

FH Aachen

Fachbereich Maschinenbau und Mechatronik

Modul

Datenmanagement und Leittechnik, 81307

Projekt

Betriebszustandsüberwachung einer Waschmaschine

vorgelegt von: Faruk Berk, Matrikelnummer: 3612875
Hakan Önder, Matrikelnummer: 3160283

Professor: Prof. Dr.-Ing. Stephan Kallweit

Abgabedatum: 29.02.2024

Inhaltsverzeichnis

1	Einleitung.....	- 1 -
1.1	Aufgabenstellung.....	- 1 -
1.2	Ziel des Projekts.....	- 1 -
2	Genutzte Hardware und Software	- 2 -
2.1	Hardware.....	- 2 -
2.2	Software	- 2 -
3	Ablauf.....	- 4 -
3.1	Datenerfassung.....	- 4 -
3.2	Datenübertragung	- 4 -
3.3	Klassifikation	- 4 -
3.4	Visualisierung.....	- 4 -
4	Fazit.....	- 5 -
5	Anhang.....	I

Abbildungsverzeichnis

Abbildung 1: ESP8266.....	- 2 -
Abbildung 2: MPU6050.....	- 2 -
Abbildung 3: Thonny.....	- 2 -
Abbildung 4: Jupyter Notebook.....	- 3 -
Abbildung 5: MongoDB Atlas	- 3 -
Abbildung 6: MongoDB Compass.....	- 3 -
Abbildung 7: Durchlauf aller Betriebszustände	I
Abbildung 8: Zustand Aus.....	II
Abbildung 9: Zustand Waschen, Schleudern und Abpumpen	II
Abbildung 10: Zustand Schleudern.....	III
Abbildung 11: Zustand Abpumpen	III

1 Einleitung

1.1 Aufgabenstellung

Im Rahmen dieses Projektes wird ein Mikrocontroller genutzt, um Daten zu sammeln und die Betriebszustände einer Waschmaschine zu ermitteln. Hierbei wird das bereits im Praktikum eingesetzte IoT-Kit verwendet, das einen integrierten Beschleunigungssensor (MPU6050) enthält. Mit Hilfe von verschiedenen Entwicklungsumgebungen sollen die Betriebszustände erfasst, gespeichert und klassifiziert werden, welche in folgende Aufgabenteile untergliedert sind:

1. Datenerfassung auf dem IoT-Device (MicroPython)
2. Übertragung der Daten mit geeigneten Protokollen (UDP, TCP/IP)
3. Daten sinnvoll lokal und „cloud-orientiert“ unter Benutzung von Datenbanksystemen (noSQL MongoDB) speichern
4. Grafische Benutzeroberfläche mit Qt, bei der die Betriebszustände ersichtlich sind (Klassifikation)

1.2 Ziel des Projekts

Dieses Projekt demonstriert die grundlegende Durchführung der Überwachung verschiedener Betriebszustände mithilfe der IoT-Technologie und deren praktischen Anwendung im Alltag. Es unterstreicht die Bedeutung der Integration von Sensoren in das IoT für die Überwachung von Anlagen. Darüber hinaus zeigt es, wie maschinelles Lernen genutzt werden kann, um einen Computer mit einer großen Datenmenge zu trainieren und datengestützte Empfehlungen und Entscheidungen basierend auf den Eingabedaten zu treffen.

2 Genutzte Hardware und Software

2.1 Hardware

1. Der ESP8266 ist ein beliebter Microcontroller für IoT-Projekte, der drahtlose Kommunikation ermöglicht und einfach zu programmieren ist. Er bietet vielseitige Möglichkeiten zur Anbindung verschiedener Sensoren.



Abbildung 1: ESP8266

2. Der MPU6050 ist ein Beschleunigungssensor für die Erfassung von Vibrationsdaten. Er kann präzise Bewegungen in drei Dimensionen messen und wird häufig genutzt, um Vibrationen in Maschinen, Strukturen oder Fahrzeugen zu überwachen. Somit können potenzielle Probleme frühzeitig erkannt werden.

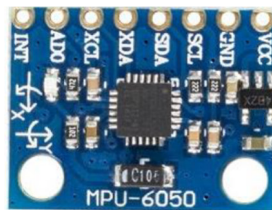


Abbildung 2: MPU6050

2.2 Software

3. Thonny ist eine Entwicklungsumgebung für Microcontroller wie den ESP8266. Es bietet eine benutzerfreundliche Oberfläche und eine einfache Integration von Bibliotheken und Erweiterungen, was es ideal für die Entwicklung von IoT-Anwendungen macht.

