerätes korrekt eingestellt ist (Abschnitt 3.6.3).
- Kommunikation zwischen Messgeräten ist (noch) nicht implementiert.
- Vorgenommene Konfigurationen gehen beim Reset oder Hochfahren verloren.
- Um die Korrektheit der Auswertung zu erhöhen ist es sehr wichtig, dass sich die Lichtverhältnisse über die Zeit nicht ändern.

### 4.4 Offene Fragen

Eine Auswertung eines Mehrkamerasytems oder Präsenz von mehrfachen Messgeräte je Video konnte aus Zeitgründen nicht näher betrachtet werden.

Ansätze um LEDs im Raum zu tracken [35] konnten aus Zeitgründen nicht angeschaut werden.

### 4.5 Ausblick

Es wurde erwähnt, dass die Auswertung nicht für Kameras mit Rolling Shutter funktioniert. Würde die Orientierung des Messgerätes zur Kamera respektiert und die Auswertungssoftware darauf abgestimmt und erweitert werden, könnte so eventuell trotzdem eine Auswertung vonstatten gehen.

Würden zwei oder mehr Kameras ein Messgerät zur gleichen Zeit filmen, könnte mit einer erweiterten Auswertungssoftware die Stabilität der Kameras oder der Auswertungssoftware näher beurteilt werden.

Bei zwei oder mehr im Bild aufgenommenen Messgeräten, deren Frequenzen leicht versetzt eingestellt sind, könnte  $s_{maxskip}$  (23) erhöht werden.

Um die Auswertungssoftware auf Korrektheit zu Überprüfen, könnten anstelle von aufgenommenen, kontrolliert Computer-Generierte Videos erstellt und ausgewertet werden.

## 5 Quellenverzeichnis

- [1] M. Billeter, G. Röthlin, J. Wezel, D. Iwai und A. Grundhöfer, "A LED-Based IR/RGB End-to-End Latency Measurement Device," in *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, 2016, S. 184–188. DOI: 10.1109/ISMAR-Adjunct.2016.0072.
- [2] Q. Huang, W.-S. Chen, X.-Y. Huang und Y. Zhu, "Data Matrix Code Location Based on Finder Pattern Detection and Bar Code Border Fitting," *Mathematical Problems in Engineering*, Jg. 2012, Apr. 2012. DOI: 10.1155/2012/515296.
- [3] "Rolling vs Global Shutter." [Online], Adresse: <https://www.photometrics.com/learn/advanced-imaging/rolling-vs-global-shutter> (besucht am 09.08.2024).
- [4] KiCad. "KiCad EDA – A Cross Platform and Open Source Electronics Design Automation Suite." [Online], Adresse: <https://www.kicad.org/> (besucht am 08.08.2024).
- [5] "STM32CubeIDE – Integrated Development Environment for STM32." [Online], Adresse: <https://www.st.com/en/development-tools/stm32cubeide.html> (besucht am 08.08.2024).
- [6] "Autodesk Fusion | Free Software for Students and Educators." [Online], Adresse: <https://www.autodesk.com/education/edu-software/fusion> (besucht am 08.08.2024).
- [7] "PrusaSlicer 2.8.0. An open-source, feature-rich, frequently updated tool that contains everything you need to export the perfect print files for your 3D printer." [Online], Adresse: [https://www.prusa3d.com/page/prusaslicer\\_424/](https://www.prusa3d.com/page/prusaslicer_424/) (besucht am 08.08.2024).
- [8] "Home | LibreOffice - Free and private office suite." [Online], Adresse: <https://www.libreoffice.org/> (besucht am 08.08.2024).
- [9] "MobaXterm – Enhanced terminal for Windows with X11 server, tabbed SSH client, network tools and much more." [Online], Adresse: <https://mobaxterm.mobatek.net/> (besucht am 08.08.2024).
- [10] "Excalidraw | Hand-drawn look & feel • Collaborative • Secure." [Online], Adresse: <https://excalidraw.com/> (besucht am 08.08.2024).
- [11] "Welcome to Python.org." [Online], Adresse: <https://www.python.org/> (besucht am 08.08.2024).
- [12] "C++ standard." [Online], Adresse: <https://isocpp.org/std/the-standard> (besucht am 08.08.2024).
- [13] "OpenCV – Open Source Computer Vision." [Online], Adresse: <https://docs.opencv.org/4.x/> (besucht am 08.08.2024).
- [14] "NumPy – The fundamental package for scientific computing with Python." [Online], Adresse: <https://numpy.org/> (besucht am 08.08.2024).
- [15] "Matplotlib: Visualization with Python." [Online], Adresse: <https://matplotlib.org/> (besucht am 08.08.2024).
- [16] "SciPy – Fundamental algorithms for scientific computing in Python." [Online], Adresse: <https://scipy.org/> (besucht am 08.08.2024).
- [17] "Vim – the ubiquitous text editor." [Online], Adresse: <https://www.vim.org/> (besucht am 08.08.2024).
- [18] "neovim – hyperextensible Vim-based text editor." [Online], Adresse: <https://neovim.io/> (besucht am 08.08.2024).

