

Dokumentation zur „Prüfungsleistung Teil 2: Rechnerprogramm“ im
Rahmen der Veranstaltung Ingenieurwissenschaftliche Applikationen

Entwicklung eines Smart Mirrors zur
Verwendung im Home Office mit Fokus auf einer
Beleuchtungssteuerung
SoSe 2023

Prüfer: Dr. Christian Hinrichs
Abgabetermin: 27.06.2023

von:
Timon Faß
Matrikelnummer: 6032129

Studiengang Ingenieurinformatik
Fachbereich Ingenieurwissenschaften
Jade Hochschule Wilhelmshaven/Oldenburg/Elsfleth
Friedrich-Paffrath-Straße 101
D-26389 Wilhelmshaven



Inhaltsverzeichnis

1	Einleitung	1
2	Entwurf	2
2.1	Funktionale und nichtfunktionale Anforderungen an die Software	2
2.2	Welche Programmumgebung oder welche Software steht bereits zur Verfügung?	3
2.3	Welche Bestandteile müssen neu erstellt oder angepasst werden?	3
2.4	Wie soll dies in einer Codestruktur abgebildet werden?	4
2.5	Welche Programmiersprache soll verwendet werden?	6
3	Implementierung	7
4	Qualitätssicherung	7
5	Auswertung	7
5.1	Ergebnisse	8
5.2	Reflexion	8
5.2.1	War die Forschungsfrage gut gewählt?	8
5.2.2	Konnten Arbeitsplan und Softwareentwurf wie geplant umgesetzt werden?	9
5.2.3	Was könnte beim nächsten Projekt anders gemacht werden?	9
5.3	Ausblick	9
6	Vorführung des Programms	10

1 Einleitung

Die Relevanz sowohl von Smart Home Technologien in eigenen Haushalten, als auch das Interesse von Arbeitnehmern am Home Office steigt immer weiter an. Dies wurde im beigefügten Forschungskonzept in der Datei **Forschungskonzept_Faß_Timon_SoSe23.pdf** bereits dargelegt. Deshalb werden im Zuge dieser Aufgabe beide Bereiche miteinander kombiniert. Es entsteht ein Smart Mirror, der im Home Office verwendet werden kann, um neben den standardmäßigen Funktionen eine Beleuchtungssteuerung des Raumes zu ermöglichen. Spiegel sind in privaten Haushalten weit verbreitet und bieten somit eine Möglichkeit, die Hausautomatisierung weiter voranzubringen, ohne das Haus oder die Wohnung mit Technik zu überfüllen. Für das Home Office bietet sich die Umsetzung einer Beleuchtungssteuerung an, da die Beleuchtung an Arbeitsplätzen oftmals genormt ist, zu Hause jedoch keinen Richtlinien unterliegt.

Die Aufgabenstellung gilt als erfüllt, da folgende Ziele des Projektes erreicht werden konnten: Es liegt eine graphische Benutzeroberfläche (GUI) vor, die für einen solchen Smart Mirror verwendet werden kann. Diese beinhaltet grundlegende Funktionen eines Smart Mirrors, wie Anzeige des Datums, der Uhrzeit und des Wetters und eine Kontrolloberfläche für die Beleuchtung. Die Steuerung der Beleuchtung erfolgt mit einem Arduino, der über Bluetooth mit der GUI kommuniziert und RGB-Leuchtdioden ansteuert.

In dieser Dokumentation wird sich in Kapitel 2 zunächst mit dem Entwurf befasst. Dieser enthält neben den funktionalen und nichtfunktionalen Anforderungen die bereits bestehenden und neu zu erstellenden Bestandteile der Software, sowie die grundlegende Codestruktur und die verwendeten Programmiersprachen. Kapitel 3 behandelt die Implementierung, während Kapitel 4 auf die Qualitätssicherung eingeht. Schließlich ist in Kapitel 5 die Auswertung mit den Ergebnissen, der Reflexion und dem Ausblick aufgeführt und Kapitel 6 beschreibt die Vorführung des Rechnerprogramms.