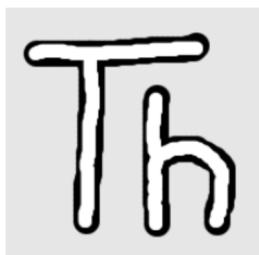


Abbildung 3: Thonny

4. Im Jupyter Notebook ist es möglich den Microcontroller für die Entwicklung von IoT-Anwendungen zu verwenden. Jupyter bietet eine interaktive Plattform zum Schreiben, Ausführen und Testen von Code in verschiedenen Programmiersprachen wie Python.



Abbildung 4: Jupyter Notebook

5. MongoDB Atlas ist eine cloudbasierte Plattform, die es vereinfacht, Datenbanken zu erstellen und zu verwalten. Mit MongoDB Atlas ist es möglich schnell hochverfügbare und skalierbare Datenbankinstanzen in der Cloud bereitstellen, ohne sich um die zugrunde liegende Infrastruktur kümmern zu müssen.



Abbildung 5: MongoDB Atlas

6. MongoDB Compass ist eine grafische Benutzeroberfläche für MongoDB-Datenbanken. Es ermöglicht Benutzern, Datenbanken einfach zu durchsuchen, zu visualisieren und zu verwalten, ohne umfangreiche Kenntnisse der MongoDB-Query-Sprache zu benötigen.



Abbildung 6: MongoDB Compass

3 Ablauf

3.1 Datenerfassung

Zunächst wurde der Microcontroller über Thonny so programmiert, dass er Daten vom Beschleunigungssensor einlesen kann, diese über UDP, als UDP-Client, nach Jupyter Notebook senden und anschließend die gesendeten Daten nochmals auf dem Microcontroller speichert. Der dazugehörige Programmcode ist in der Datei „Erfassung_der_Zustände.py“ hinterlegt.

3.2 Datenübertragung

Anschließend wurden die Daten von Jupyter Notebook als UDP-Server empfangen, verarbeitet und in die Datenbank MongoDB hochgeladen und gespeichert. Dies wurde für alle gemessenen Betriebszustände gemacht. Die Datenbank wurde über MongoDB Atlas erstellt und konfiguriert. Mit MongoDB Compass sind die Daten ersichtlich und weiter bearbeitbar.

3.3 Klassifikation

Nachdem alle Zustände erfasst sind und in der Datenbank abgespeichert wurden, konnten die Daten genutzt werden, um das Neuronale Netzwerk zu trainieren, damit anhand dessen bestimmt werden kann in welchem Zustand sich aktuell die Waschmaschine befindet. Dabei sind die Iterationen des Trainings anpassbar, aber sollte so lange laufen bis eine Genauigkeit von ca. 90% erreicht wird. Dadurch sollte gewährleistet sein, dass die Daten richtig erfasst werden.

3.4 Visualisierung

Nach dem Training folgt der Test mit einem zufälligen Zustand: Als Beispiel könnte der Zustand „Schleudern“ genommen und betrachtet werden, ob es dementsprechend als „Schleudern“ klassifiziert wird. Dieses Ergebnis wird anschließend visuell ausgegeben.

4 Fazit

Insgesamt hat sich dieses Projekt als äußerst wertvoll erwiesen, da es uns ermöglicht hat, nicht nur theoretische Kenntnisse zu erlangen, sondern auch praktische Erfahrungen zu sammeln und anzuwenden. Die Anwendung des Verfahrens an einer Waschmaschine zeigt, dass diese Methode ebenso auf industrielle Maschinen oder andere technische Anlagen übertragbar ist. Überall dort, wo Zustände erfasst werden können, kann dieses Verfahren erfolgreich eingesetzt werden.

Es ist wichtig anzumerken, dass trotz des Erfolgs des Projekts noch Raum für Verbesserungen besteht. Ein Beispiel hierfür ist die Inflexibilität einiger Schritte, wie zum Beispiel beim Empfangen von Daten vom UDP-Client an den UDP-Server, der in einem Jupyter Notebook läuft. Während des Empfangs kann der Server nicht gleichzeitig andere Aufgaben erledigen und somit eine Live-Visualisierung der Daten ausschließt. Dies ist ein Bereich, der weiterentwickelt werden sollte, um die Effizienz und Flexibilität des Systems zu verbessern.

Alles in allem war das Projekt ein wichtiger Schritt, um die Brücke zwischen Theorie und Praxis zu schlagen und ein tieferes Verständnis für die Anwendung von IoT-Technologien im Datenmanagement und Leittechnik zu vermitteln. Es hat gezeigt, dass durch kontinuierliche Weiterentwicklung und Anpassung auch komplexe Herausforderungen bewältigt werden können.

5 Anhang



Abbildung 7: Durchlauf aller Betriebszustände



Abbildung 8: Zustand Aus



Abbildung 9: Zustand Waschen, Schleudern und Abpumpen



Abbildung 10: Zustand Schleudern



Abbildung 11: Zustand Abpumpen