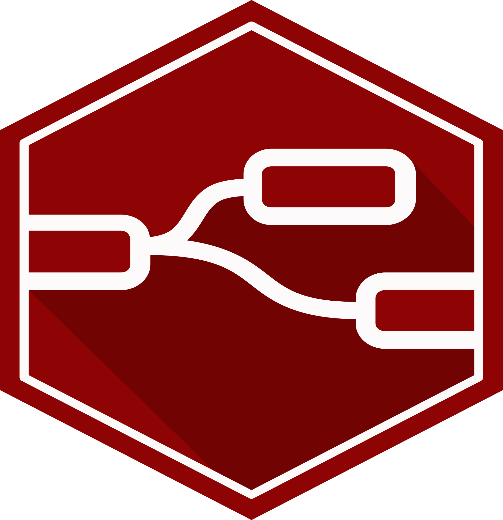


|  |
| --- |
| Labortemperatur und Luftfeuchtigkeit kundengerecht darstellen  -  Mit MQTT und Node-RED |

|  |
| --- |
|  |
| **1. Dezember 2024**  Josef-Durler-Schule Rastatt  BFKO E2FI/FS – 2024  Lehrer: Mike Stefanski  Verfasst von: Daniel Kouvaris  Ralph Heinkel IT-Beratung |



Inhaltsverzeichnis

[Abbildungsverzeichnis 4](#_Toc183207514)

[Tabellenverzeichnis 4](#_Toc183207515)

[Abkürzungen 4](#_Toc183207516)

[1 Ausgangssituation 5](#_Toc183207517)

[1.1 Projektumfeld 5](#_Toc183207518)

[1.2 Projektziele 5](#_Toc183207519)

[1.3 Teilaufgaben 5](#_Toc183207520)

[1.4 Kundenwünsche 6](#_Toc183207521)

[2 Ressourcen- und Ablaufplanung 6](#_Toc183207522)

[2.1 Personalmittel 6](#_Toc183207523)

[2.2 Sachmittel 6](#_Toc183207524)

[2.2.1 Arbeitsplatz 6](#_Toc183207525)

[2.2.2 Technische Sachmittel 6](#_Toc183207526)

[2.3 Terminplanung 7](#_Toc183207527)

[2.4 Kostenplanung 7](#_Toc183207528)

[2.5 Ablaufplan 8](#_Toc183207529)

[3 Durchführung und Auftragsbearbeitung 9](#_Toc183207530)

[3.1 Prozessschritte 9](#_Toc183207531)

[3.1.1 Analyse 9](#_Toc183207532)

[3.1.2 Entwurf 10](#_Toc183207533)

[3.1.3 Implementierung 11](#_Toc183207534)

[3.1.4 Abnahme und Einführung 12](#_Toc183207535)

[3.1.5 Dokumentation 12](#_Toc183207536)

[3.2 Vorgehensweise 12](#_Toc183207537)

[3.3 Qualitätssicherung 12](#_Toc183207538)

[3.4 Abweichung 12](#_Toc183207539)

[3.5 Anpassung 12](#_Toc183207540)

[3.6 Entscheidung 12](#_Toc183207541)

[4 Projektergebnisse 13](#_Toc183207542)

[4.1 Soll – Ist- Vergleich 13](#_Toc183207543)

[4.2 Abweichungen und Anpassungen 13](#_Toc183207544)

[4.3 Qualitätssicherung 14](#_Toc183207545)

[5 Dokumentation 14](#_Toc183207546)

[5.1 Vorbereitung des Raspberry Pis: 14](#_Toc183207547)

[5.2 Vorbereitung von Node-RED 15](#_Toc183207548)

[Literaturverzeichnis 16](#_Toc183207549)

[Anhang 17](#_Toc183207550)

[A.1 Detaillierte Zeitplanung 17](#_Toc183207551)

# Abbildungsverzeichnis

[Abbildung 1: Gantt Diagramm – Ablaufplan 9](#_Toc183207346)

[Abbildung 2: Regelkreis der Implementierung 9](#_Toc183207347)

[Abbildung 3: Möglicher Anschluss des DHT11 Sensors an den Raspberry Pi 11](file:///C:\Users\danie\Documents\Geschäftlich\Umschlung\Projekte\2024-11-22-Stefanski_Projektkompetenz.docx#_Toc183207348)

[Abbildung 4: Skalen der Nutzeroberfläche 12](#_Toc183207349)

# Tabellenverzeichnis

Tabelle 1: Anschaffungskosten Messeinrichtung 7

Tabelle 2: Preisübersicht Software Komponenten 7

Tabelle 3: Übersicht Personalkosten 8

Tabelle 4: Gesamtkostenkalkulation 8

Tabelle 5: Grobe Zeitplanung 8

Tabelle 6: Übersicht Personal- & Maschinenkosten 10

Tabelle 7: Anschlusstabelle Raspberry Pi - DHT11 10

Tabelle 8: Entscheidungsmatrix 13

# Abkürzungen

CSV Comma-separated values

EPL Eclipse Public License

GPIO General Purpose Input/Output

GPL 3 GNU General Public License 3

LIMS Laboratory Information Management System

MQTT Message Queuing Telemetry Transport

NFTC No-Fee Terms and Conditions License

PC Personal Computer

# Ausgangssituation

Die folgende Projektdokumentation schildert den Ablauf eines Kundenprojekts, welches der Autor in seinem dritten Lehrjahr, im Ausbildungsgang Fachinformatiker für Anwendungsentwicklung, durchgeführt hat.

## Projektumfeld

Die Ralph Heikel IT-Beratung ist ein kleines mittelständisches Unternehmen in Gaggenau, dass maßgeschneiderte Softwarelösungen nach Kundenwunsch erstellt. Gegründet wurde das Unternehmen 1996 durch Ralph Heinkel nach dessen erfolgreich abgeschlossenem Studium der Medizin-Informatik an der Universität Heidelberg. Im Jahr 2023 vergrößerte sich das Unternehmen durch die Anstellung des ersten Auszubildenden, dem Autor.

Zu den Kunden gehört unter anderem ein multinationaler Pharmazie Konzern. Dies ist der primäre Einsatzort des Auszubildenden. Zu seinen Aufgaben gehört es, das Laboratory Information Management System (LIMS) weiterzuentwickeln und nach Kundenwünschen anzupassen. Es handelt sich dabei um ein in Python3 geschriebenes Framework mit Jinjar2! Verknüpfung um das Nutzerinterface in Webbrowsern darzustellen.

