

Projektdokumentation Optikerkette SchönesGlas

Projektarbeit

Ausbildung Fachinformatik

Fachrichtung Anwendungsentwicklung

Elektronikschule Tettnang

von

Malte Blumenthal, Benjamin Schick

Abgabedatum:	24.04.2023
Bearbeitungszeitraum:	24.03.2023 - 24.04.2023
Klasse:	EFI222

Inhaltsverzeichnis

1	Projektplanung	1
1.1	Ausgangssituation	1
1.1.1	Projektziele	1
1.1.2	Teilaufgaben	3
1.1.3	Projektumfeld, Projektschnittstellen	4
1.2	Ressourcen- und Ablaufplanung	4
1.2.1	Terminplanung	4
1.2.2	Personalplanung	5
1.2.3	Sachmittelplanung	6
1.2.4	Kostenplanung	7
1.2.5	Qualitätsplanung	8
2	Durchführung- und Auftragsbearbeitung	9
2.1	ESP8266	9
2.1.1	Anschließen der Sensorik	9
2.1.2	Sensordaten auslesen	9
2.1.3	WLAN Verbindung	11
2.1.4	Verbinden zu und publishen auf MQTT Broker	11
2.2	Raspberry Pi	11
2.2.1	Installation benötigter Software	11
2.2.2	Einrichtung der Datenbank	11
2.2.3	Programmierung des NodeJS-Servers	11
2.2.4	Programierung der Website zur Datenvisualisierung	11

3	Projektergebnisse	12
3.1	Projektergebnisse	12
3.1.1	Abnahme	12
3.1.2	Soll-Ist-Vergleich	13
3.1.3	Bewertung (Fazit, Ausblick)	13

1 Projektplanung

1.1 Ausgangssituation

1.1.1 Projektziele

1. Mobiles ESP8622

Als Station für die Aufzeichnung der Daten soll ein Microcontroller verwendet werden. Vorgabe der Stadt Tettnang war ein ESP 8266, welches über eine mobile Stromversorgung betrieben werden soll. Als Sensoren für das Aufzeichnen der Daten sollen ein DHT22 und ein BMP 180 verwendet werden.

2. DHT22

Der DHT22 Sensor wird für die Aufzeichnung von Luftfeuchtigkeit und Temperatur verwendet.

3. BMP180

Für die Aufzeichnung der Höhe über Normal Null und Luftdruck wird ein BMP180 Sensor verwendet.

4. Raspberry Pi 4

Der Webserver braucht Hardware um betrieben zu werden. Da das Verkehrsvolumen auf dem Server als relativ gering eingeschätzt wird, sollte ein Raspberry Pi 4 als Hardwareinfrastruktur ausreichen.

5. Webserver

Für das Bereitstellen von Visualisierungen der Daten und Aufzeichnung der Daten in einem Datenbank System wird ein Webserver als Interface benötigt. Dafür wird in diesem Projekt NodeJS verwendet.

6. Frontend

Für die Darstellung der Daten soll eine Website zur Verfügung bereitgestellt werden. Diese soll vorerst eine einzelne Seite sein, die nur eine Tabelle die die Höhe über Normal Null auf einer Horizontalen Zeitachse darstellt.

7. Backend

Das Backend soll das Topic im MQTT-Broker abonnieren, und dann alle Updates in die Datenbank speichern. Außerdem soll das Backend die Daten aus der Datenbank auslesen und für die Frontendimplementierung bereitstellen.

8. Datenbankserver

Der Datenbankserver soll auch auf dem Raspberry Pi laufen und die Daten, die von den Sensoren ausgeliefert wurden, permanent abspeichern. Als Datenbanksystem soll MariaDB, eine SQL -Implementierung, verwendet werden.

9. Kommunikation mit MQTT-Broker

Die Daten müssen zwischen den verschiedenen Geräten ausgetauscht werden. Für die Kommunikation wird im Vorfeld ein öffentlicher MQTT-Broker verwendet. In diesem Projekt wird voraussichtlich mit HiveMQ gearbeitet. Optimal wäre das Hosten eines privaten MQTT-Brokers, aber dies liegt außerhalb des Projekterahmens.

10. Webserver und Broker

Der Webserver soll sich als Abonnent am entsprechenden Topic des MQTT-Brokers anmelden, um die regelmäßigen Updates des Topics und somit neue Daten zu erhalten.

