

Entwicklung eines automatischen Timing-Messgeräts für Kameras

Bachelorthesis der Fachhochschule Graubünden

30.08.2024

Raphael Seitz

Inhaltsverzeichnis

- Aufgabenstellung
- Konzept
- Prototyp
- Hardware + Software
- Auswertung
- Resultate
- Demonstration
- Verteidigung

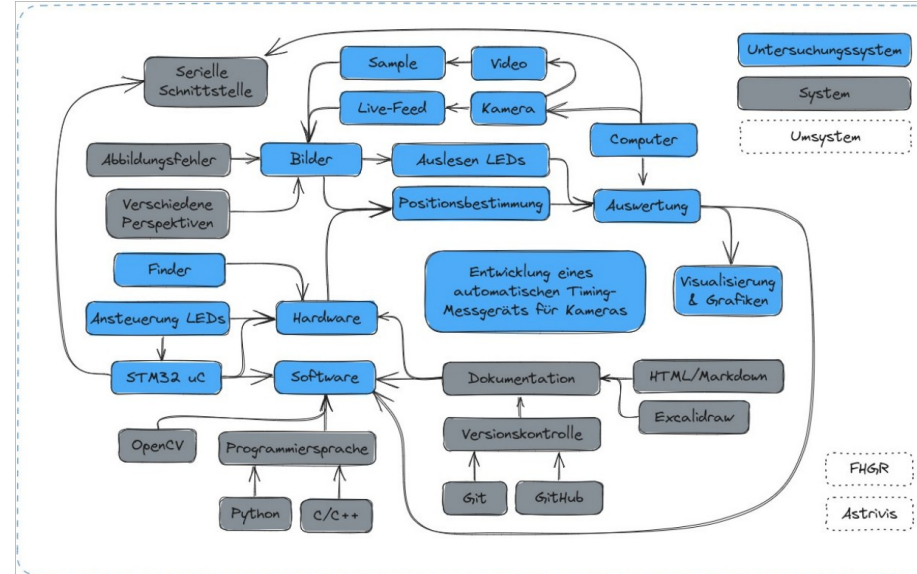


Aufgabenstellung

- Messsystem mit LEDs
- Kamera schaut auf Messsystem
- Messsystem verändert LEDs über Zeit
- Software kann aus Video Zeitverhalten analysieren
- Messsystem mit PC über serielle Schnittstelle verbinden

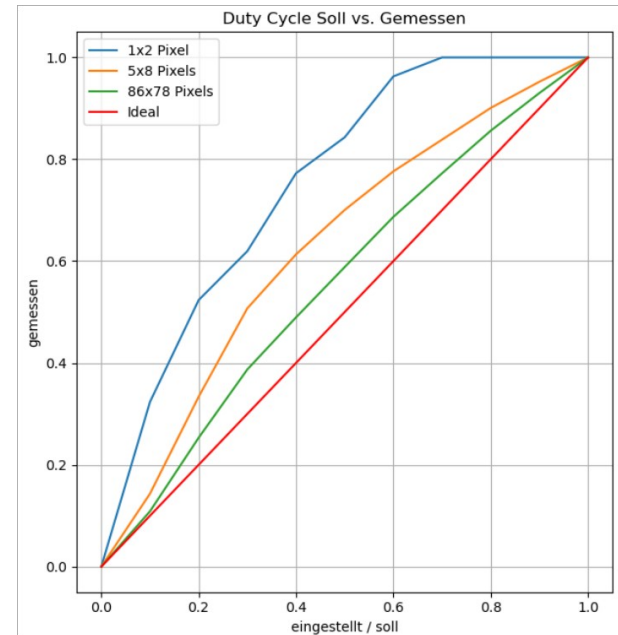
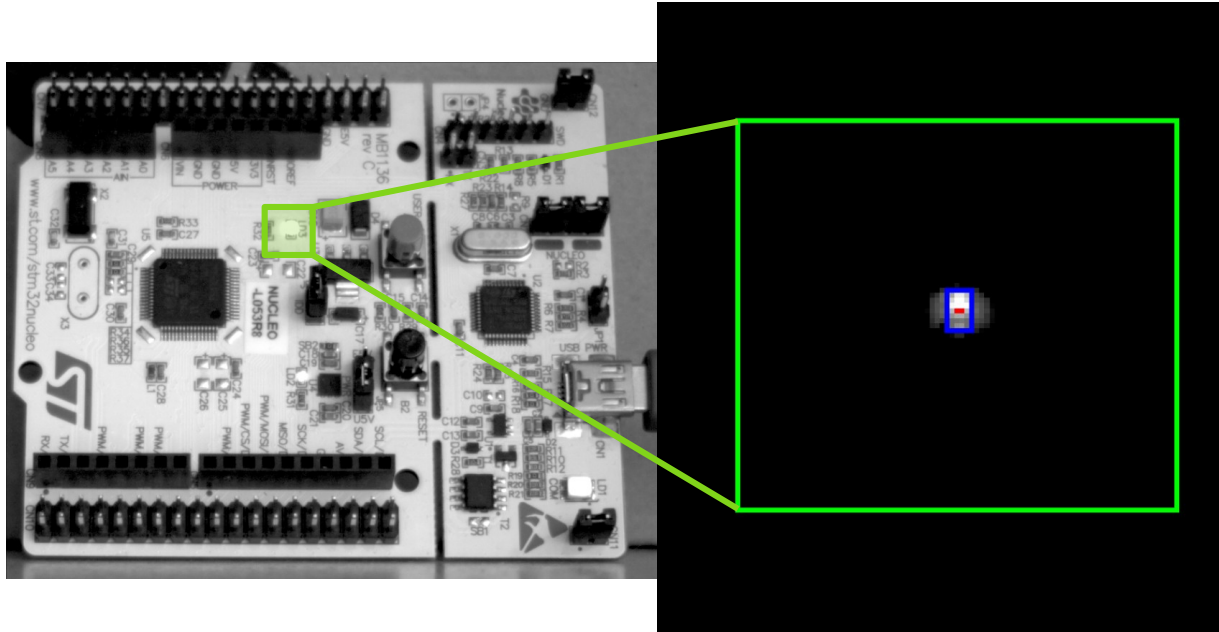
Konzept

- eine wandernde LED abgestimmt auf FPS der Kamera
- Position LEDs → Zeitstempel
- Anzahl LEDs → Beleuchtungszeit



