



- Projektarbeit -

Sensor Monitor Dashboard

| | | |
|-------|------------------------------|---------|
| Name: | Celine Wendt | 6070678 |
| | Yamen Alhammoud | 6022375 |
| | Zainab Raad Hussein, Hussein | 6045492 |
| | Yazan Alhammoud | 6022520 |

| | |
|--------------|---|
| Studiengang: | Ingenieurinformatik MASTER |
| Modul: | Objektorientierte Programmierung - Vertiefung |

| | |
|---------|----------------------------|
| Dozent: | Professor Dr. Ammar Memari |
|---------|----------------------------|

Eidesstattliche Erklärung

Hiermit versichern wir, dass wir die vorliegende Arbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der entsprechenden Quellen kenntlich gemacht worden.

Bremen, 08.06.2025, Celine Wendt, 5139064

Celine Wendt

Ort, Datum, Vor- und Nachname, Matrikelnummer

Unterschrift

Wilhelmshaven, 08.06.2025, Yamen Alhammoud, 6022375



Ort, Datum, Vor- und Nachname, Matrikelnummer

Unterschrift

Wilhelmshaven, 08.06.2025, Zainab Raad Hussein Hussein, 6045492



Ort, Datum, Vor- und Nachname, Matrikelnummer

Unterschrift

Wilhelmshaven, 08.06.2025, Yazan Alhammoud, 6022520



Ort, Datum, Vor- und Nachname, Matrikelnummer

Unterschrift

Inhaltsverzeichnis

| | |
|---|---|
| 1 Projektbeschreibung | 1 |
| 2 Installationsanleitung | 2 |
| 3 Beschreibung der GUI-Funktionen | 3 |
| 4 Reflexion | 5 |
| 5 Quellenverzeichnis | 6 |

1 Projektbeschreibung

Der Sensor Monitor ist eine interaktive Anwendung, mit der sich Temperaturdaten von Sensoren in Echtzeit überwachen, analysieren und dokumentieren lassen. Entwickelt wurde das Projekt im Rahmen eines Softwareentwicklungsprojekts mit Python und PyQt6. Ziel war es, eine benutzerfreundliche Oberfläche zu schaffen, die nicht nur Messwerte live darstellt, sondern auch Auswertungen ermöglicht und praktische Zusatzfunktionen bietet.

Die Anwendung zeigt die Temperaturdaten in einem Live-Diagramm an und protokolliert alle Werte automatisch in einer Tabelle, sodass der Verlauf jederzeit nachvollzogen werden kann. Neben der reinen Anzeige werden auch statistische Kennzahlen wie Minimum, Maximum und Durchschnitt berechnet und grafisch aufbereitet. Zusätzlich erkennt das System kritische Wertebereiche und reagiert bei Überschreitungen mit visuellen Hinweisen und einer Sprachausgabe als akustische Warnung.

Ein weiteres Highlight ist die integrierte Kamerafunktion. Damit kann man ein Livebild anzeigen lassen, Schnappschüsse machen oder Videoaufnahmen starten – etwa zur zusätzlichen Dokumentation von Umgebungsbedingungen. Auch ein Screenshot der Benutzeroberfläche kann mit einem Klick erstellt werden.

Die Daten lassen sich bei Bedarf als CSV-Datei exportieren oder als PDF-Bericht ausgeben. Darüber hinaus gibt es eine Druckfunktion, um Ergebnisse direkt zu Papier zu bringen. Die Anwendung unterstützt sowohl einen hellen als auch einen dunklen Modus und passt die Abtastrate automatisch an die aktuelle CPU-Auslastung an, um effizient zu arbeiten. So bietet das Sensor Monitor eine umfassende Lösung zur Messwernerfassung und -auswertung mit vielen nützlichen Funktionen, die im Alltag eines technischen oder wissenschaftlichen Projekts echten Mehrwert schaffen.

2 Installationsanleitung

Damit der Sensor Monitor lokal ausgeführt werden kann, sind einige Voraussetzungen notwendig. Zunächst muss Python in der Version 3.10 oder höher auf dem System installiert sein. Die Anwendung basiert vollständig auf Python und nutzt keine externen ausführbaren Programme, wodurch sie plattformunabhängig und leicht portierbar ist.

Nach der Installation von Python müssen die benötigten Bibliotheken über pip installiert werden. Diese beinhalten unter anderem PyQt6 für die Benutzeroberfläche, PyQtGraph für die grafische Darstellung der Daten, OpenCV für die Kamerafunktionen, pytsx3 für die Sprachausgabe und psutil zur Ermittlung der Systemauslastung. Die Installation erfolgt über folgenden Befehl im Terminal oder in der Eingabeaufforderung:

- `pip install pyqt6 pyqtgraph opencv-python pytsx3 psutil`

Nach erfolgreicher Installation kann die Anwendung direkt durch Ausführen der Python-Datei gestartet werden. Es sind keine weiteren Konfigurationsschritte notwendig – alle Einstellungen werden beim ersten Start automatisch im lokalen Benutzerprofil gespeichert.

3 Beschreibung der GUI-Funktionen

Die Benutzeroberfläche des Sensor Monitor Dashboards ist übersichtlich gestaltet und lässt sich gut und einfach bedienen. Im oberen Bereich, dem sogenannten Header, können Nutzer:innen ihren Namen aus einer Liste auswählen oder selbst eingeben und per Klick übernehmen. Die aktuelle Uhrzeit und das Datum werden in Echtzeit angezeigt. Über ein Auswahlménü kann direkt zwischen den verschiedenen Tabs der Anwendung gewechselt werden. Weitere Schaltflächen ermöglichen das Speichern eines Screenshots sowie das Umschalten zwischen hellem und dunklem Design (Light-/Dark-Mode), je nach Umgebung oder persönlicher Vorliebe.