1.1.2 Teilaufgaben

1. Festlegung Projektziele
2. Festlegung Teilaufgaben
3. Beschreibung Projektumfeld
4. Beschreibung Projektschnittstellen
5. Personalplanung
6. Sachmittelplanung
7. Terminplanung
8. Ablaufplan
9. Kostenplanung
10. Definition Technologiestack
11. Erstellen Netzwerkplan
12. Einrichten Raspberry Pi
13. Programmierung ESP
14. Programmierung Backend
15. Programmierung Frontend
16. Testen der Infrastruktur mittels Testfahrt

17. Dokumentation des Pilotprojekts

1.1.3 Projektumfeld, Projektschnittstellen

Tabelle 1.1: Projektumfeld

Auftraggeber	Elektronikschule Tettnang im Auftrag der Stadt Tettnang
Auftragnehmer	Schüler der Klassen EFI222
Räumliches Umfeld	Klassenzimmer an der Elektronikschule Tettnang
Ansprechpartner	Herr Rauschmaier Frau Wattenbach Klassenkameraden
Einstieg	Anfrage der Stadt Tettnang
Ausstieg	Übergabe des Prototypen

1.2 Ressourcen- und Ablaufplanung

1.2.1 Terminplanung

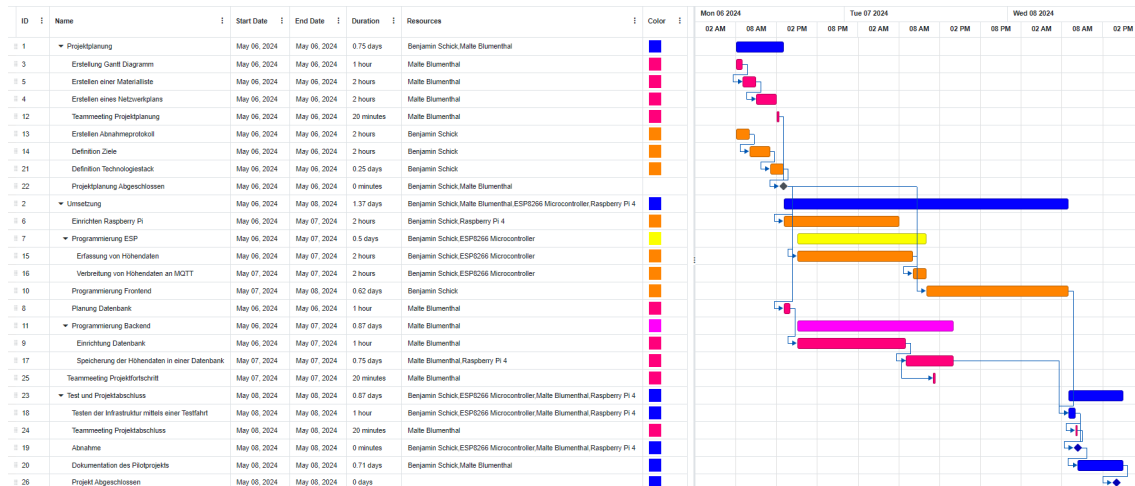


Abbildung 1.1: Terminplanung Tabellarische Ansicht

In dieser Zeitplanung sind keine Puffer berücksichtigt worden, da die gesamte Projektzeit sehr knapp bemessen ist.