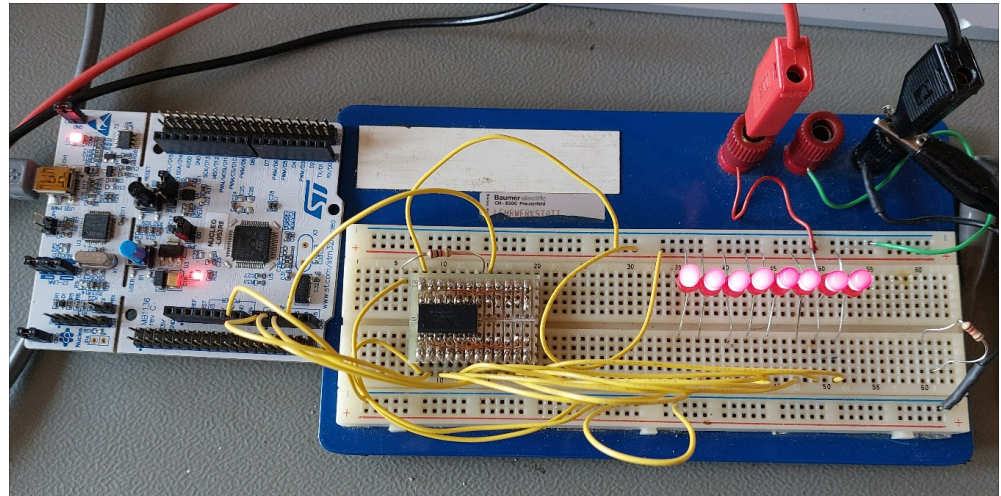
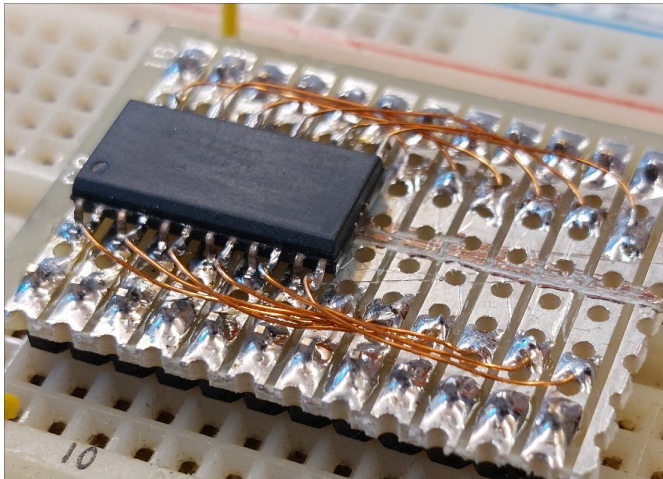
Prototyp – PWM / Testkalibration

- Einzelne LED (Nucleo Board)
- Helligkeitsauswertung bei 10 Stufen

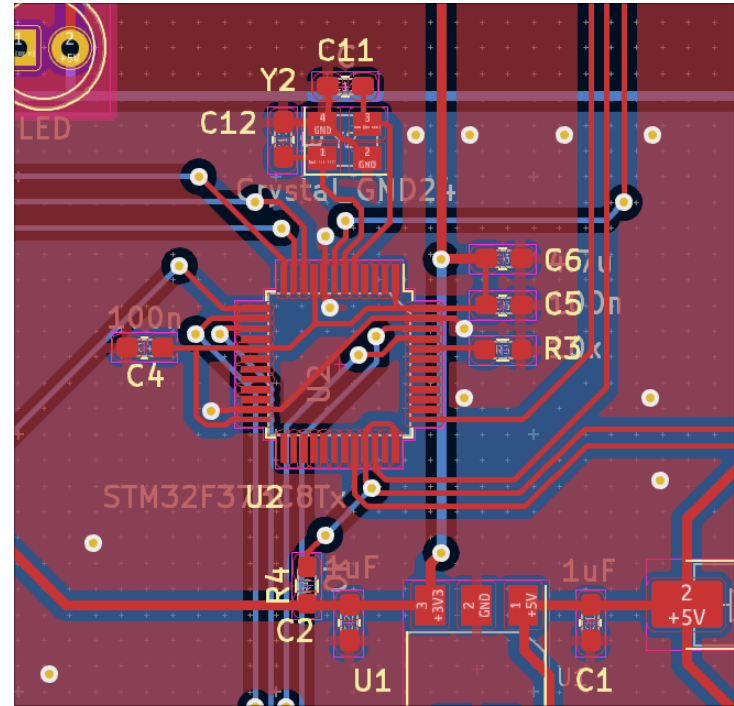


Prototyp – Schieberegister

- Aufbau auf Steckbrett
- Maximal 30MHz
- Ein Widerstand gibt LED Treiberstrom vor