Die Josef-Durler-Schule Rastatt ist eine innovative berufliche Schule, die unter ihrem Dach viele Schularten und Berufsfelder vereint. Der Autor besucht hier die Berufsschule im Rahmen einer auf zwei Jahre verkürzten betrieblichen Einzelumschulung und befindet sich im Abschlussjahr. Betreut wurde das Projekt im Rahmen des BFKO Unterrichts durch den Fachlehrer Mike Stefanski.

## Projektziele

Ziel des Projektes ist die Temperatur und Luftfeuchtigkeitswerte in einem Labor via MQTT (Message Queuing Telemetry Transport) zu veröffentlichen und zu abonnieren, so dass die Messwerte von überall im Netzwerk auf einem PC (Personal Computer) oder einem mobilen Datenendgerät betrachtet werden können. Die Darstellung der Messwerte soll sowohl zweckmäßig als auch graphisch ansprechend gestaltet sein. Ein speichern der Messwerte soll automatisiert erfolgen. Erhoben werden sollen die Messwerte durch einen DHT11 Temperatur- und Feuchtigkeitssensor der an die GPIO (General Purpose Input/Output) Pins eines Raspberry Pi 4 Modell B angeschlossen ist.

## Teilaufgaben

Das Projektziel lässt sich in drei Teilaufgaben unterteilen.

1. Projektplanung

Die Projektplanung erfolgte im BFKO Unterricht und umfasste das Vertraut machen mit dem Projektziel, der zur Verfügung gestellten Hardware (Raspberry Pi und DHT11 Sensor) sowie der verschiedenen Software Optionen (Java 17, Python 3.12 und Node-RED v4.0.5) und dem Nachrichtenprotokoll MQTT.

Abweichungen von der Projektplanung während der Umsetzung war möglich. So konnte der Autor, wenn er während der Umsetzung eine ihm bisher unbekannte Möglichkeit entdeckte, zur Projektplanung zurückkehren und diese entsprechend der neuen Möglichkeiten anpassen. Damit konnte sichergestellt werden, ein möglichst hochwertiges Ergebnis zu produzieren.

1. Umsetzung

Unterteilt in Hardware und Software beinhaltet diese Teilaufgabe das Verkabeln des DHT11 Sensors mit dem Raspberry Pi sowie die Erstellung einer Softwarelösung um die Sensoren Daten auszulesen, per MQTT zu übermitteln und anschließend Kundengerecht darzustellen.

1. Dokumentation

Nach erfolgreicher Umsetzung ist es notwendig den Prozess zu dokumentieren. Das erfolgte in der vorliegenden Projektdokumentation.

## Kundenwünsche

* Konfiguration der Hardware
* Erstellung der Anwendung
* Darstellung der Daten in ansprechender Form
* Zugang zu den Bereitgestellten Daten per PC oder mobilem Datenendgerät
* Projektdokumentation
* Durchführung der Projektplanung und -umsetzung
* Speicherung der Messdaten in einer CSV-Datei

# Ressourcen- und Ablaufplanung

In der Ressourcen- und Ablaufplanung sollen die benötigten Ressourcen und die notwendige Zeit sowie ein Ablauf der Durchführung geplant werden.

## Personalmittel

Zur Verfügung steht die Arbeitszeit des Auszubildenden. Diese wird zusammen mit den eingesetzten Sachmitteln ([siehe 2.2](#_Sachmittel) Sachmittel) mit 80 € pro Stunde angesetzt. Ein abschließendes Codereview erfolgt durch den Ausbilder. Sein Stundensatz liegt bei 240 €. Materialübergabe und Endabnahme erfolgten durch einen Mitarbeiter des Kunden.

## Sachmittel

Die verwendeten Sachmittel unterteilen sich in Arbeitsplatz und technische Sachmittel. Der Übersichtlichkeit halber wurden die anteiligen Kosten in den Arbeitsstundensatz des Autors und des Ausbilders bereits mit eingerechnet ([siehe 2.1](#_Personalmittel) Personalmittel).

### Arbeitsplatz

Der Arbeitsplatz beinhaltet ein angemietetes Zwei-Personen-Büro, dafür anfallende Heitz-, Strom- sowie Internet-Kosten. Dem Autor stehen hier ein Schreibtisch mit Bürostuhl, Monitor, Tastatur und Maus zur Verfügung. Die Reinigung erfolgt durch die Mitarbeiter des Unternehmens in ihrer Arbeitszeit.

Der Arbeitsplatz des Ausbilders ist gleichermaßen ausgestattet.

### Technische Sachmittel

Dem Ausbilder und dem Autor wurde vom Kunden Laptops mit ausreichender Systemhardware zur Verfügung gestellt. Hierdurch entstehen, abgesehen von Stromkosten [(siehe 2.2.1](#_Arbeitsplatz)) keine weiteren Kosten für den Ausbildungsbetrieb, da auch die Wartung durch den IT-Service des Kunden gewährleistet wird.

Zur Gesamtkalkulation werden deshalb für jede Stunde Laptopeinsatz 10€ angesetzt.

Tätigkeiten für den Kunden erfolgen ausschließlich über die zur Verfügung gestellte Hardware um Datensicherheit und Integrität zu gewährleisten.

Zur Erfassung der Daten wird ein Raspberry Pi 4 Modell B mit 4GM RAM und passender Stromversorgung zusammen mit einem Gehäuse, einem Kühlkörper-Set, einer 32 GB Speicherkarte, einem Steckbrett, einem DHT11 Temperatur- und Feuchtigkeitssensor, einem Cobbler, einem GPIO Flachbandkabel und drei Entwicklerboards – Steckbrückenkabel m/m vom Kunden bereitgestellt.

## Terminplanung

Zur Durchführung des Projektes wurden dem Autor fünf Tage mit je acht Arbeitsstunden für den Aufbau der Hardware und die Erstellung einer Softwarelösung zur Verfügung gestellt. Da sich der Autor nur an einem Tag in der Woche beim Kunden vor Ort befindet, wurde festgelegt, dass er in den Kalenderwochen 41 bis 45 jeweils mittwochs daran arbeiten sollte. Starttermin war der 09.10.2024. Endtermin der 06.11.2024.

Eine Übergabe der Hardwarekomponenten erfolgte am 09.10.2024 durch den Systemadministrator des Kunden. Die Endabnahme und Übergabe des Produktes an den Systemadministrator erfolgten am 06.11.2024. Die Ausarbeitung der Dokumentation ist nach erfolgreicher Einführung geplant.