1 Projektplanung

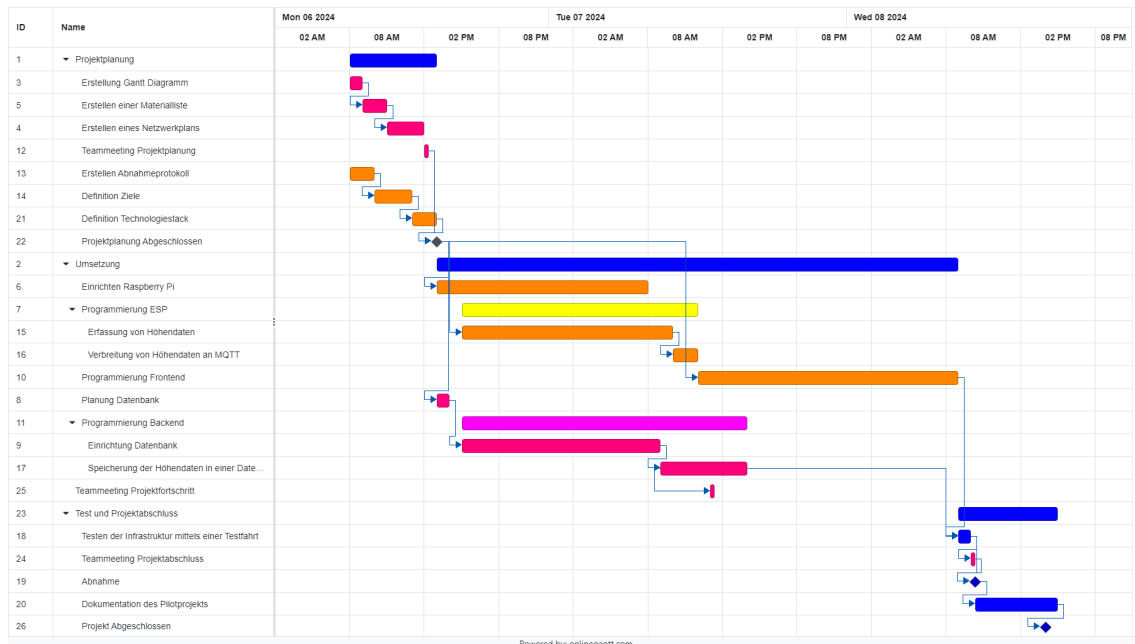


Abbildung 1.2: Terminplanung Gantt-Diagramm

1.2.2 Personalplanung

In dieser Zeitplanung sind keine Puffer berücksichtigt worden, da die gesamte Projektzeit sehr knapp bemessen ist.

1.2.3 Sachmittelplanung

Tabelle 1.2: Sachmittelplanung Hardware

Hardware	Menge	Beschreibung	Verfügbarkeit
Entwickler PC	2x	Hardware benötigt für die Entwicklung der Software und Dokumentation des Projekts	Ist Vorhanden
MQTT-Broker (öffentlich)	1x	MQTT-Server der öffentlich zugänglich ist	Muss beschafft werden
Raspberry Pi 4	1x	Hardware, welche die Rolle des Servers übernimmt. Ist Host des Webservers und der Datenbank	Muss beschafft werden
ESP8266	1x	Microcontroller, welcher die Messungen durchführen soll	Muss beschafft werden
DHT22	1x	Modul für ESP, Zeichnet Temperatur und Luftfeuchtigkeit auf	Muss beschafft werden
BMP180	1x	Modul für ESP, Zeichnet Höhe über NN und Luftdruck auf	Muss beschafft werden

Tabelle 1.3: Sachmittelplanung Software

Software	Beschreibung	Verfügbarkeit
Visual Studio Code	Entwicklungsumgebung für alle Softwareanwendungen, die benötigt werden	Freeware
MariaDB	Datenbanksystem, das für das Speichern der Messdaten verwendet wird	Freeware
MySQL	Datenbanksprache auf der MariaDB aufbaut	Freeware
PlatformIO-Tools	Tools und Bibliotheken für das Programmieren verschiedener Microcontroller	Freeware
NodeJS	Webserver für das Hosten von Webanwendungen	Freeware
UbuntuServer 22.04	Betriebssystem für den Server	Freeware
C++	Programmiersprache für ESP8266	Freeware
MQTT	IoT Netzwerkprotokoll	Freeware

1.2.4 Kostenplanung

Tabelle 1.4: text

Artikel	Menge	Preis
ESP8622	1x	8,00 €
Netzteil ESP8622	1x	5,00 €
Raspberry Pi 4	1x	50,00 €
Netzteil Raspberry Pi 4	1x	5,00 €
DHT22 Sensor	1x	10,00 €
BMP180 Sensor	1x	5,00 €
Mobile Stromquelle	1x	15,00 €
		98,00 €
Lohn Entwickler	48x	6,00 €
Betriebskosten PC	2x	10,00 €
		308,00 €
Gesamtpreis		416,00 €

Somit belaufen sich die geplanten gesamten Projektkosten auf **416,00€**.

1.2.5 Qualitätsplanung

- Qualitätssicherung des Codes mithilfe von GitHub und Versionierung
- Regelmäßiger Self-Test der Hardware
- Fehlermanagement im Code
- SCRUM

2 Durchführung- und Auftragsbearbeitung

2.1 ESP8266

2.1.1 Anschließen der Sensorik

Für die Aufgabe wird grundsätzlich nur ein BMP benötigt. Da wir uns jedoch entschieden haben, die Messdaten um Temperatur und Luftfeuchtigkeit zu erweitern, wird ein entsprechender zusätzlicher DHT mit eingebunden.