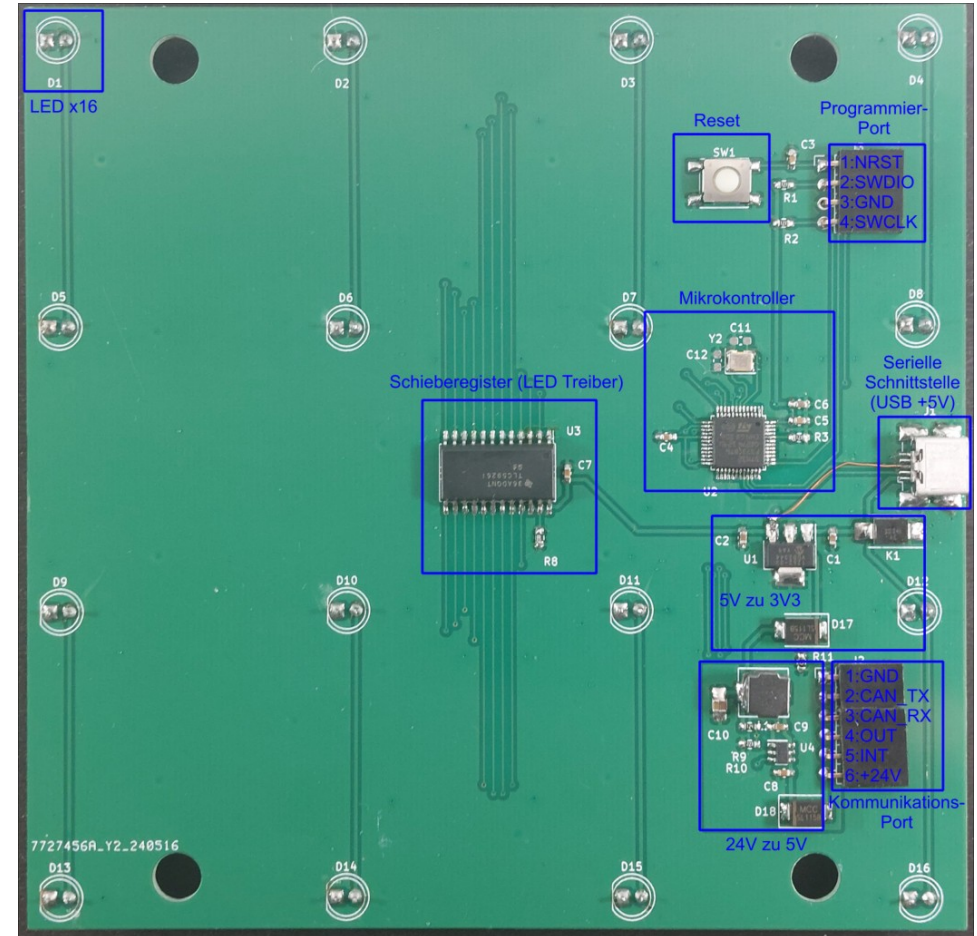


- KiCAD



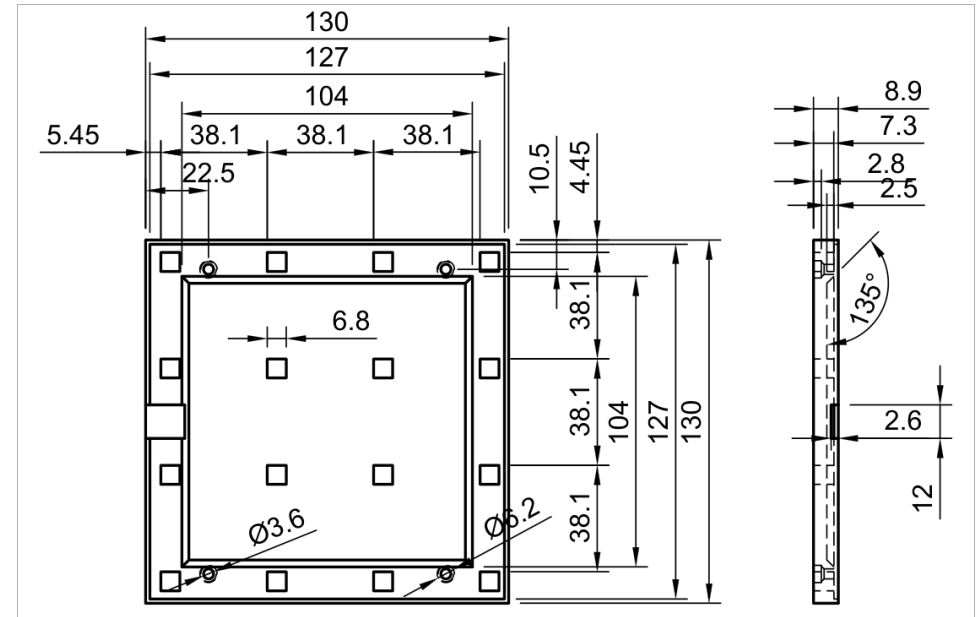
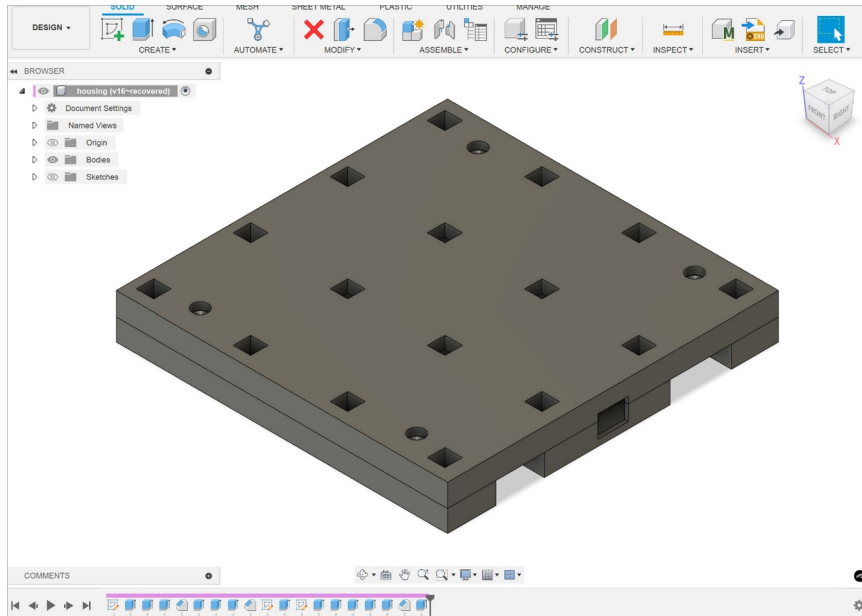
Hardware – PCB

- Bestellt bei JLCPCB



Hardware – Gehäuse

- Programm: Autodesk Fusion 360
- Drucker: Prusa MK4

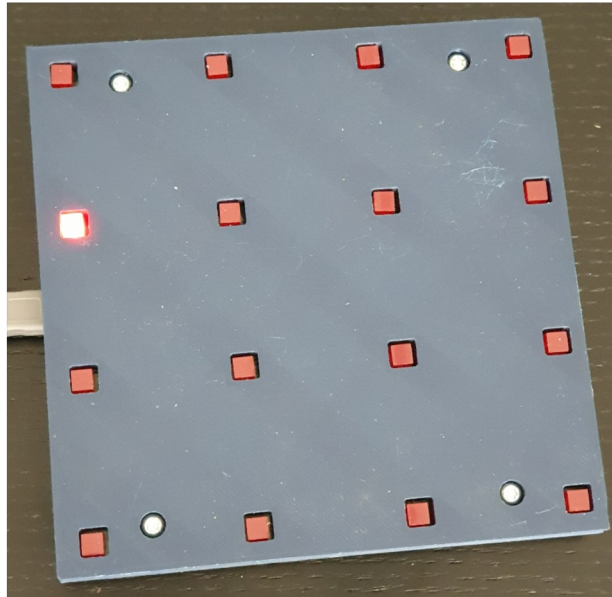


Software – Mikrokontroller (1)

- Software auf Kalibrationsgerät
- Programmiersprache: C
- Entwicklungsumgebung: STM32CubeIDE
- Steuerung über serielle Schnittstelle

Software – Mikrokontroller (2)

- Timer (Prescaler + Auto-Reload) $f_{timer} = \frac{f_{system}}{(PSC+1) \cdot (ARR+1)}$
- Schieberegister → eine LED
- (*Demonstration*)



Software – Komplette Ansteuerung

- (Beispiel) Software auf PC
- Programmiersprache: C/C++
- Ansteuerung von Kamera + Kalibrationsgerät
- Bibliotheken: opencv, ueye (Kamera)
- *(Demonstration am Schluss)*

Software – Auswertung

- Software auf PC
- Programmiersprache: Python
- Bibliotheken: opencv, numpy, scipy, matplotlib
- *(Demonstration am Schluss)*

Auswertung – Funktionsprinzip

- Änderung Position → Änderung FPS
- Breite der LEDs → Belichtungszeit
- Anordnung der LEDs “im Kreis”



Auswertung – Datengenerierung (1)

- FPS Kalibrationsgerät \neq Kamera $n = \frac{1}{2 \cdot |1 - \frac{f_K}{f_G}|}$
- Maximal einstellbare FPS = 4.5kHz
- Ganzes LED-Feld ausgeschaltet aber im Bild der Kamera
- Beginnen mit Filmen

Auswertung – Datengenerierung (2)