Aus Gründen der besseren Lesbarkeit wird im Folgenden auf die gleichzeitige Verwendung weiblicher und männlicher Sprachformen verzichtet und das generische Maskulinum verwendet. Sämtliche Personenbezeichnungen gelten gleichermaßen für alle Geschlechter.

2 Entwurf

Für den Entwurf wird auf das beiliegende Forschungskonzept in der Datei **Forschungskonzept_Faß_Timon_SoSe23.pdf** verwiesen. Dieses enthält in Kapitel 2 die theoretischen Voraussetzungen und in Kapitel 5 die verwendete Methodik. Bei diesem Projekt handelt es sich um eine reine Softwareentwicklung einer GUI und eines Arduino Programmes, der keine Algorithmen zugrunde liegen.

An dieser Stelle wird der Entwurf mit funktionalen und nichtfunktionalen Anforderungen, Programmumgebung, Software, Codestruktur und Programmiersprachen in den Kapiteln 2.1 bis 2.5 genauer betrachtet.

2.1 Funktionale und nichtfunktionale Anforderungen an die Software

Es wird eine GUI in Python entwickelt, welche die grundlegenden Funktionen eines Smart Mirrors erfüllt und darüber hinaus eine Beleuchtungssteuerung ermöglicht. Die Beleuchtung wird in einem Modell mit RGB-Leuchtdioden und einem Arduino umgesetzt. Der Modellaufbau enthält darüber hinaus einen RGB-Sensor, der die Beleuchtungsstärke und Farbtemperatur misst. Zwischen der GUI und dem Arduino besteht eine Bluetooth Verbindung.

Funktionale Anforderungen:

- Graphische Benutzeroberfläche mit Basisfunktionen eines Smart Mirror (Anzeigen des Wetters, Uhrzeit, Datum)
- Benutzeroberfläche für Beleuchtungssteuerung mit Buttons für unterschiedliche Lichtmodi und Anzeige der gemessenen Beleuchtungsstärken und Farbtemperaturen
- Bluetooth Verbindung zwischen Python GUI und Arduino
- Arduino mit Programm, das Bluetooth Nachrichten empfängt und sendet und entsprechend unterschiedliche Lichtmodi an den LEDs steuert.

Nichtfunktionale Anforderungen:

- Die GUI muss sich zur Laufzeit ständig aktualisieren
- Die GUI muss so robust sein, dass sie im Dauerbetrieb laufen könnte
- Die GUI muss übersichtlich sein
- Die GUI läuft auch, ohne dass der Arduino eingeschaltet ist

2.2 Welche Programmumgebung oder welche Software steht bereits zur Verfügung?

Einige Grundlagen und erste Ansätze liegen bereits vor. Als erstes bestehen bereits Vorerfahrungen mit Smart Mirrors. Diese stammt aus der Mitarbeit an einem Projekt, indem ein „Social Mirror“ entwickelt wurde. Dieser ist ein Smart Mirror, der ein gesundheitliches Monitoring für den Benutzer unter Verwendung eines Fitnessarmbandes bietet. Das Armband und das System selbst stehen nicht mehr zur Verfügung, jedoch ist der Python Code noch immer vorhanden. Aus diesem können Grundlagen für den Aufbau einer GUI des Smart Mirrors und die Umsetzung der standardmäßigen Funktionen entnommen werden. Dazu zählen das Anzeigen der Uhrzeit, des Datums und des aktuellen Wetters.