## Kostenplanung

Die durch den Kunden zur Verfügung gestellte Hardware hatte Anschaffungskosten in Höhe von:

|  |  |  |
| --- | --- | --- |
| Name | Brutto-Preis[[1]](#footnote-1) | Netto-Preis |
| Raspberry Pi 4 Modell B 4GB | 56,90 € | 47,82 € |
| Kühlkörper-Set | 1,90 € | 1,60 € |
| Speicherkarte | 7,85 € | 6,60 € |
| Gehäuse | 7,50 € | 6,30 € |
| USB-C Netzteil | 7,90 € | 6,64 € |
| DHT11 Temperatur- und Feuchtigkeitssensor | 1,95 € | 1,64 € |
| 40pin Jumper / Dupont Kabel Male - Male trennbar | 1,60 € | 1,34 € |
| Cobbler für Raspberry Pi inkl. GPIO Kabel | 3,90 € | 3,28 € |
| Steckbrett | 1,50 € | 1,26 € |
|  |  |  |
| Summe: | 91,00 € | 76,48 € |

Tabelle : Anschaffungskosten Messeinrichtung

Für die zur Option gestellten Software-Optionen Java 17, Python 3.12 und Node-RED v4.0.5 ist kein Lizenzerwerb notwendig. Ebenso stehen die benötigte Bibliothek BCM 2835, der Paketmanager von Node npm und der Eclipse Mosquitto™ Nachrichten Broker als open source im Internet zur Verfügung.

|  |  |  |
| --- | --- | --- |
| Name | Lizenz | Netto-Preis |
| Java 17 | **No-Fee Terms and Conditions License (NFTC)** | 0 € |
| Python 3.12 | **Python License, Version 2** | 0 € |
| Node-RED v4.0.5 | **Apache License 2.0** | 0 € |
| BCM 2835 Bibliothek | **GNU General Public License 3 (GPL 3)** | 0 € |
| npm | **Artistic License 2.0** | 0 € |
| Eclipse Mosquitto™ | **Eclipse Public License (EPL)** | 0 € |
|  |  |  |
| Summe: |  | 0 € |

Tabelle : Preisübersicht Software Komponenten

Zur Kalkulation der Kosten für den Personaleinsatz wurden sowohl die dem Kunden in Rechnung gestellte Arbeitszeit als auch die des Mitarbeiters des Kunden berücksichtigt.

Die deutlich unterschiedlichen Stundensätze kommen durch die verschiedenen Vertragsarten zustande. Zu beachten ist, dass die Abnahme und Einführung eine Veranstaltung des Autors im Beisein des Kundenmitarbeiters ist. Sie ist in der Zeit des Autors bereits unter Umsetzung inkludiert. Somit wird nur die Zeit des Kundenmitarbeiters zusätzlich berechnet.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Person | Preis pro Stunde | Zeit | Netto-Preis |
| Entwicklung und Umsetzung | Autor | 80 € | 40 h | 3.200,00 € |
| Dokumentation | Autor | 80 € | 10 h | 800,00 € |
| NAchkontrolle | Ausbilder | 240 € | 1 h | 240,00 € |
| Abnahme & Einführung | Kundenmitarbeiter | 25 € | 1 h | 25,00 € |
|  |  |  |  |  |
| Summe: |  |  |  | 4.265,00 € |

Tabelle : Übersicht Personalkosten

Die geplanten Gesamtkosten ergaben sich somit aus der Summe der aufgelisteten Teilkosten, wobei sonstige Materialkosten pauschal in die Stundensätze der beteiligten Personen integriert wurden.

|  |  |  |
| --- | --- | --- |
| Art |  | Netto-Preis |
| Anschaffungskosten |  | 76,48 € |
| Softwarekosten |  | 0 € |
| Personalkosten |  | 4.265,00 € |
| sonstige MAterialkosten |  | 0 € |
|  |  |  |
| Gesamtkosten: |  | 4.341,48 € |

Tabelle : Gesamtkostenkalkulation

Um sicher zu stellen, dass ein solcher Aufwand im Verhältnis zum Nutzen steht wurde in der Analysephase eine Nutzwertanalyse durchgeführt ([siehe 3.1.1 Analyse](#_Analyse)).

## Ablaufplan

Für das Projekt standen dem Autor 10 Stunden für die Dokumentation und 40 Stunden für die Umsetzung zur Verfügung. Diese wurden vor Projektbeginn in verschiedene Schritte aufgeteilt, die während der Umsetzung durchlaufen wurden. Eine entsprechende Übersicht bietet Tabelle 1: Grobe Zeitplanung. Hier sind auch die Hauptphasen erkennbar. Eine detaillierte Zeitplanung befindet sich im [Anhang A.1: Detaillierte Zeitplanung](#_Detaillierte_Zeitplanung).

|  |  |  |
| --- | --- | --- |
| Prozessschritte |  | GEplante Zeit |
| Analyse |  | 8 h |
| Entwurf |  | 10 h |
| Implementierung |  | 20 h |
| NAchkontrolle |  | 1 h |
| Abnahme |  | 0,5 h |
| Einführung |  | 0,5 h |
| Dokumentation |  | 10 h |
|  |  |  |
| Gesamt: |  | 50 h |

Tabelle : Grobe Zeitplanung

Die Projektphasen wurden nun zur Verdeutlichung des Ablaufs in ein Gantt Diagramm übertragen.

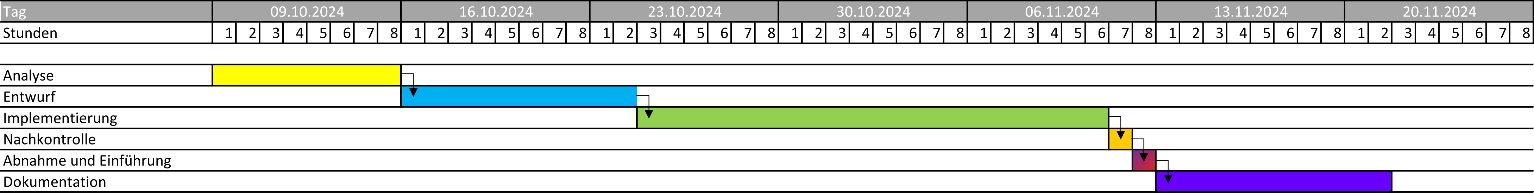


Abbildung : Gantt Diagramm – Ablaufplan

In der Phase Implementierung wurden regelmäßig Zwischenstände erhoben. Da ein Auseinandersetzen mit der Materie für den Autor mit einem tieferen Verständnis der Möglichkeiten einher ging, wurden hiermit sichergestellt, dass den Anforderungen der Kunden auf dem höchstmöglichen Niveau entsprochen wurde und nicht nur mit dem Wissensstand nach der Analyse-Phase. Dies lässt sich als Regelkreis darstellen.

Abbildung : Regelkreis der Implementierung

Auf diese Weise wurden mehrere Lösungsversionen für die Softwarelösung überarbeitet ([siehe 3.1.3 Implementierung](#_Implementierung)).