Übertragung von Einstellungen → Video

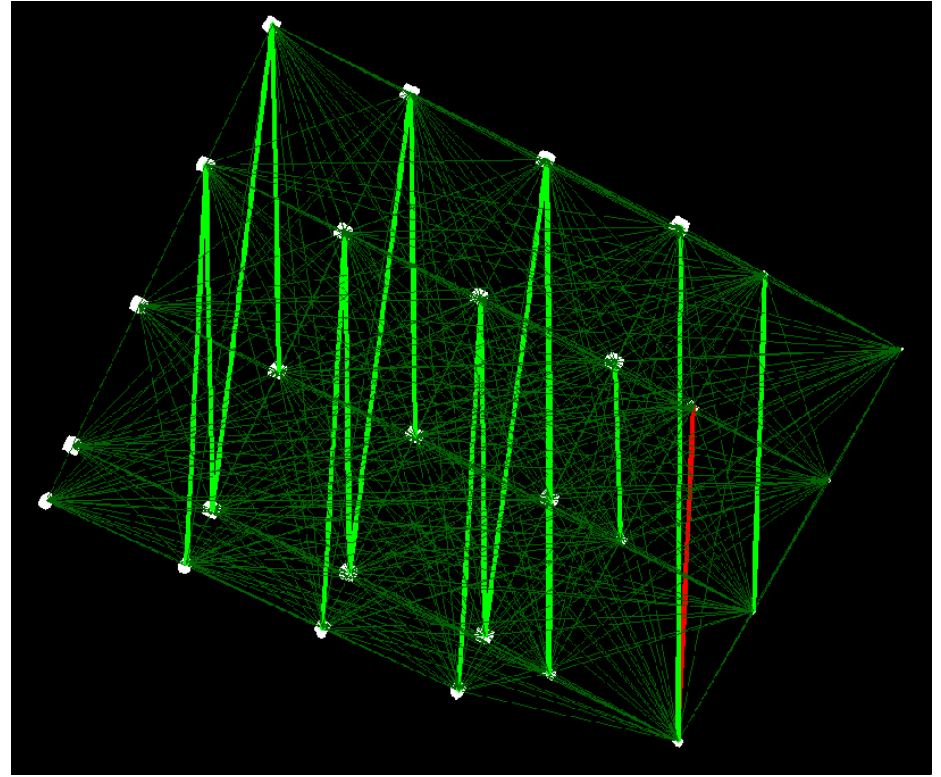
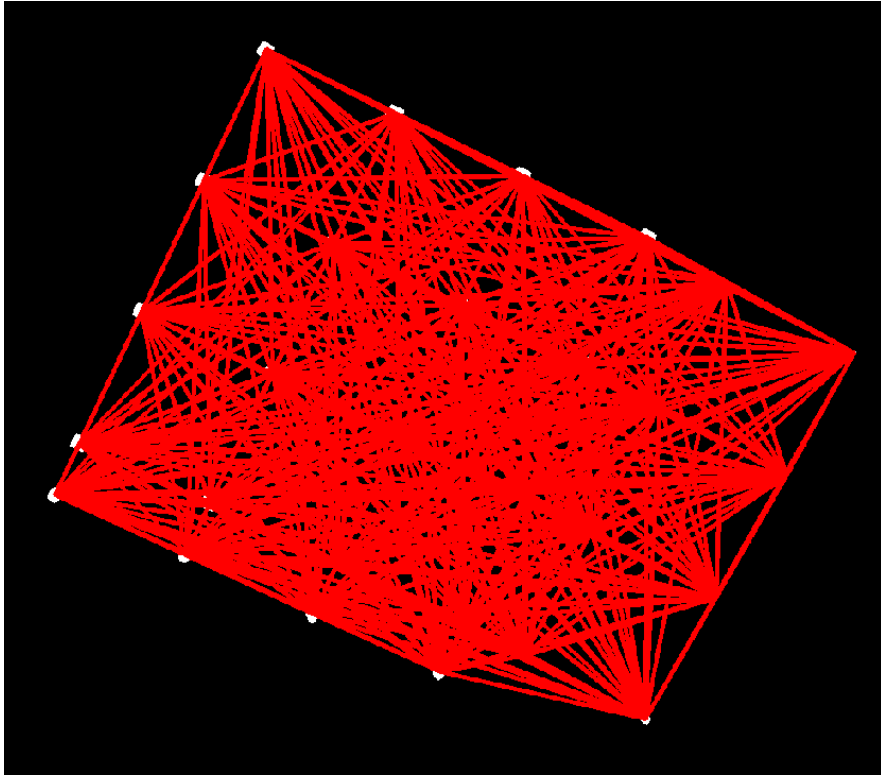
- | | |
|------------------------|--------------------------|
| 1) Alle LEDs aus | 6) Auto-Reload (Timer) |
| 2) Alle LEDs an | 7) Anzahl PWM Stufen |
| 3) Orientierungsmuster | 8) PWM Stufen |
| 4) Systemfrequenz | 9) Alle LEDs aus |
| 5) Prescaler (Timer) | 10) Wandernde LED |

(Video)

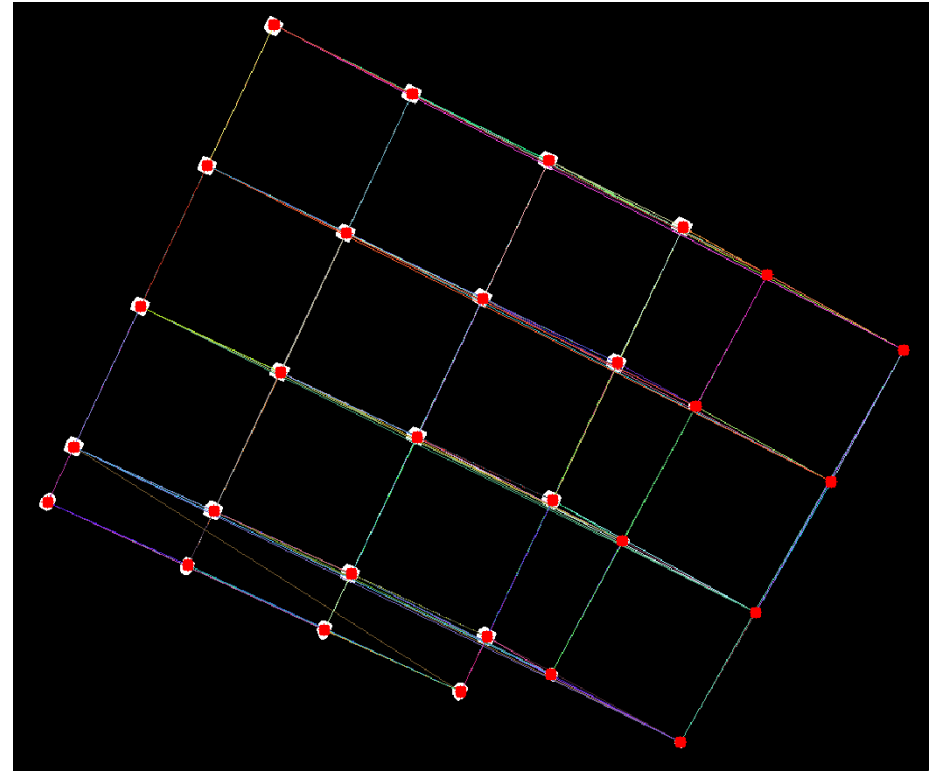
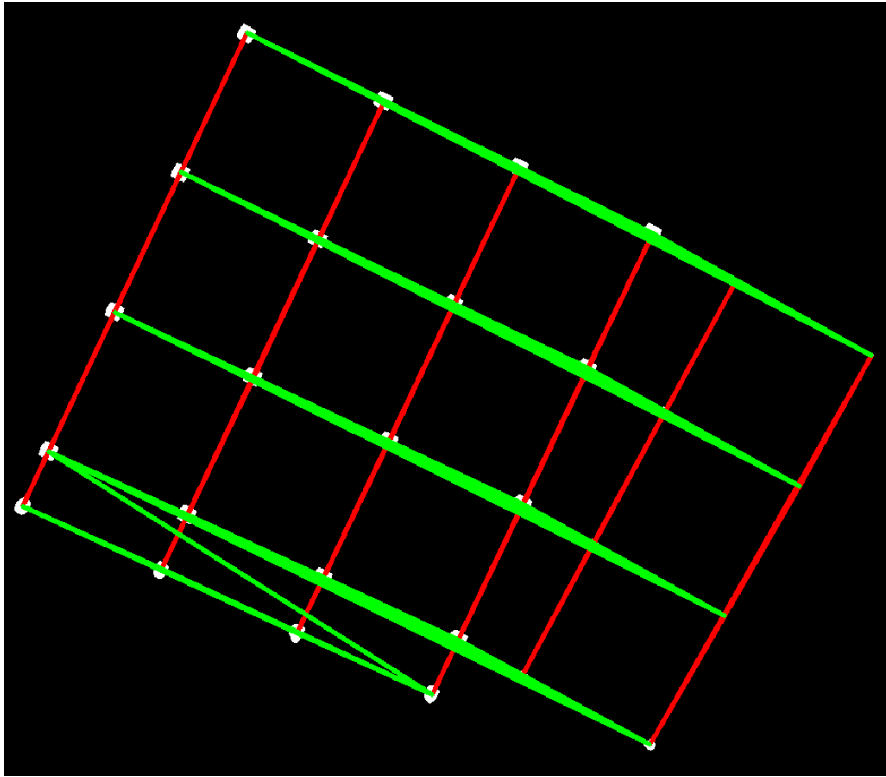
Auswertung – Datengenerierung (3)