- [19] “Git is a fast and efficient tool for managing small to large projects with branching, staging, and workflows.” [Online], Adresse: <https://git-scm.com/> (besucht am 08.08.2024).
- [20] “GitHub: Let’s build from here · GitHub.” [Online], Adresse: <https://github.com/> (besucht am 08.08.2024).
- [21] “Linux – Wikipedia.” [Online], Adresse: <https://en.wikipedia.org/wiki/Linux> (besucht am 08.08.2024).
- [22] “Microsoft Windows – Wikipedia.” [Online], Adresse: [https://en.wikipedia.org/wiki/Microsoft\\_Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) (besucht am 08.08.2024).
- [23] “VOLTCRAFT DSO-2154 150MHz 4 Ch 1GS/s 80kpts 8 Bit Digital Storage Oscilloscope.” [Online], Adresse: <https://www.rapidonline.com/voltcraft-dso-2154-150mhz-4-ch-1gs-s-80kpts-8-bit-digital-storage-oscilloscope-65-4568> (besucht am 08.08.2024).
- [24] “Original Prusa MK4 3D Printer.” [Online], Adresse: <https://www.prusa3d.com/product/original-prusa-mk4-2/> (besucht am 08.08.2024).
- [25] “UI-3250CP Rev. 2 - ids-imaging.us.” [Online], Adresse: [https://www.ids-imaging.us/store\\_us/ui-3250cp-rev-2.html](https://www.ids-imaging.us/store_us/ui-3250cp-rev-2.html) (besucht am 08.08.2024).
- [26] “AOS Promon U | CHROMOS Industrial.” [Online], Adresse: <https://www.chromos.ch/industrial/en/products/highspeed/hs-cameras/aos-promon-u/> (besucht am 08.08.2024).
- [27] “Galaxy A52 5G | Features & Specs | Samsung US.” [Online], Adresse: <https://www.samsung.com/us/smartphones/galaxy-a52-5g/> (besucht am 08.08.2024).
- [28] JLCPCB. “PCB Prototype & PCB Fabrication Manufacturer - JLCPCB.” [Online], Adresse: <https://jlcpcb.com> (besucht am 08.08.2024).
- [29] “DigiKey - Electronic Components Distributor.” [Online], Adresse: <https://www.digikey.com/> (besucht am 08.08.2024).
- [30] “StackOverflow | Where Developers Learn, Share, & Build Careers.” [Online], Adresse: <https://stackoverflow.com/> (besucht am 08.08.2024).
- [31] “StackExchange | Stack Overflow and the Stack Exchange network help people find the answers they need, when they need them.” [Online], Adresse: <https://stackexchange.com/> (besucht am 08.08.2024).
- [32] “Mathematics Stack Exchange.” [Online], Adresse: <https://math.stackexchange.com/> (besucht am 08.08.2024).
- [33] “GeeksforGeeks | A computer science portal for geeks.” [Online], Adresse: <https://www.geeksforgeeks.org/> (besucht am 08.08.2024).
- [34] “OpenCV Thresholding (cv2.threshold).” [Online], Adresse: <https://pyimagesearch.com/2021/04/28/opencv-thresholding-cv2-threshold/> (besucht am 08.08.2024).
- [35] D. Zheng, K. Cui, B. Bai, G. Chen und J. A. Farrell, “Indoor localization based on LEDs,” in *2011 IEEE International Conference on Control Applications (CCA)*, 2011, S. 573–578. DOI: 10.1109/CCA.2011.6044360.

## 6 Abbildungs- & Tabellenverzeichnis

### Abbildungsverzeichnis

1	Systemabgrenzung	7
2	Blockschaltbild	10
3	PCB	11
4	PWM Machbarkeitsanalyse	14
5	PWM Machbarkeitsanalyse (Ausgewertete Flächen)	14
6	Messpunkt LED	14
7	Messpunkt Ground	14
8	Aufbau Aufnahme	15
9	Aufbau (Weit)	15
10	Aufbau (Weit + Schräg)	15
11	Aufbau (Nah + Schräg)	15
12	PWM Kurven bei offener Blende	18
13	PWM Kurven bei mittlerer Blende	18
14	Überprüfte Frequenzen	20
15	Verschwommene LEDs	22
16	FPS über Zeit	22
17	Belichtung über Zeit	22
18	Belichtung zu Winkel	22

### Tabellenverzeichnis

1	Abkürzungsverzeichnis	5
2	Technische Ausrüstung & Verwendete Tools	9
3	Liste der Infobefehle	12
4	Ansteuerung der Konfiguration	12
5	Ansteuerung der Modi	13
6	Reihenfolge der Modi	16
7	Initiale Detektion	16
8	Orientierungsmuster	17
9	Überprüfte Frequenzen	20
10	Überprüfte Duty Cycles	20
11	Auswertung bei gut eingestellten Frequenzen	21
11	Auswertung bei schlecht eingestellten Frequenzen	21
13	Auswertung bei schlechter Bildqualität	22