# Durchführung und Auftragsbearbeitung

## Prozessschritte

Die in 2.5 Ablaufplan definierten Prozessschritte wurden bei der Durchführung befolgt und werden nun im Detail beschrieben.

### Analyse

Um zu ermitteln, ob die veranschlagten Kosten (siehe Tabelle 4) ein zu vertretendes Investment sind wurde ermittelt, wie hoch der Ausfall und die zusätzlichen Kosten sind, die entstehen, wenn die Temperatur oder die Luftfeuchtigkeit den Grenzwert überschreiten.

Herrschen im Labor die falschen Bedingungen, so ist dies nicht sofort ersichtlich. Ein betroffenes Experiment wird also weiterverarbeitet, bis es zur Auswertung kommt. Erst in der Auswertung ist es möglich betroffene Experimente zu erkennen. Im Labor 1 wird der UE145 Experimentator verwendet. Ein Experiment durchläuft ihn im Durchschnitt in drei Stunden. Es können bis zu 20 Experimente gleichzeitig von ihm verarbeitet werden. Die Mitarbeiter entnehmen dann diese Experimente und führen sie nach entsprechender Vorbereitung zur Auswertung in den UA216 Analysator. Dieser kann ebenfalls 20 Experimente gleichzeitig verarbeiten und benötigt dafür eine Stunde. Die Vorbereitungsdauer für die 20 Experimente durch die Labormitarbeiter beträgt 30 Minuten. Innerhalb von weiteren 30 Minuten kann ein Labormitarbeiter erkennen, ob es in der Charge fehlerhafte Experimente gibt. Eine Übersicht der Personen und Maschinenstunden mit entsprechenden Kosten für 20 gleichzeitig durchgeführte Experimente sind in Tabelle 6 aufgelistet.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Aktion | Akteur | Preis pro Stunde | Zeit | Netto-Preis |
| Experimentdurchführung | UE145 | 1750 € | 3 h | 5.250,00 € |
| Vorbereitung | Labormitarbeiter | 25 € | 0,5 h | 12,50 € |
| Analyse | UA216 | 2500 € | 1 h | 2.500,00 € |
| Auswertung | Labormitarbeiter | 25 € | 0,5 h | 12,50 € |
|  |  |  |  |  |
| Summe: |  |  |  | 7.775,00 € |

Tabelle : Übersicht Personal- & Maschinenkosten

Die Materialkosten pro Experiment werden auf 8,25 € veranschlagt. In einem Monat werden durchschnittlich 2400 Experimente durchgeführt. Die Quote für die durch das Projekt vermeidbaren Experimente liegt bei 0,01%.

95,28 €

Formel : Kosten pro Monat durch vermeidbare fehlerhafte Experimente

Mit 95,28 € an Kosten im Monat, die durch das Projekt vermeidbar sind, amortisiert sich die Investition nach 3,797 Jahren.

Formel : Amortisationsrechnung

### Entwurf

Der Raspberry Pi 4 Modell B wurde über die GPIO Pins gemäß der Anschlusstabelle (Tabelle 7) und Abbildung 3 mit dem DHT11 Sensor verbunden. Nun konnte, durch einen dem DHT11 Sensor beiliegendem Python-Beispielcode (Ibrahim, 2019), getestet werden ob der Sensor Daten liefert.

|  |  |
| --- | --- |
| Raspberry Pi | DHT11 |
| GPIO 23 (Pin 16) | Signal |
| +3,3 V (Pin 1) | +V |
| GND (Pin 6) | GND |

Tabelle : Anschlusstabelle Raspberry Pi - DHT11

War dieser Test erfolgreich, so konnte der Code weiter modifiziert werden um die Daten an einen öffentlichen MQTT Broker zu übertragen.

Node-RED sollte primär zum Empfangen der Daten und zur Erstellung der graphischen Nutzeroberfläche verwendet werden. Ein entsprechendes Dashboard ließ sich via Paletten-Import schnell hinzufügen.

Gewünscht waren Skalen und Liniendiagramme.

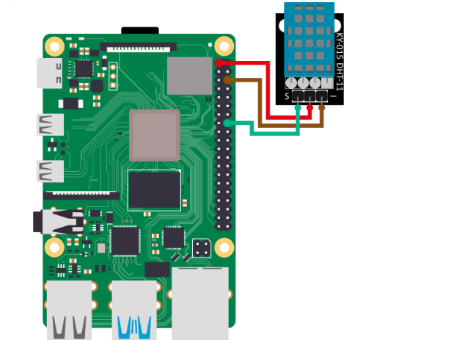


Abbildung : Möglicher Anschluss des DHT11 Sensors an den Raspberry Pi

### Implementierung

Während der Implementierung und der unter [2.5 Ablaufplan](#_Ablaufplan) geschilderten Verwendung der Selbstkontrolle durch Regelkreis kam es zu Verbesserungen des Systems die hier mit Versionen bezeichnet werden.

* Version 1:

Zu Beginn gab es eine Lösung mit Python Code der automatisiert die Werte des DHT11 Sensors an einen öffentlichen MQTT Broker übermittelte. Anschließend wurden diese Daten via MQTT Abonnement von Node-RED empfangen und optisch aufbereitet. Der Python Code musste hierfür von Hand auf dem Raspberry Pi gestartet werden.

* Version 2:

Durch Import der Entsprechenden Paletten ist Node-RED in der Lage Python Code selbständig auszuführen. Eine Übertragung an den öffentlichen MQTT Broker muss nun nicht mehr über den Python Code erfolgen, sondern wird von Node-RED übernommen.

* Version 3:

Nach Installation der BCM 2835 Bibliothek ist Node-RED in der Lage direkt Informationen des Sensors zu verarbeiten. Ein Python Skript ist nicht länger notwendig.

* Version 4:

Mit Installation des Eclipse Mosquitto™ Nachrichten Brokers ist kein öffentlicher MQTT Broker mehr notwendig. Die Verarbeitung und das Veröffentlichen werden nun komplett über den Raspberry Pi und Node-RED geregelt.

Somit ist eine Abhängigkeit von äußeren Komponenten nicht mehr gegeben und die Komplexität enorm gesunken. Dies fördert auch die Wartbarkeit. Darüber hinaus ist es so nun möglich dieses System ohne großen Aufwand zu erweitern oder auf eine andere Plattform zu übertragen.

Allen Versionen war eine Nutzung von Node-RED zur Erstellung einer graphischen Nutzeroberfläche gemein, die über http://<IP\_des\_Raspberry\_Pi>:1880/dashboard/ im Netzwerk des Raspberry Pi 4 Modell B zur Verfügung gestellt wurde.