- Filmen beenden
- Messgerät evtl. Ausschalten
- Auswertung mit Python
- PC → Zugriff auf Video

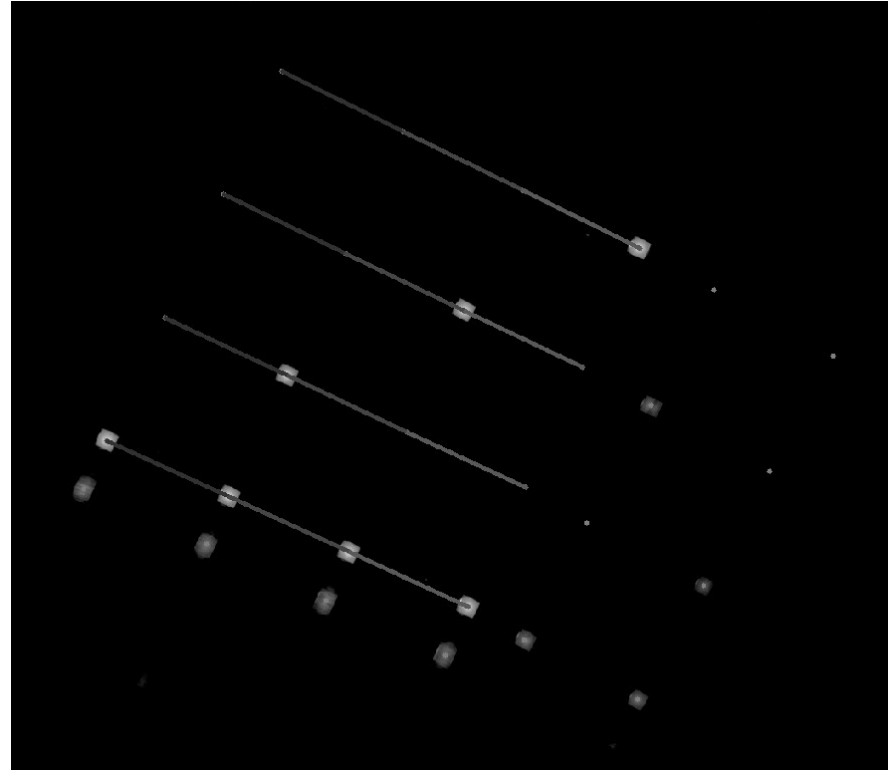
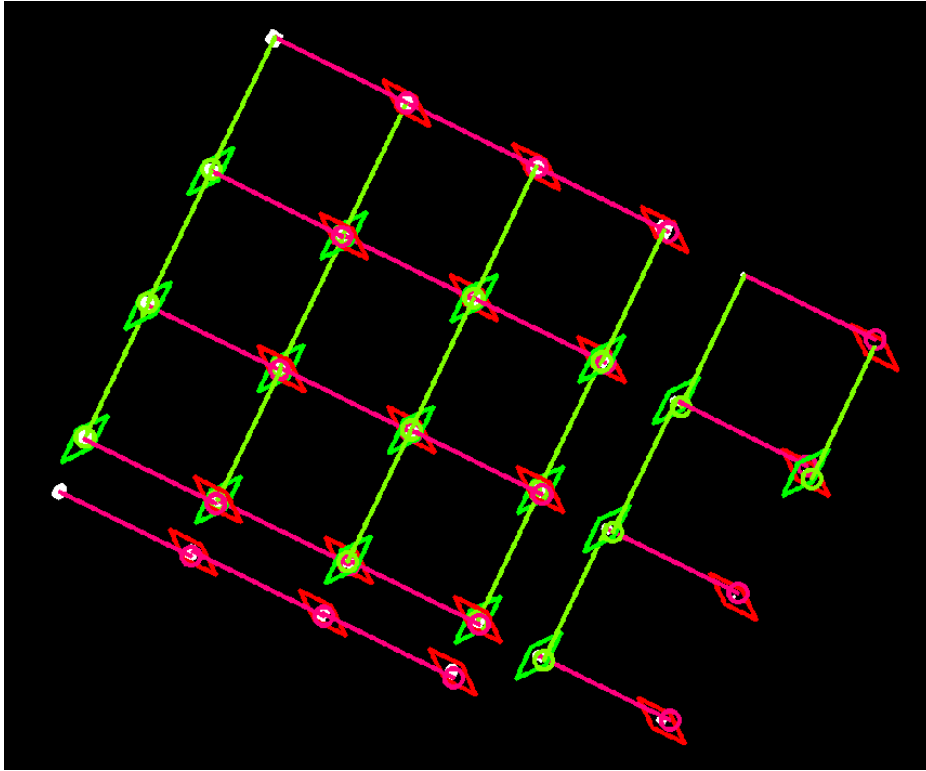
Auswertung – Positionsdetektion (1)