## 7 Anhang

1. Kurzbeschreibung: Bachelorarbeit L<sup>A</sup>T<sub>E</sub>X-Source Code  
Ordner: doc/tex  
Datum und Zeit: 09.08.2024 09:40
2. Kurzbeschreibung: Originale Exposé Datei, PDFs (Exposé und Auftragsblatt)  
Ordner: doc/expose  
Datum und Zeit: 09.08.2024 04:25
3. Kurzbeschreibung: Diverse Bilder die verwendet sind in diesem Dokument  
Ordner: doc/images  
Datum und Zeit: 09.08.2024 04:20
4. Kurzbeschreibung: Gehäuse der Hardware  
Ordner: ledrunner/housing-ledrunner  
Datum und Zeit: 08.08.2024 22:34  
Programm: Fusion 360
5. Kurzbeschreibung: Schema und PCB der Hardware  
Ordner: ledrunner/hw-ledrunner  
Datum und Zeit: 09.08.2024 04:51  
Entwicklungsumgebung: KiCad
6. Kurzbeschreibung: STM32 Software der Hardware  
Ordner: ledrunner/sw-ledrunner  
Datum und Zeit: 06.08.2024 12:44  
Entwicklungsumgebung: STM32CubeIDE
7. Kurzbeschreibung: STM32 Software der Machbarkeitsanalyse für die Kalibration  
Ordner: ledrunner/sw-camera-calibration  
Datum und Zeit: 06.08.2024 12:44  
Entwicklungsumgebung: STM32CubeIDE
8. Kurzbeschreibung: Auswertesoftware der Machbarkeitsanalyse für die Kalibration  
Ordner: src/camera-calibration  
Datum und Zeit: 08.08.2024 12:05  
Programmiersprache: C/C++
9. Kurzbeschreibung: Beispielsoftware zur Steuerung einer Kamera und des Kalibrationsgerätes  
Ordner: src/camera-control  
Datum und Zeit: 08.08.2024 12:05  
Programmiersprache: C/C++
10. Kurzbeschreibung: Evaluationssoftware der gefilmten Videos (automatische Positionsdetektion)  
Ordner: src/eval-dynamic  
Datum und Zeit: 09.08.2024 05:59  
Programmiersprache: Python
11. Kurzbeschreibung: Evaluationssoftware der gefilmten Videos (Prototyp, manuelle Positionseinzeichnung)  
Ordner: src/eval-static  
Datum und Zeit: 24.05.2024 10:23  
Programmiersprache: Python
12. Kurzbeschreibung: Oszilloskop Messresultate

Ordner: doc/measurements

Datum und Zeit: 09.08.2024 06:22

13. Kurzbeschreibung: Resultate

Ordner: doc/results

- Inhalt:
- a) 30fps-4999us-at-29.85fps-\*.png
  - b) 59-99fps-4999us-at-59fps-\*.png
  - c) 59-99fps-4999us-at-59-700699fps-\*.png
  - d) 59-99fps-4999us-at-60-299084fps-\*.png
  - e) 59-99fps-4999us-at-61fps-\*.png
  - f) 60fps-5000us-22k-\*.png
  - g) 285fps-2055us-22k\_002-\*.png
  - h) 752fps-600us-\*.png
  - i) 59-99fps-4999us-at-40-000000fps-\*.png
  - j) 59-99fps-4999us-at-50.000000fps-\*.png
  - k) 59-99fps-4999us-at-56-250000fps-\*.png
  - l) 59-99fps-4999us-at-60fps-\*.png
  - m) 59-99fps-4999us-at-64-285713fps-\*.png
  - n) 59-99fps-4999us-at-75-000000fps-\*.png
  - o) 59-99fps-4999us-at-120-000000fps-\*.png
  - p) 59-99fps-4999us-at-59-700699fps-blurry-\*.png
  - q) 60fps-at-59.700699fps-rolling-\*.png

Datum und Zeit: 09.08.2024 04:34

## **Selbstständigkeitserklärung**

"Erklärung: Ich erkläre hiermit, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und erlaubten Hilfsmittel benutzt habe, einschliesslich der Verwendung von KI-Systemen. Alle Stellen, die wörtlich oder sinngemäss aus Quellen entnommen worden sind, habe ich als solche gekennzeichnet. Ich bin den Vorgaben des Leitfadens wissenschaftliches Arbeiten gefolgt. Mir ist bekannt, dass andernfalls die Hochschulleitung zum Entzug der aufgrund meiner Arbeit verliehenen Qualifikation oder des für meine Arbeit verliehenen Titels berechtigt ist."

Ort, Datum

Unterschrift