Die Anwendung ist in fünf Tabs unterteilt, die verschiedene Schwerpunkte abdecken. Im Live-Plot-Tab werden die aktuellen Temperaturwerte grafisch in Echtzeit dargestellt. Ein individuell einstellbarer Grenzwert dient zur Einordnung der Temperatur. Wichtig dabei: Die Bewertung erfolgt nicht exakt am Grenzwert selbst, sondern innerhalb eines Toleranzbereichs von ± 5 °C. Werte, die mehr als 5 Grad unter dem Grenzwert liegen, gelten als „**normal**“. Liegt ein Wert im Bereich von ± 5 Grad um den Grenzwert, wird er als „**Grenzwert nah**“ eingeordnet. Wird der Grenzwert hingegen um mehr als 5 Grad überschritten, erfolgt eine **Warnung**, die zusätzlich durch eine akustische Sprachausgabe unterstützt wird. Die Frequenz, mit der die Temperatur überprüft wird, passt sich automatisch an die aktuelle Auslastung des Systems an, um eine möglichst effiziente Ausführung zu gewährleisten. Die Messung selbst lässt sich manuell starten und stoppen.

Im Historie-Tab werden alle bisher erfassten Messwerte in einer Tabelle gespeichert, inklusive Zeitstempel, Sensorbezeichnung, Messwert und Status. So lassen sich vergangene Werte einfach nachvollziehen.

Der Statistik-Tab bietet eine kompakte Übersicht der erfassten Daten in Form eines Balkendiagramms. Neben dem Minimum, Maximum und Durchschnitt werden auch Zähler für die Statuskategorien „Normal“, „Grenzwert nah“ und „Warnung“ angezeigt, um ein besseres Gefühl für die Datenlage zu bekommen.

Im Analyse-Tab wird eine Trendlinie auf Basis der letzten Messwerte angezeigt. Diese erlaubt eine Einschätzung darüber, ob die Temperatur über Zeit tendenziell steigt oder fällt – eine wertvolle Ergänzung zur punktuellen Messung.

Im Kamera-Tab wird das Livebild einer angeschlossenen Kamera angezeigt. Nutzer:innen können direkt Schnappschüsse speichern oder Videoaufnahmen starten. Diese Funktion ist besonders wichtig, um den tatsächlichen Zustand und die Umgebung des Sensors mit aufzuzeichnen – etwa zur späteren Auswertung, für Berichte oder zur Protokollierung von Anomalien. Die Kamerafunktion trägt somit entscheidend zur Nachvollziehbarkeit und Dokumentation der Messungen bei. Der aktuell gewählte Benutzername wird dabei ebenfalls angezeigt, um eine eindeutige Zuordnung zu gewährleisten.

Zusätzlich zur Hauptansicht steht ein klassisches Menü zur Verfügung, das weitere wichtige Funktionen bietet. Über das Datei-Menü lassen sich gespeicherte Daten direkt als CSV-Datei exportieren oder für die Dokumentation drucken. Dadurch können die erfassten Messwerte beispielsweise in Berichten oder zur Archivierung weiterverwendet werden. Ein Menüpunkt zum Beenden der Anwendung ist ebenfalls vorhanden. Die Menüeinträge sind während einer laufenden Messung deaktiviert, um unbeabsichtigte Unterbrechungen oder Datenverluste zu vermeiden. Sobald die Messung gestoppt wird, stehen sie wieder zur Verfügung.

Insgesamt kombiniert das Sensor Monitor Dashboard die Live-Visualisierung, Analyse und Dokumentation von Sensordaten mit nützlichen Zusatzfunktionen wie Kamera, Export und Sprachausgabe – und das in einer einzigen, benutzerfreundlichen Oberfläche.

4 Reflexion

Die Umsetzung des Projekts erfolgte im Team und wurde von Anfang an gemeinsam geplant und besprochen. Wir haben alle Schritte, vom ersten Entwurf über die Umsetzung bis hin zum Testen, als Gruppe abgestimmt, wodurch jeder jederzeit einen Überblick über das Gesamtprojekt hatte. Gleichzeitig hat aber auch jede Person eine konkrete Aufgabe übernommen, um die Arbeit effizient zu strukturieren und Verantwortung zu verteilen. In der folgenden Abbildung 1 sieht man die Verantwortung der einzelnen Projektmitglieder:

| Name | Beiträge |
|--------------------------------|--|
| Yamen Alhammoud | Kamera- und Speicherfunktion <ul style="list-style-type: none">• Nutzung von OpenCV zur Anzeige von Live-Bildern• Speichern von Bildern und Videos in Ordnern• Integration der Kamerabilder in die GUI• CSV |
| Yazan Alhammoud | Diagramme und Statistik-Update <ul style="list-style-type: none">• Datenvisualisierung• Trendanalyse• Statistische Auswertung von Sensorwerten• Balkendiagramm |
| Celine Wendt | Bedienelemente und Menüführung <ul style="list-style-type: none">• Start- und Stopp-Funktionen• Steuerung der Sensor Datenübertragung• Allgemeine Menüstruktur• Tabs |
| Zainab Raad Hussein Hussein | Benutzerinteraktion und Anzeige <ul style="list-style-type: none">• Themenauswahl via ComboBox• Statistische Kennzahlen• Frequenz-Slider und Eingabeelemente• Timer |

Abbildung 1: Aufgabenverteilung

5 Quellenverzeichnis

Teile der Projektarbeit, insbesondere bei der Codeentwicklung, wurden mit Hilfe von ChatGPT (ein KI-gestützter Sprachassistent von OpenAI) erstellt und überarbeitet. Die finale Umsetzung, Anpassung und Integration erfolgte eigenständig durch das Projektteam.