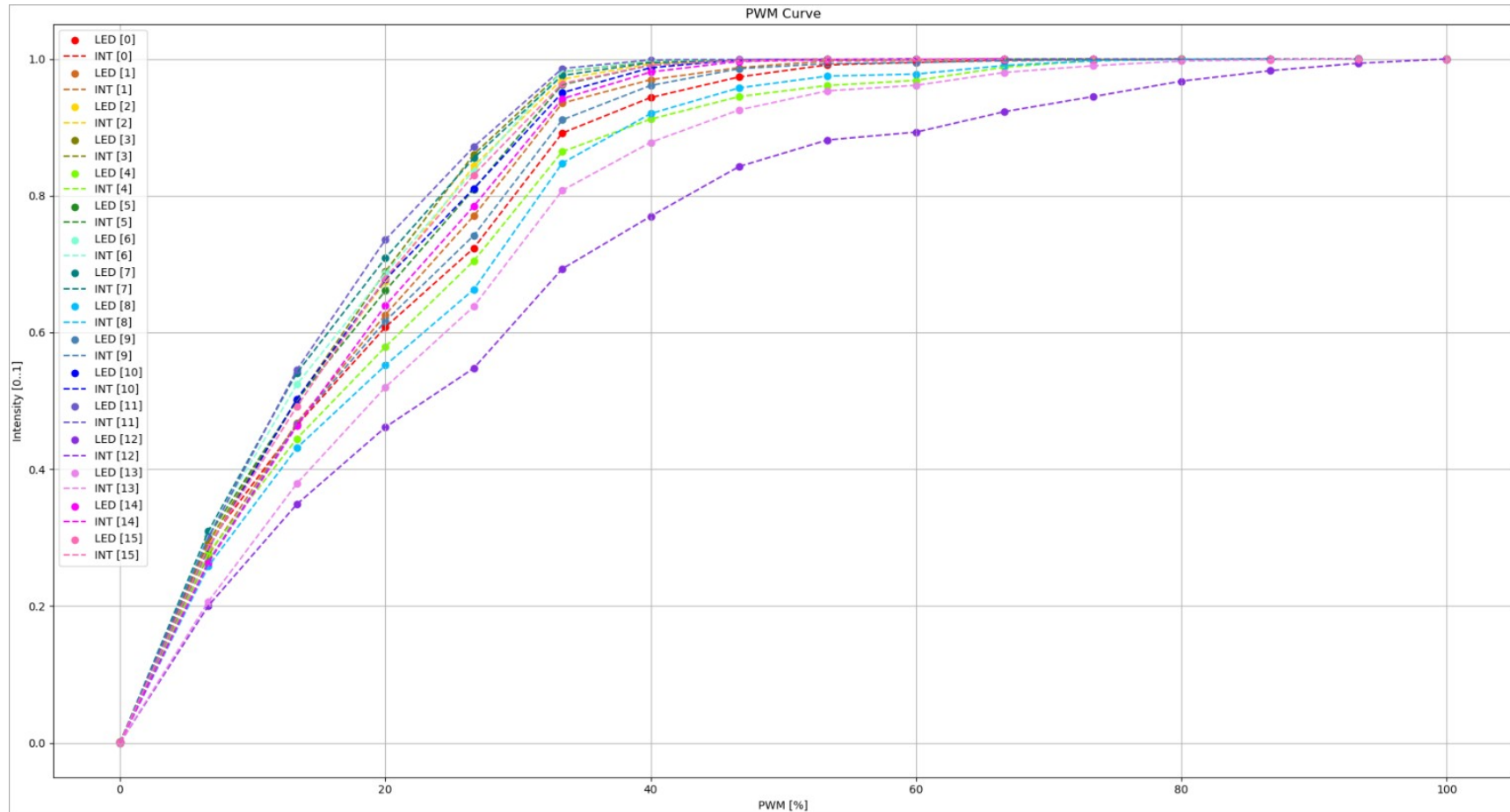
Auswertung – Positionsdetektion (2)



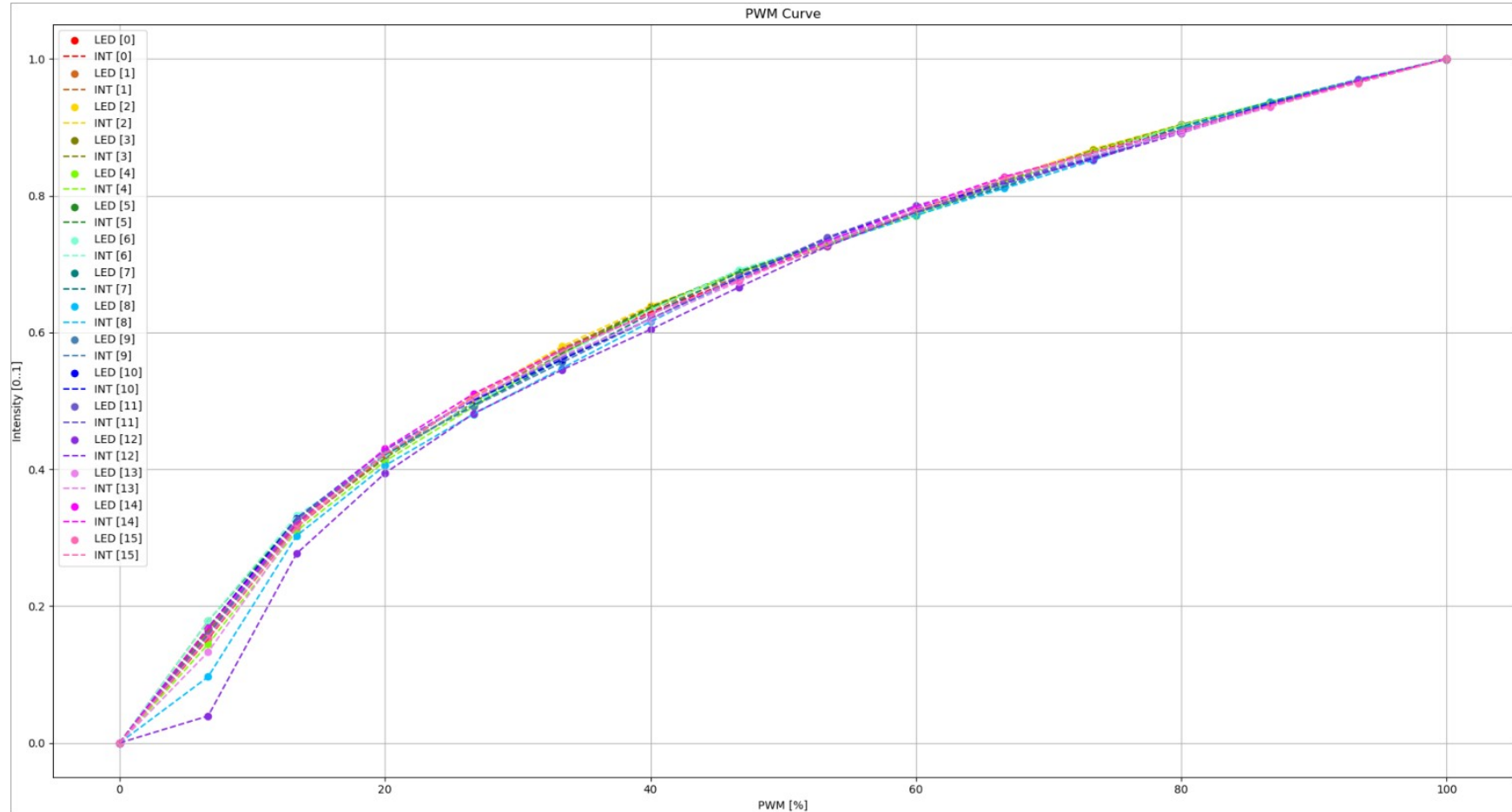
Auswertung – Positionsdetektion (3)