Da es die Sensoren für den ESP bereits auf entsprechenden Daughterboards angebracht gibt, war es nur noch nötig diese aufzustecken:

2.1.2 Sensordaten auslesen

Weil MicroPython auf dem ESP in Kombination mit einem MQTT-Broker schon zuvor im Unterricht Probleme bereitet hat, haben wir uns dazu entschieden, den ESP mit C++ zu programmieren.

Für sowohl den DHT als auch den BMP wurde die „Adafruit_Sensor“ Bibliothek verwendet.

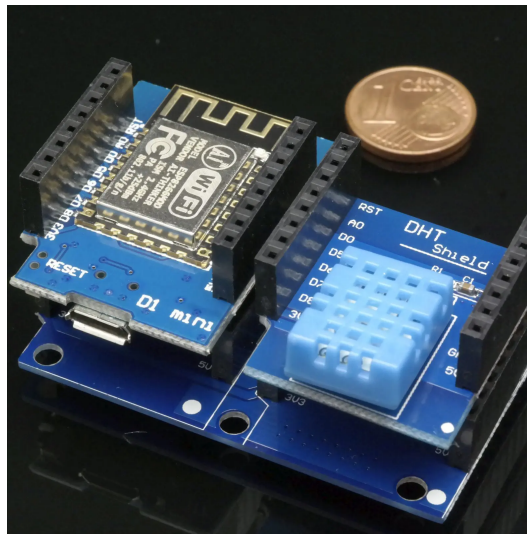


Abbildung 2.1: DHT an ESP8266 angesteckt

Auslesen DHT

Beim DHT war es noch zusätzlich notwendig, die

Auslesen BMP

Erzeugung Timestamp

2.1.3 WLAN Verbindung

2.1.4 Verbinden zu und publishen auf MQTT Broker

2.2 Raspberry Pi

2.2.1 Installation benötigter Software

2.2.2 Einrichtung der Datenbank

2.2.3 Programmierung des NodeJS-Servers

2.2.4 Programmierung der Website zur Datenvisualisierung

3 Projektergebnisse

3.1 Projektergebnisse

3.1.1 Abnahme

Mit der Abnahme des Projektes sind die Ziele des Kunden für den Rahmen des Projektes vollständig erfüllt worden. Es bestehen trotzdem noch Möglichkeiten das System zu erweitern. Ein Punkt wäre das Layout der Website. Die Website beinhaltet zwar das Impressum, aber dies ist aber auf der „About“- oder „Über Uns“-Seite doch sehr versteckt. Man könnte das Impressum beispielsweise in einen Footer verschieben den man auf allen Seiten sehen kann. Ein weiterer Punkt wäre ein Shopsystem. Dafür müssten die nötigen Anpassungen auf den relevanten Seiten getroffen werden, aber auch die Datenbank müsste für diesen Schritt sehr erweitert werden. Noch dazu kommt, dass das Usermanagement momentan sehr rudimentär implementiert ist, was in einer zukünftigen Version etwa aussehen könnte wie in Abbildung 3.1.

Der dritte Punkt wäre die Verschlüsselung der Kundendaten. Aktuell wird lediglich das Passwort mit der Methode md5 im Backend, also von Nodejs, gehashed. Das Ziel war ursprünglich alle vom Nutzer eingegebenen Daten im Frontend, also Clientside, zu hashen und zu salten, damit ein potentieller Man-in-the-middle-Angriff sehr viel weniger gefährlich für die Nutzer des Systems und die Optikerkette sind. Man würde durch diese Maßnahme außerdem die Privatsphäre der Nutzer schützen, da keiner der Administratoren die Daten der Nutzer dann willkürlich aus der Datenbank auslesen kann.



Abbildung 3.1: Benutzerspeicherung in der Datenbank

3.1.2 Soll-Ist-Vergleich

3.1.3 Bewertung (Fazit, Ausblick)

Erklärung

Wir versichern hiermit, dass wir unsere Dokumentation zur Projektarbeit Projektarbeit mit dem Thema:

Projektdokumentation Optikerkette SchönesGlas

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben. Wir versichern zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Tettnang, den 13. Mai 2024

Malte Blumenthal, Benjamin Schick