Darüber hinaus bestehen Vorerfahrungen mit einer Beleuchtungssteuerung über einen Arduino. Es steht Arduino Code zur Verfügung, der eine Regelung von Innenbeleuchtung in Abhängigkeit von der Außenbeleuchtung umsetzt. Diese kann über Bluetooth mit einer Android App gesteuert werden. Die Beleuchtung wird hierbei mit RGB-Leuchtdioden umgesetzt. Hier steht neben der verwendeten Software auch die Hardware zur Verfügung. Der Code kann verwendet werden, um verschiedene Farbtemperaturen und Beleuchtungsstärken einzustellen und diese mit einem RGB-Farbsensor zu überprüfen.

2.3 Welche Bestandteile müssen neu erstellt oder angepasst werden?

Um das Projekt wie beschrieben umzusetzen, müssen die Vorarbeiten aus Kapitel 2.2 angepasst werden. Es ist nicht nötig, die gesamte Software neu zu entwickeln. Der Arduino Code wird so angepasst, dass RGB-Leuchtdioden gesteuert werden können, indem Bluetooth Nachrichten an das System gesendet werden. Außerdem werden Nachrichten über Bluetooth zurückgegeben, die den jeweiligen Lichtmodus und die gemessenen Wer-

te des Sensors enthalten.

Auch die Python GUI muss lediglich für das Projekt angepasst werden. Die Grundfunktionen eines Smart Mirrors sind bereits vorhanden. Nicht benötigte Funktionen werden entfernt und durch Buttons und einen Ausgabebereich für die Beleuchtungssteuerung ersetzt. Neu zu entwickeln ist die Kommunikation über Bluetooth zwischen der Python GUI und dem Arduino Programm. Hierfür kann auf Python Bibliotheken zurückgegriffen werden. Außerdem muss die vorhandene Hardware für die Beleuchtung mit Arduino, Bluetooth Modul, RGB-Sensor und RGB-LEDs neu aufgebaut werden.

2.4 Wie soll dies in einer Codestruktur abgebildet werden?

Basierend auf den Anforderungen in Kapitel 2.1 ist es nötig, zwei Softwarelösungen zu implementieren. Eine erfüllt den Zweck, die GUI zu erstellen, die andere läuft auf dem Mikrocontroller und steuert das Licht der RGB-LEDs. Außerdem ist eine Bluetooth Verbindung zwischen beiden Programmen nötig.

Der Code für die GUI muss so strukturiert sein, dass er Funktionen enthält, die jeweils folgende Aufgaben erfüllen:

Darstellen einer graphischen Benutzeroberfläche (GUI). Wird das Programm ausgeführt erscheint eine GUI mit den Basisfunktionen eines Smart Mirrors auf dem Bildschirm. Die GUI hat einen schwarzen Hintergrund, damit sich hinter einer Spiegelfolie nur die Schrift und anderen Elemente abheben würden. Außerdem wird sie im Vollbild Modus ausgeführt, um die volle Fläche des Bildschirms einzunehmen. Die Basisfunktionen beinhalten das Anzeigen des Datums, der Uhrzeit und des Wetters für den Ort Wilhelmshaven. Diese GUI wird in einer Klasse mit gleichem Namen im Code hinterlegt. In dieser Klasse werden Funktionen verwendet, welche die jeweiligen Basisfunktionen implementieren.

Benutzeroberfläche für die Beleuchtungssteuerung in der GUI. Um die Beleuchtungssteuerung zu realisieren, werden der GUI Buttons für verschiedene Funktionalitäten hinzugefügt. Diese Buttons sind im Code mit Funktionen verknüpft, welche die jeweilige Funktionalität implementieren. Die Buttons sind im Folgenden mit ihren Beschriftungen aufgeführt:

- Ein-/Ausblenden: Dieser Button kann geklickt werden, um den Ausgabebereich und alle anderen Buttons zu verstecken und wieder einzublenden, damit die Übersichtlichkeit der GUI gewährleistet ist.