Diese Nutzeroberfläche zeigte sowohl die Messwerte in Form von Skalen (Abbildung 4) als auch in Liniendiagrammen. Darüber hinaus wurde ein Warnhinweis per Pop-Up eingeblendet, wenn Messwerte über oder Unterschritten wurden. Wartung und Anpassung der Grenzwerte konnte über http://<IP\_des\_Raspberry\_Pi>:1880/ erfolgen.

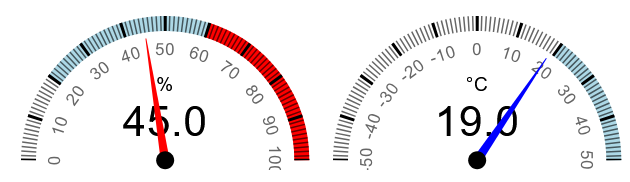


Abbildung : Skalen der Nutzeroberfläche

### Abnahme und Einführung

Die Abnahme erfolgte nach erfolgreicher Nachkontrolle des Ausbilders durch einen Mitarbeiter des Kunden im Rahmen einer Präsentation. Die Rückfragen des Kundenmitarbeiters konnten zufriedenstellend beantwortet werden. Im Anschluss wurde das System in Labor 1 in Betrieb genommen.

### Dokumentation

Im Rahmen der Dokumentation wurde das Projekt schriftlich festgehalten. Die ausgearbeitete Form liegt in diesem Schriftstück vor. Sie beinhaltet sowohl die Projektbeschreibung als auch die verwendeten Befehle ([siehe 5 Dokumentation](#_Dokumentation)).

## Vorgehensweise

Der Autor arbeitete in Einzelarbeit und ohne direkte Kontrolle an diesem Projekt. Für Fragen standen der Ausbilder und der Kundenvertreter zur Verfügung. Die Arbeit erfolgte primär am Arbeitsplatz des Autors und wurde erst nach der Endabnahme an den Einsatzort im Labor 1 verbracht.

## Qualitätssicherung

Zur Qualitätssicherung wurden die Überprüfung des Projekts durch den Ausbilder vereinbart.

## Abweichung

Wurde festgestellt, dass die Planung einen Aspekt nicht richtig berücksichtigt hatte, oder kamen Neuerungen hinzu, die zum Zeitpunkt der Planung noch unbekannt waren, so wurden diese Abweichungen unter Berücksichtigung der Kundenwünsche bewertet. Konnte keine eindeutige Aussage getroffen werden

## Anpassung

Anpassungen wurden vorgenommen, wenn sie eine Vereinfachung des Gesamtsystems oder der Handhabbarkeit darstellten, sofern sie weiterhin den Kundenwünschen entsprachen.

## Entscheidung

Die Entscheidung, welche Softwarelösung genutzt werden sollte, wurde mit Hilfe einer Entscheidungsmatrix getroffen. Da alle drei zur Auswahl gestellten Optionen (Java, Python und Node-RED) verschiedene Vorzüge hatten, wurde in der Entscheidungsmatrix die für den Autor wesentlichen Eigenschaften gegenübergestellt. Dem Autor waren Testbarkeit, sein eigener Kenntnisstand und die damit verbundene Einarbeitungsdauer, die Handhabbarkeit und das zur Verfügung stellen einer Benutzeroberfläche wichtig. Eine entsprechende Gewichtung der Eigenschaften führte zu einer Entscheidung für Node-RED (siehe Tabelle 8).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Eigenschaften | Gewichtung | Java | Python | Node-RED | Java | Python | Node-RED |
|  |  | (bew.) | (bew.) | (bew.) | (gew.) | (gew.) | (gew.) |
| Testbarkeit | 1 | 5 | 5 | 2 | 5 | 5 | 2 |
| Kenntnisstand Entwickler | 3 | 2 | 5 | 1 | 6 | 15 | 1 |
| Handhabbarkeit | 5 | 2 | 2 | 5 | 10 | 10 | 25 |
| Benutzeroberfläche | 5 | 2 | 2 | 5 | 10 | 10 | 25 |
|  |  |  |  |  |  |  |  |
| Gesamt | 14 |  |  |  | 31 | 40 | 53 |
| Nutzwert |  |  |  |  | 2,21 | 2,86 | 3,79 |

Tabelle : Entscheidungsmatrix

# Projektergebnisse

Abschließend sollen nun die erzielten Ergebnisse evaluiert werden.

## Soll – Ist- Vergleich

Die Übergabe an den Kunden enthielt, wie gewünscht, Hardwarekomponente mit dazugehöriger Softwarelösung die den Temperatur- und den Luftfeuchtigkeitswert erfasst, graphisch darstellt und als CSV-Datei abspeichert. Die Nutzeroberfläche ist in Netzwerk für jeden Nutzer verfügbar und ein Warn-Pop-Up weist auf das Über- und Unterschreiten von Grenzwerten hin. Eine Dokumentation liegt vor. Somit sind die in 1.4 Kundenwünsche erfassten Wünsche erfüllt.

## Abweichungen und Anpassungen

Bei der Verkabelung des Raspberry Pi 4 Modell B und des DHT11 Sensors wurde von der in der Dokumentation des DHT11 vorgeschlagenen Verkabelung abgewichen um den Cobbler und das Steckbrett zu nutzen. So konnte ein schneller Auf- und Abbau für einen einfachen Transport gewährleistet und eine mögliche Fehlerquelle durch falsches Anschließen des DHT11 Sensors vermieden werden. Zum Transport ist es nun nur noch notwendig das Flachbandkabel vom Cobbler zu trennen und am Ziel wieder zu verbinden. Durch die Gestaltung der Steckverbindung ist dies nur in einer Ausrichtung möglich und stellt so die richtige Neuverbindung sicher.

Im Rahmen der Gestaltung der Softwarelösung wurden mehrere Lösungsvorschläge erprobt, die jeweils ein größeres Verständnis der Software zur Folge hatten. Dies führte wiederum zu einer verbesserten Integration der Aufgaben in einzelne Softwarekomponenten und reduzierte somit die Komplexität und die benötigten Teilkomponenten. So wurde zum Beispiel zu Beginn des Projektes ein MQTT-Publisher in Python3 verwendet, auf den später dank der Funktionalität von Node-RED komplett verzichtet werden konnte. Das gleiche gilt auch für das Auslesen der Sensordaten des DHT11 Sensors. Diese wurden zuerst durch ein Python3 Script erfasst, was nach der Installation der BCM 2835 Bibliothek auf dem Raspberry Pi nicht mehr notwendig war, da Node-RED diese nun selbst erheben konnte.