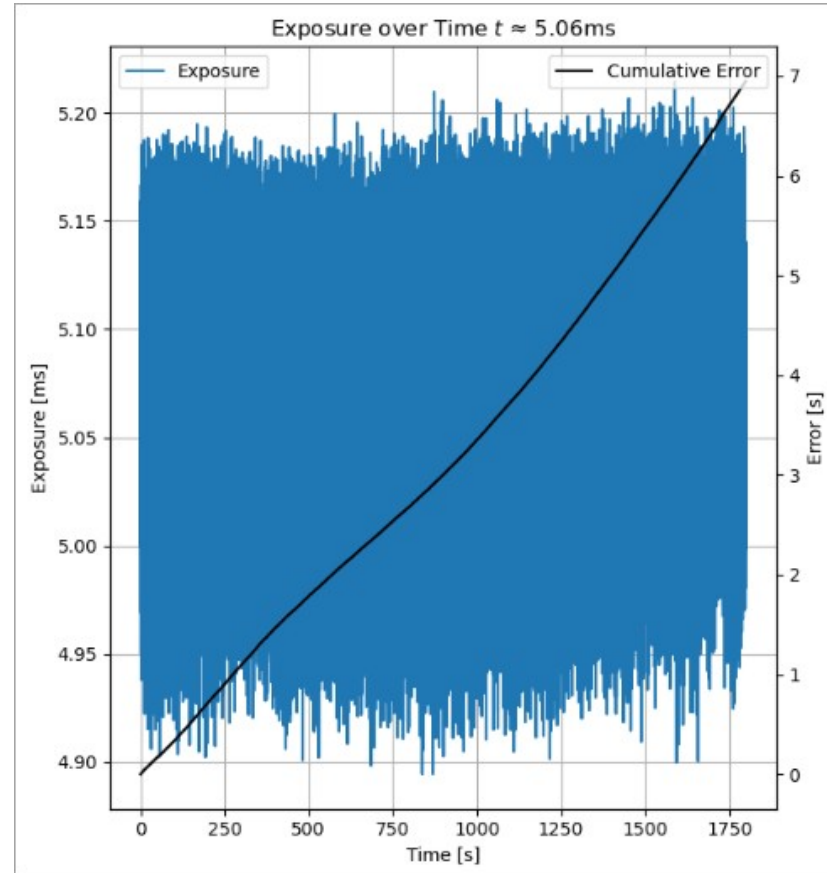
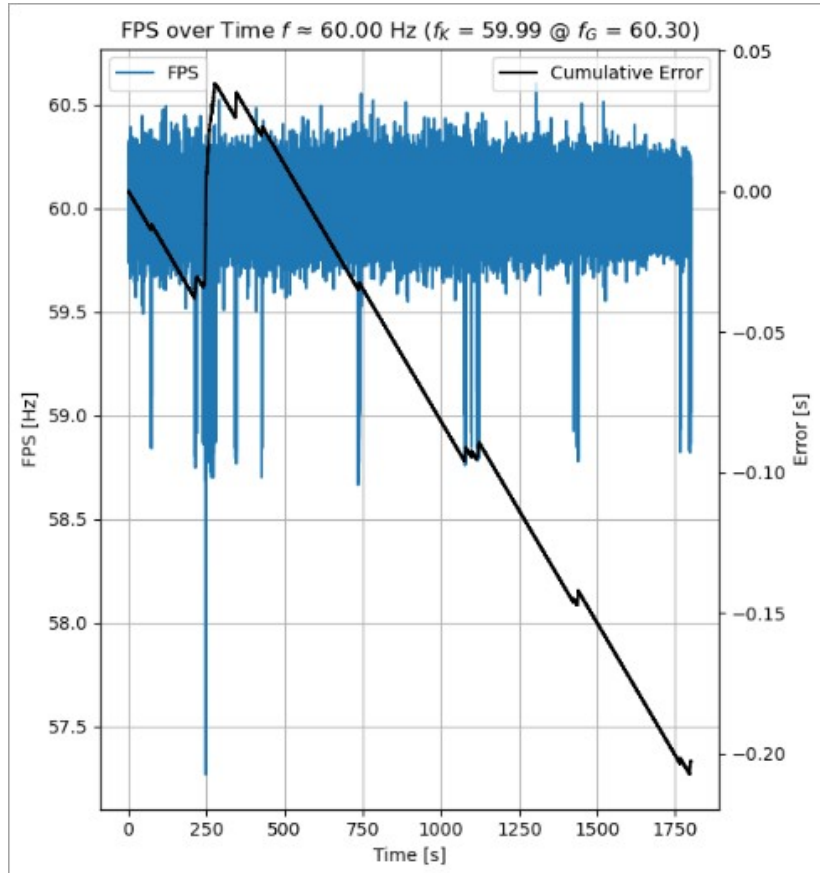
Auswertung – Kalibration (1)



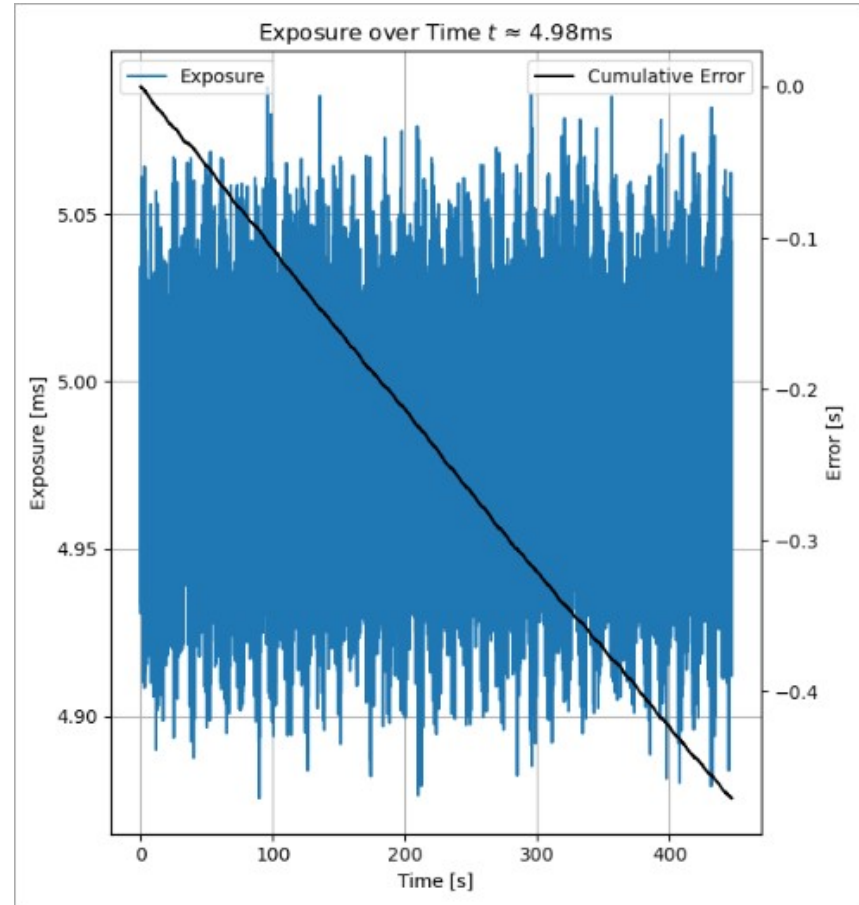
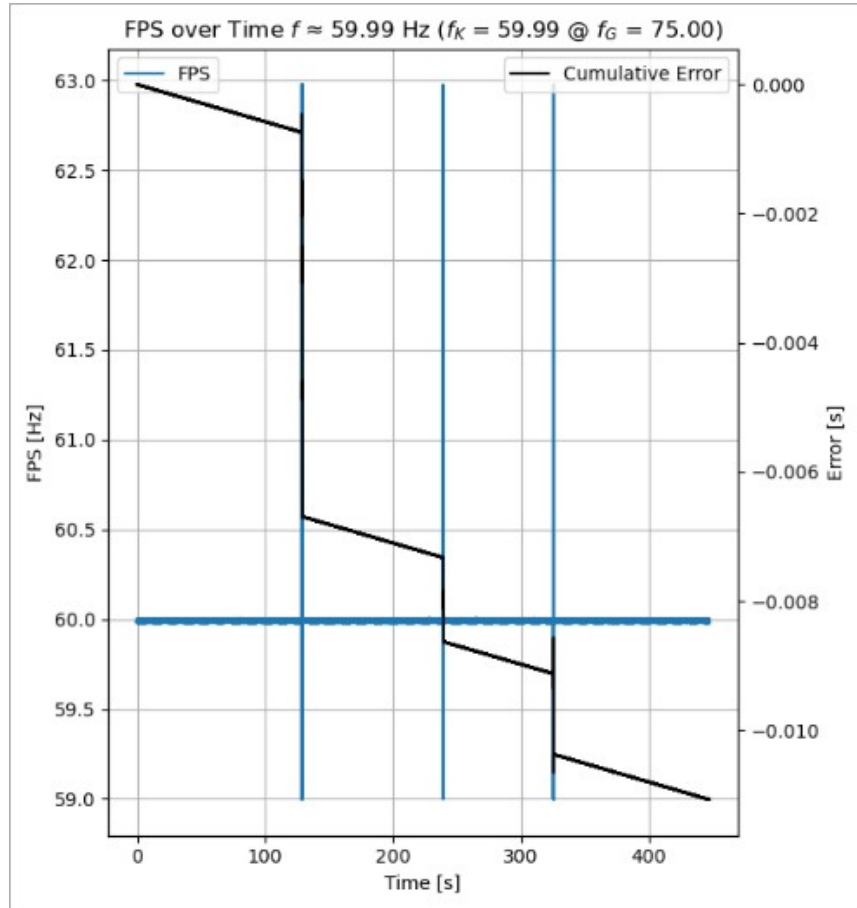
Auswertung – Kalibration (2)