- **Kaltweiß:** Dieser Button aktiviert einen voreingestellten Lichtmodus, der kaltweißes Licht an den RGB-LEDs einschaltet, indem die Funktion zum Senden und Empfangen von Bluetooth Nachrichten aufgerufen wird.
- **Helles Weiß:** Dieser Button aktiviert einen voreingestellten Lichtmodus, der helles weißes Licht an den RGB-LEDs einschaltet, indem die Funktion zum Senden und Empfangen von Bluetooth Nachrichten aufgerufen wird.
- **Tageslichtweiß:** Dieser Button aktiviert einen voreingestellten Lichtmodus, der tageslichtweißes Licht an den RGB-LEDs einschaltet, indem die Funktion zum Senden und Empfangen von Bluetooth Nachrichten aufgerufen wird. Dieses Licht erscheint im Vergleich zu den anderen Modi wärmer.
- **OFF:** Dieser Button schaltet das Licht der RGB-LEDs wieder aus.

Außerdem beinhaltet diese Benutzeroberfläche Labels, um die Messwerte des Sensors für den Benutzer in einem Ausgabebereich darzustellen.

Senden und Empfangen von Bluetooth Nachrichten. Diese Funktion wird aufgerufen um Bluetooth Nachrichten an den Mikrocontroller zu senden und von ihm zu empfangen, indem unter anderem die Nachricht übergeben wird. Des Weiteren wird in dieser Funktion der Text der entsprechenden Ausgabelabels mit den Messwerten versehen. Die weiteren übergebenen Parameter sind in der Kommentierung des Codes in der Datei **Smart_Mirror_GUI.py** aufgeführt.

Der Code für den Mikrocontroller muss so strukturiert sein, dass er Funktionen enthält, die jeweils folgende Aufgaben erfüllen:

Senden und Empfangen von Bluetooth Nachrichten. Diese Funktion enthält ein Switch-Case-Statement, welches den Lichtmodus entsprechend der Bluetooth Nachricht der Python GUI verändert und die Messwerte printet.

Setzen der RGB-Lichtzusammensetzung für die LEDs. Dieser Funktion werden entsprechend des Lichtmodus die voreingestellten Werte für den Rot-, Grün- und Blauanteil des Lichtes übergeben und gibt diese an die analogen Pins der LEDs weiter.

Auslesen des RGB-Farbsensors. Diese Funktion nutzt weitere Funktionen der vom Hersteller des Sensors zur Verfügung gestellten Bibliothek, um die Messwerte des Sensors auszulesen.

Der gesamte Softwareentwurf für die Codestruktur ist in Abbildung 1 dargestellt. Alle Übergabe- und Rückgabeparameter der Funktionen, sowohl im Code der GUI als auch des Mikrocontrollers sind in der Dokumentation des jeweiligen Code-Files aufgeführt.