Auch der MQTT-Broker wurde mit fortschreitendem Projekt überarbeitet. In den ersten Versionen der Softwarelösung wurden kostenfreie MQTT-Broker aus dem Internet verwendet. Doch auch darauf konnte nach Installation von Eclipse Mosquitto™ verzichtet werden, da hiermit ein eigener MQTT-Broker auf dem Raspberry Pi zur Verfügung gestellt werden konnte.

Da alle Anpassungen sich darum handelten die Funktionsweise von Node-RED weiter auszunutzen und somit die Wartbarkeit und Stabilität des Systems sicherten wurden die Veränderungen von Version 1 bis Version 4 vorgenommen.

Sonstige Abweichungen, z.B. im Zeitplan, kamen keine vor. Weitere Anpassungen waren somit nicht notwendig.

## Qualitätssicherung

Die Nachkontrolle des Ausbilders bestätigte den Autor in der Qualität der Arbeit. Eine Korrekturschleife war nicht notwendig.

# Dokumentation

Die Dokumentation beinhaltet die Übersicht über die notwendigen Vorbereitungen zur Neuinstallation. Die Reihenfolge ist dabei relevant und kann nicht beliebig ausgetauscht werden. Ist sich der Leser über die Vorgehensweise im Klaren, so kann er entsprechende Änderungen vornehmen.

## Vorbereitung des Raspberry Pis:

Es muss sichergestellt sein, dass der Raspberry Pi über ein funktionierendes und aktuelles Betriebssystem verfügt. Die Funktionalität der folgenden Befehle ist sonst nicht gewährleistet.

* Installation von npm:

sudo apt-get install npm

* Installation des MQTT Broker:

1. Herunterladen von Eclipse Mosquitto™

sudo apt install -y mosquitto mosquitto-clients

1. Registrierung des MQTT-Brokers als Service (für Autostart)

sudo systemctl enable mosquitto.service

1. Damit Clients sich ohne Benutzer und PW anmelden können muss die /mosquitto.conf bearbeitet werden:

sudo nano /etc/mosquitto/mosquitto.conf

An das Dateiende wird hinzugefügt:

listener 1883

allow\_anonymous true

1. Neustart Mosquitto Service

sudo systemctl restart mosquito

* Install Node-RED:

1. Herunterladen von Node-RED

bash <(curl -sL <https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered>)

1. Registrieren von Node-RED als Service (für Autostart)

sudo systemctl enable nodered.service

1. Neustart Node-RED Service

sudo systemctl restart nodered

* Installation der Sensoren Bibliothek BCM 2835:

1. Herunterladen der Bibliothek

wget <http://www.airspayce.com/mikem/bcm2835/bcm2835-1.75.tar.gz>

1. Entpacken der Bibliothek

tar zxvf bcm2835-1.75.tar.gz

1. Wechsel in das entpackte Verzeichnis

cd bcm2835-1.75

1. Konfiguration und Abschluss der Installation

./configure  
make  
sudo make check  
sudo make install

## Vorbereitung von Node-RED

* Installation der Paletten:
  + @flowfuse/node-red-dashboard
  + @colinl/node-red-dashboard-2-ui-gauge-classic
  + node-red-contrib-dht-sensor
  + @colinl/node-red-dashboard-2-ui-gauge-classic
* Import des Quellcodes[[2]](#footnote-2):