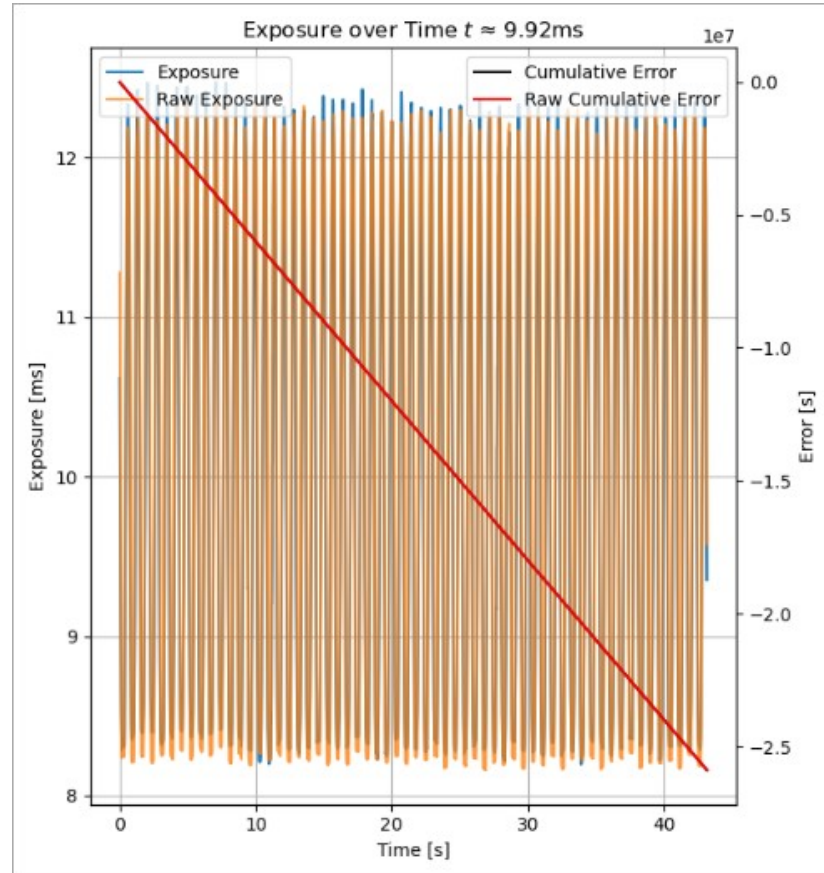
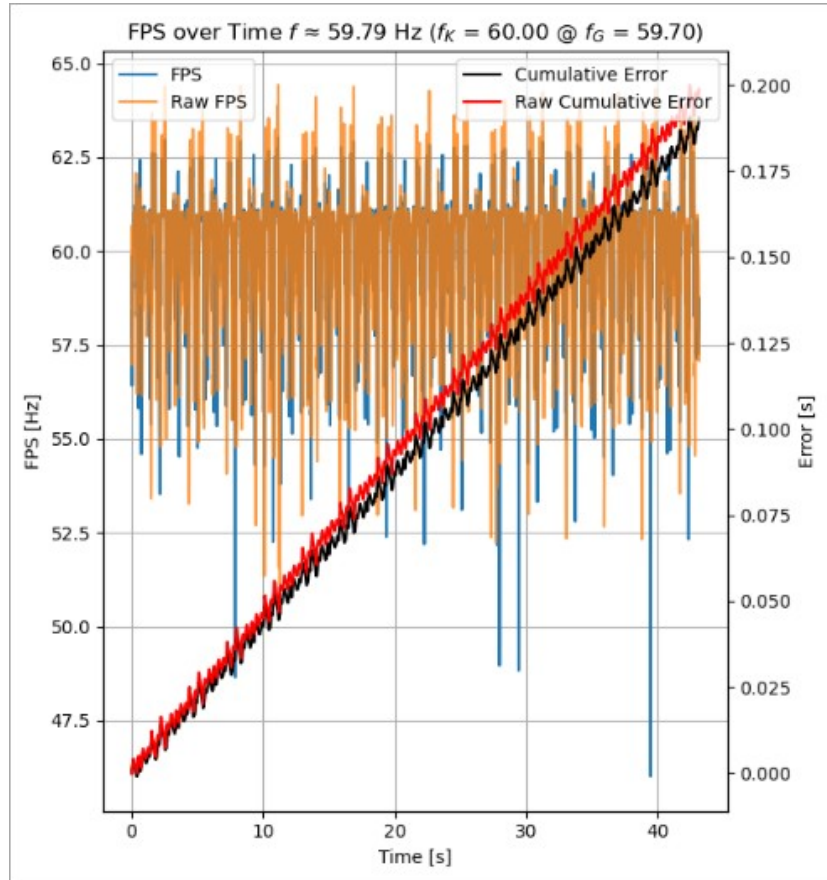
Resultate – gut abgestimmt



Resultate – schlecht abgestimmt



Resultate – Rolling Shutter *(Video)*



Open source

- GitHub
- <https://github.com/rseitz01/fhgr-bachelorthesis-camera-calibration>



Demonstration

Entwicklung eines automatischen
Timing-Messgeräts für Kameras

30.08.2024

Raphael Seitz

Vielen Dank für eure
Aufmerksamkeit

Entwicklung eines automatischen
Timing-Messgeräts für Kameras

30.08.2024
Raphael Seitz

Verteidigung

Entwicklung eines automatischen
Timing-Messgeräts für Kameras

30.08.2024

Raphael Seitz