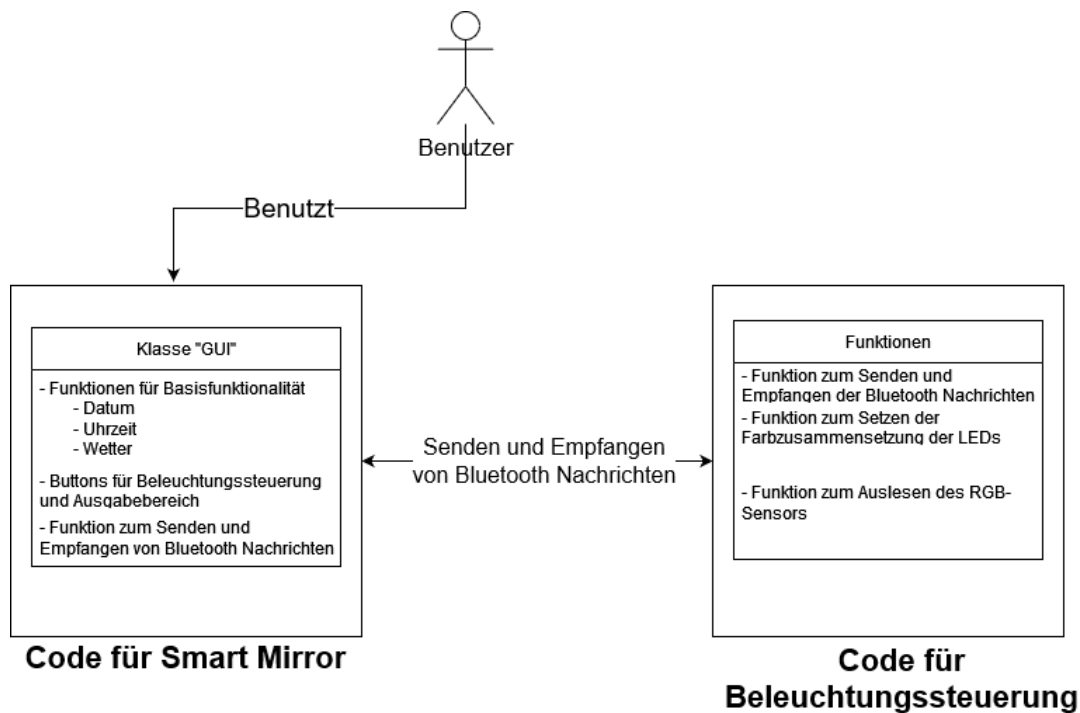


Abbildung 1: Darstellung des Softwareentwurfes.

2.5 Welche Programmiersprache soll verwendet werden?

Um die Kompatibilität auf möglichst vielen Geräten zu gewährleisten, ist eine Programmiersprache zu wählen, die weit verbreitet ist. Außerdem sollen wissenschaftliche Fragestellungen mit dieser gut umsetzbar und Möglichkeiten zum Erstellen einer GUI vorhanden sein. Unter Berücksichtigung der Vorerfahrung des Autors wird **Python** gewählt. Für die Programmierung des Arduino wird **C/C++** gewählt, da die Arduino IDE dem Benutzer diese Programmiersprache vorschreibt.

3 Implementierung

Die Implementierung wurde, entsprechend der Codestruktur in Kapitel 2.4, sowohl in Python als auch C/C++ umgesetzt. Details finden sich in der **README.md**, sowie der Kommentierung der Programme in den Dateien **Smart_Mirror_GUI.py**, **Smart_Mirror_Light.ino**, **RGB_and_Sensor_Functions.ino** und **Bluetooth_Function.ino**.

Zusätzlich zum ursprünglichen Entwurf der GUI wurde dem Ausgabebereich für die Messwerte des Sensors eine Interpretation hinzugefügt. Diese zeigt dem Benutzer auf Basis der Messwerte und der Norm DIN EN 12464-1 an, ob die Beleuchtung ausreichend hell oder zu dunkel ist.

4 Qualitätssicherung

Die Qualitätssicherung wurde durch eine konstante Verwendung des Versionsverwaltungstools Git im Laufe der Entwicklung gewährleistet. Die GUI und das Arduino Programm wurden bei jeder Änderung händisch getestet. Nach Abschluss des Projektes erfolgte ein letzter händischer Funktionstest. Automatische Tests sind bei diesem Projekt nicht vorhanden, da es eine geringe Komplexität vorweist und keine Funktionalität überprüfbare Werte generiert.

5 Auswertung

In diesem Abschnitt wird die Dokumentation des Programmes dargestellt, indem in Kapitel 5.1 bis 5.3 die Ergebnisse, die Reflexion und der Ausblick betrachtet werden. Während an dieser Stelle alle Ergebnisse allgemein zusammengefasst werden, sind alle Dokumentationen des Programmcodes in der Kommentierung der jeweiligen Code-Files vorzufinden. Die Python GUI befindet sich in der Datei **Smart_Mirror_GUI.py**, während der Arduino Code auf die Dateien **Smart_Mirror_Light.ino**, **RGB_and_Sensor_Functions.ino** und **Bluetooth_Function.ino** aufgeteilt ist. Weitere Informationen finden sich zudem in der Datei **README.md**.