[{"id":"70ea02ec769faef7","type":"tab","label":"Flow 6","disabled":false,"info":"","env":[]},{"id":"8313db1951132671","type":"rpi-dht22","z":"70ea02ec769faef7","name":"Sensor","topic":"rpi-dht22","dht":"11","pintype":"0","pin":"23","x":260,"y":40,"wires":[["a6e9d7c7e3bc4adc"]]},{"id":"a932ccf193cbaaae","type":"inject","z":"70ea02ec769faef7","name":"","props":[],"repeat":"1","crontab":"","once":true,"onceDelay":0.1,"topic":"","x":110,"y":40,"wires":[["8313db1951132671"]]},{"id":"b4d0dc2fcb5fdde0","type":"mqtt out","z":"70ea02ec769faef7","name":"","topic":"E3FI/TempHumPiRed","qos":"2","retain":"","respTopic":"","contentType":"","userProps":"","correl":"","expiry":"","broker":"bf137e49e1cd17fa","x":640,"y":40,"wires":[]},{"id":"2a427d2e27f7fe5c","type":"function","z":"70ea02ec769faef7","name":"Humidity function","func":"const humidity = msg.payload.humidity;\nmsg.payload=humidity;\nreturn msg;","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs":[],"x":410,"y":160,"wires":[["f8e9d5bee80237b1","ae8b84e23170421e"]]},{"id":"9484717e4f2b83eb","type":"mqtt in","z":"70ea02ec769faef7","name":"","topic":"E3FI/TempHumPiRed","qos":"2","datatype":"json","broker":"bf137e49e1cd17fa","nl":false,"rap":true,"rh":0,"inputs":0,"x":140,"y":260,"wires":[["2a427d2e27f7fe5c","cf1ebb31bfd64f0d","b7dbb29a63d21bec","ab2b4213d81a1158"]]},{"id":"f8e9d5bee80237b1","type":"ui-gauge","z":"70ea02ec769faef7","name":"Luftfeuchtigkeit","group":"3df67dd039124b5e","order":1,"width":"2","height":"4","gtype":"gauge-half","gstyle":"needle","title":"Humidity","units":"","icon":"water-percent","prefix":"","suffix":"","segments":[{"from":"0","color":"#5cd65c"},{"from":"40","color":"#ffc800"},{"from":"70","color":"#ff0000"}],"min":0,"max":"100","sizeThickness":16,"sizeGap":4,"sizeKeyThickness":8,"styleRounded":true,"styleGlow":false,"className":"","x":660,"y":160,"wires":[]},{"id":"ae8b84e23170421e","type":"ui-chart","z":"70ea02ec769faef7","group":"3df67dd039124b5e","name":"","label":"Humidity Chart","order":2,"chartType":"line","category":"topic","categoryType":"msg","xAxisLabel":"","xAxisProperty":"","xAxisPropertyType":"timestamp","xAxisType":"time","xAxisFormat":"","xAxisFormatType":"HH:mm","xmin":"","xmax":"","yAxisLabel":"Humidity","yAxisProperty":"payload","yAxisPropertyType":"msg","ymin":"","ymax":"","bins":10,"action":"append","stackSeries":false,"pointShape":"circle","pointRadius":4,"showLegend":false,"removeOlder":"5","removeOlderUnit":"60","removeOlderPoints":"","colors":["#0095ff","#ff0000","#ff7f0e","#2ca02c","#a347e1","#d62728","#ff9896","#9467bd","#c5b0d5"],"textColor":["#666666"],"textColorDefault":true,"gridColor":["#e5e5e5"],"gridColorDefault":true,"width":"10","height":"6","className":"","x":660,"y":120,"wires":[[]]},{"id":"b069155e609012af","type":"ui-chart","z":"70ea02ec769faef7","group":"81f159a36b29c383","name":"","label":"Temperature Chart","order":2,"chartType":"line","category":"topic","categoryType":"msg","xAxisLabel":"","xAxisProperty":"","xAxisPropertyType":"timestamp","xAxisType":"time","xAxisFormat":"","xAxisFormatType":"HH:mm","xmin":"","xmax":"","yAxisLabel":"Temperature","yAxisProperty":"payload","yAxisPropertyType":"msg","ymin":"","ymax":"","bins":10,"action":"append","stackSeries":false,"pointShape":"circle","pointRadius":4,"showLegend":false,"removeOlder":"5","removeOlderUnit":"60","removeOlderPoints":"","colors":["#0095ff","#ff0000","#ff7f0e","#2ca02c","#a347e1","#d62728","#ff9896","#9467bd","#c5b0d5"],"textColor":["#666666"],"textColorDefault":true,"gridColor":["#e5e5e5"],"gridColorDefault":true,"width":"10","height":"6","className":"","x":670,"y":400,"wires":[[]]},{"id":"cf1ebb31bfd64f0d","type":"function","z":"70ea02ec769faef7","name":"Temperature function","func":"var temperature = msg.payload.temperature;\nmsg.payload=temperature;\nreturn msg;","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs":[],"x":420,"y":360,"wires":[["4addbf8859b4f1a6","51c45fb1fa218744","b069155e609012af"]]},{"id":"56c607ad973657e0","type":"csv","z":"70ea02ec769faef7","name":"CSV-Parser","spec":"rfc","sep":",","hdrin":true,"hdrout":"once","multi":"one","ret":"\\n","temp":"temperature,humidity","skip":"1","strings":false,"include\_empty\_strings":"","include\_null\_values":"","x":530,"y":260,"wires":[["6de665297d503777"]]},{"id":"4addbf8859b4f1a6","type":"ui-gauge","z":"70ea02ec769faef7","name":"Temperature","group":"81f159a36b29c383","order":1,"width":"2","height":"4","gtype":"gauge-half","gstyle":"needle","title":"Temperature","units":"","icon":"temperature-celsius","prefix":"","suffix":"","segments":[{"from":"-50","color":"#80ffff"},{"from":"-10","color":"#65f3c9"},{"from":"18","color":"#5cd65c"},{"from":"25","color":"#ffc800"},{"from":"35","color":"#ea5353"}],"min":"-50","max":"+50","sizeThickness":16,"sizeGap":4,"sizeKeyThickness":8,"styleRounded":true,"styleGlow":false,"className":"","x":650,"y":360,"wires":[]},{"id":"51c45fb1fa218744","type":"function","z":"70ea02ec769faef7","name":"Warnmeldungen","func":"if (msg.payload >= 25) {\n msg.payload = \"ACHTUNG! ÜBERHITZUNG!\"\n return msg\n} else if (msg.payload < 20) {\n msg.payload = \"Achtung! Zu kalt!\"\n return msg\n}","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs":[],"x":660,"y":460,"wires":[["45bc76f6ab5a6ab2"]]},{"id":"6de665297d503777","type":"file","z":"70ea02ec769faef7","name":"node-red-output","filename":"node-red-output.csv","filenameType":"str","appendNewline":false,"createDir":false,"overwriteFile":"false","encoding":"none","x":740,"y":260,"wires":[[]]},{"id":"45bc76f6ab5a6ab2","type":"ui-notification","z":"70ea02ec769faef7","ui":"d8803d39422b1c35","position":"center center","colorDefault":false,"color":"#ff0000","displayTime":"3","showCountdown":true,"outputs":1,"allowDismiss":true,"dismissText":"Close","allowConfirm":false,"confirmText":"Confirm","raw":false,"className":"","name":"Temperature-Message","x":880,"y":460,"wires":[[]]},{"id":"b7dbb29a63d21bec","type":"json","z":"70ea02ec769faef7","name":"","property":"payload","action":"","pretty":false,"x":370,"y":260,"wires":[["56c607ad973657e0"]]},{"id":"a6e9d7c7e3bc4adc","type":"function","z":"70ea02ec769faef7","name":"Create JSON","func":"const temperature = msg.payload;\nconst humidity = msg.humidity;\nmsg.payload = {\n \"temperature\": temperature,\n \"humidity\": humidity\n}\nreturn msg;","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs":[],"x":430,"y":40,"wires":[["b4d0dc2fcb5fdde0"]]},{"id":"ab2b4213d81a1158","type":"function","z":"70ea02ec769faef7","name":"function 6","func":"const temperature = msg.payload.temperature;\nconst humidity = msg.payload.humidity;\nvar msg = {};\nvar msg1 = {};\nmsg.topic='Temperatur'\nmsg.payload=temperature;\nmsg1.topic='Luftfeuchtigkeit'\nmsg1.payload=humidity;\nreturn [msg, msg1];","outputs":2,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs":[],"x":310,"y":620,"wires":[["328dfdb9e589f7a9","81279d491678acac"],["81279d491678acac","89706d66b2ea4a7b"]]},{"id":"81279d491678acac","type":"ui-chart","z":"70ea02ec769faef7","group":"bad1fb197a22ad60","name":"Test","label":"","order":3,"chartType":"line","category":"topic","categoryType":"msg","xAxisLabel":"","xAxisProperty":"","xAxisPropertyType":"timestamp","xAxisType":"time","xAxisFormat":"","xAxisFormatType":"HH:mm","xmin":"","xmax":"","yAxisLabel":"","yAxisProperty":"payload","yAxisPropertyType":"msg","ymin":"","ymax":"","bins":10,"action":"append","stackSeries":false,"pointShape":"circle","pointRadius":4,"showLegend":false,"removeOlder":1,"removeOlderUnit":"3600","removeOlderPoints":"","colors":["#0095ff","#ff0000","#ff7f0e","#2ca02c","#a347e1","#d62728","#ff9896","#9467bd","#c5b0d5"],"textColor":["#666666"],"textColorDefault":true,"gridColor":["#e5e5e5"],"gridColorDefault":true,"width":6,"height":8,"className":"","x":530,"y":620,"wires":[[]]},{"id":"328dfdb9e589f7a9","type":"ui-gauge-classic","z":"70ea02ec769faef7","name":"","group":"bad1fb197a22ad60","order":2,"width":"2","height":"1","min":"-50","max":"50","sectors":[{"start":"0","color":"Iceblue"},{"start":"20","color":"lightblue"},{"start":"60","color":"red"}],"major\_division":"10","minor\_division":"1","value\_decimal\_places":1,"scale\_decimal\_places":0,"label":"","measurement":"","units":"°C","needles":[{"topic":"Temperatur","color":"Blue","lengthPercent":"100"}],"sweep\_angle":"180","myclass":"","x":560,"y":580,"wires":[]},{"id":"89706d66b2ea4a7b","type":"ui-gauge-classic","z":"70ea02ec769faef7","name":"","group":"bad1fb197a22ad60","order":1,"width":"2","height":"1","min":"0","max":"100","sectors":[{"start":"0","color":"Iceblue"},{"start":"20","color":"lightblue"},{"start":"60","color":"red"}],"major\_division":"10","minor\_division":"1","value\_decimal\_places":1,"scale\_decimal\_places":0,"label":"","measurement":"","units":"%","needles":[{"topic":"Luftfeuchtigkeit","color":"Red","lengthPercent":"100"}],"sweep\_angle":"180","myclass":"","x":560,"y":680,"wires":[]},{"id":"bf137e49e1cd17fa","type":"mqtt-broker","name":"Local-Broker","broker":"192.168.0.11","port":"1883","clientid":"","autoConnect":true,"usetls":false,"protocolVersion":"4","keepalive":"60","cleansession":true,"autoUnsubscribe":true,"birthTopic":"","birthQos":"0","birthRetain":"false","birthPayload":"","birthMsg":{},"closeTopic":"","closeQos":"0","closeRetain":"false","closePayload":"","closeMsg":{},"willTopic":"","willQos":"0","willRetain":"false","willPayload":"","willMsg":{},"userProps":"","sessionExpiry":""},{"id":"3df67dd039124b5e","type":"ui-group","name":"Pure NODE-RED-Humid","page":"82b622ddbdde2ab9","width":"12","height":"6","order":3,"showTitle":false,"className":"","visible":"true","disabled":"false","groupType":"default"},{"id":"81f159a36b29c383","type":"ui-group","name":"Pure NODE-RED-Temp","page":"82b622ddbdde2ab9","width":"12","height":"6","order":2,"showTitle":false,"className":"","visible":"true","disabled":"false","groupType":"default"},{"id":"d8803d39422b1c35","type":"ui-base","name":"My Dashboard","path":"/dashboard","appIcon":"","includeClientData":true,"acceptsClientConfig":["ui-notification","ui-control"],"showPathInSidebar":false,"showPageTitle":true,"navigationStyle":"default","titleBarStyle":"default"},{"id":"bad1fb197a22ad60","type":"ui-group","name":"Mix","page":"82b622ddbdde2ab9","width":"12","height":"1","order":1,"showTitle":false,"className":"","visible":"true","disabled":"false","groupType":"default"},{"id":"82b622ddbdde2ab9","type":"ui-page","name":"Pure NODE-RED ","ui":"d8803d39422b1c35","path":"/page4","icon":"home","layout":"grid","theme":"4add868658f741a9","breakpoints":[{"name":"Default","px":"0","cols":"3"},{"name":"Tablet","px":"576","cols":"6"},{"name":"Small Desktop","px":"768","cols":"9"},{"name":"Desktop","px":"1024","cols":"12"}],"order":1,"className":"","visible":"true","disabled":"false"},{"id":"4add868658f741a9","type":"ui-theme","name":"Default Theme","colors":{"surface":"#ffffff","primary":"#0094CE","bgPage":"#eeeeee","groupBg":"#ffffff","groupOutline":"#cccccc"},"sizes":{"density":"default","pagePadding":"12px","groupGap":"12px","groupBorderRadius":"4px","widgetGap":"12px"}}]