5.1 Ergebnisse

Um die Forschungsfrage zu beantworten, wurde der Softwareentwurf aus Kapitel 2 implementiert. Damit sind alle Anforderungen erfüllt. Es liegt ein Python Programm vor, das eine GUI mit den Basisfunktionen, Datum, Uhrzeit und Wetter erstellt. Außerdem bietet es eine Benutzeroberfläche für die Beleuchtungssteuerung. Diese ist in Form von Buttons für die unterschiedlichen Lichtmodi und einem Ausgabebereich für die Darstellung der Messwerte und einer Interpretation gegeben. Das Python Programm kann eine Bluetooth Verbindung zum Bluetooth Modul aufbauen, dass an einem Arduino betrieben wird. Durch das Senden von Nachrichten lassen sich mit dem Programm auf dem Arduino verschiedene Lichtmodi an den RGB-LEDs aufrufen. Außerdem misst der RGB-Sensor die Beleuchtungsstärke und Farbtemperatur des Lichtes und sendet diese Werte über Bluetooth zurück an das Python Programm.

In Bezug auf die Forschungsfrage „Entwicklung eines Smart Mirrors zur Verwendung im Home Office mit Fokus auf einer Beleuchtungssteuerung“ liegt der Smart Mirror durch eine Python GUI vor. Die Beleuchtungssteuerung im Home Office wird ebenfalls durch diese GUI und einen Versuchsaufbau mit Arduino Mikrocontroller realisiert. Entsprechend der Definition der Forschungsfrage in Kapitel 4 des Forschungskonzeptes in der Datei **Forschungskonzept_Faß_Timon_SoSe23.pdf** konnte diese somit beantwortet werden.

5.2 Reflexion

Das Forschungsprojekt konnte wie geplant umgesetzt werden. Die folgenden Kapitel 5.2.1 bis 5.2.3 betrachten einige Aspekte der Reflexion genauer.

5.2.1 War die Forschungsfrage gut gewählt?

Die Forschungsfrage war gut gewählt. Alle in Kapitel 2 beschriebenen Anforderungen konnten wie geplant umgesetzt werden, weshalb die Frage als beantwortet gilt. Damit war die Forschungsfrage für den zur Verfügung stehenden Zeitraum ausreichend eingegrenzt. Trotzdem bietet sie genügend Freiheiten um sie im Anschluss weiterzuführen und beispielsweise Aspekte anzugehen, die in Kapitel 5.3 betrachtet werden. Somit könnte dieses Projekt schlussendlich dazu genutzt werden, einen echten Smart Mirror zur Beleuchtungssteuerung im Home Office zu konstruieren.

5.2.2 Konnten Arbeitsplan und Softwareentwurf wie geplant umgesetzt werden?

Der Arbeitsplan, der im Forschungskonzept in der Datei **Forschungskonzept_Faß_Timon_SoSe23.pdf** in Kapitel 5.3 dargestellt ist, hat sich als gut ausgearbeitet erwiesen. Für jeden Arbeitsschritt wurde ausreichend Zeit eingeplant, sodass die Abarbeitung nach Plan funktionierte. Hierbei war hilfreich, dass im Falle von Problemen andere Lösungsansätze vorgeschlagen und Pufferwochen angeboten wurden. Bei Problemen im Initialisieren der Bluetooth Verbindung konnten diese genutzt werden, sodass zeitlich keine weiteren Probleme entstanden sind.

Der Softwareentwurf konnte ebenfalls wie geplant umgesetzt werden. Alle in Kapitel 2.4 beschriebenen Funktionalitäten sind im Ergebnis vorhanden. Jedoch war es notwendig zur Umsetzung der Bluetooth Verbindung eine andere Python Bibliothek zu verwenden, als ursprünglich geplant. Da die „PyBluez“ Bibliothek aufgrund von Kompatibilitätsproblemen nicht verwendet werden konnte, wurde die Bluetooth Verbindung mit der Bibliothek „Serial“ initialisiert.

Außerdem wurde eine Ergänzung zum Entwurf vorgenommen. Damit der Benutzer mit den gemessenen Werten der Beleuchtung etwas anfangen kann, wurde eine kleine Interpretation im Ausgabebereich der GUI hinzugefügt. Diese trifft Aussagen darüber, ob die Beleuchtung ausreichend hell oder zu dunkel ist.

5.2.3 Was könnte beim nächsten Projekt anders gemacht werden?

Da die Forschungsfrage wie geplant beantwortet werden konnte, wird ein zukünftiges Projekt ebenfalls auf diese Weise geplant und umgesetzt. Wichtig ist zu beachten, dass der Arbeitsplan ausführlich aufgestellt wird und Alternativen bei möglichen Problemen bietet. Außerdem ist es sinnvoll, Pufferwochen einzuplanen, damit immer genügend Zeit vorhanden ist, um das Projekt zum gewünschten Abschluss zu bringen. Um den Arbeitsplan auf diese Weise aufstellen zu können, muss im Voraus gesichtet werden, welche Vorarbeiten bereits vorhanden sind und an welchen Stellen mit Problemen gerechnet werden muss.

5.3 Ausblick

Die Forschungsfrage konnte wie geplant beantwortet werden, stellt jedoch nur einen ersten Schritt in die Forschungsthematik dar. Um einen marktreifen Smart Mirror zur

Beleuchtungssteuerung im Home Office zu entwickeln, müssen einige weitere Forschungs- und Entwicklungsschritte durchgeführt werden. Einige zu berücksichtigende Aspekte sind:

- Das Bauen eines echten Smart Mirrors mit Touchbildschirm hinter einer durchlässigen Spiegelfolie.
- Das Ersetzen der RGB-LEDs durch eine echte Lampe, die jedoch immer noch programmierbar ist und über Bluetooth mit der GUI kommunizieren kann.
- Eine ausführlichere Interpretation der gemessenen Beleuchtungsstärken und Farbtemperaturen, um die Usability zu erhöhen.
- Alternativ zum vorherigen Punkt: Eine Regelung, welche die Beleuchtung automatisch anpasst, wenn das Licht zu dunkel oder zu hell oder die Farbtemperatur zu unangenehm ist.

6 Vorführung des Programms

Das Rechnerprogramm wurde am 06. Juni 2023 im Rahmen der Veranstaltung „Ingenieurwissenschaftliche Applikationen“ an der Jade Hochschule in Wilhelmshaven durchgeführt. Neben der Präsentation der Forschungsergebnisse, der Softwareumsetzung und der Reflexion wurde erfolgreich eine Live-Demonstration des Programms durchgeführt.

Erklärung der eigenständigen Anfertigung

Ich, Timon Faß, erkläre hiermit, dass diese Dokumentation des Rechnerprogramms mit dem Titel „Entwicklung eines Smart Mirrors zur Verwendung im Home Office mit Fokus auf einer Beleuchtungssteuerung“ vollständig mein Eigentum ist. Ich bestätige, dass:

- ich die Arbeit selbständig verfasst habe, und dass die benutzten Hilfsmittel vollständig angegeben sind.
- dort wo diese Arbeit auf Arbeiten basiert, die gemeinsam mit anderen durchgeführt wurden, habe ich im Abschnitt „Erklärung zum eigenen Beitrag“ deutlich gemacht, welche Teile von mir und welche von anderen stammen.

Diese Arbeit soll als Teil der Veranstaltung „Ingenieurwissenschaftliche Applikationen“ im SoSe 2023 gewertet werden.

Wilhelmshaven, 27. Juni 2023

Timon Faß: 