# Literaturverzeichnis

Electronics, M., kein Datum *DHT11 Humidity & Temperature Sensor - DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf.* [Online]   
Available at: https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf  
[Zugriff am 22 November 2024].

Ibrahim, P. D. D., 2019. *The Ultimate Compendium of Sensor Projects.* London: Elektor International Media B.V..

# Anhang

## Detaillierte Zeitplanung

|  |  |
| --- | --- |
| Analysephase | 8 h |
| 1. ISt-Analyse durchführen | 4 h |
| 1. Wirtschaftlichkeitsanalyse und Nutzwertanalyse | 2 h |
| 1. Unterstützung des FAchbereichs bei der ERstellung des Lastenhefts | 2 h |

©

|  |  |
| --- | --- |
| Entwurfsphase | 10 h |
| 1. Messstation aufbauen | 1 h |
| 1. Installation der Softwarekomponenten | 1 h |
| 1. Benutzeroberfläche entwerfen | 1 h |
| 1. Planung der Logik | 4 h |
| 1. Pflichtenheft erstellen | 3 h |

|  |  |
| --- | --- |
| Implementierung | 20 h |
| 1. Implementierung von MQTT | 2 h |
| 1. Implementierung der Node-RED Logik | 6 h |
| 1. Implementierung der SEnsorik | 4 h |
| 1. Re-evaluation | 3 h |
| 1. Implementierung der Nutzeroberfläche inkl. TEsts | 5 h |

|  |  |
| --- | --- |
| NAchkontrolle | 1 h |
| 1. Kontrolle durch den Ausbilder | 1 h |

|  |  |
| --- | --- |
| Abnahme und Einführung | 1 h |
| 1. Abnahme durch den KundenMitarbeiter | 0,5 h |
| 1. Einrichtung der Messstation in Labor 1 | 0,5 h |

|  |  |
| --- | --- |
| Dokumentation | 10 h |
| 1. Erstellen der Projektdokumentation | 8 h |
| 1. Erstellen der Entwicklerdokumentation | 1 h |
| 1. Erstellen der BEnutzerdokumentation | 1 h |
|  |  |
| Gesamt: | 50 h |

1. Da die Komponenten durch den Kunden zur Verfügung gestellt wurden, hat der Autor diese Preise nachträglich recherchiert um eine möglichst aktuelle Kalkulation vornehmen zu können. Die tatsächlichen Anschaffungskosten können durch Rabatte und Preisschwankungen abweichen. [↑](#footnote-ref-1)
2. Der Quellcode ist der Vollständigkeit und nicht der Lesbarkeit halber eingefügt. Das kopieren und importieren in Node-RED wird durch die Schriftgröße nicht beeinflusst. Die Lesbarkeit des Gesamtdokuments wird durch die geringe Schriftgröße beibehalten. [↑](#footnote-ref-2)