

---

# Wetteranalyse

## - Abschlussarbeit Informatik -

Leon Hübner

Aus der 10/1 vom Wilhelm-Ostwald-Gymnasium

März bis Juni 2022

---

# Inhaltsverzeichnis

Kurzreferat.....	3
1. Einleitung.....	4
1.1. Themenwahl.....	4
1.2. Zielstellung .....	4
1.3. Motivation.....	4
2. Vorbetrachtung .....	4
2.1. Meteorologie und Wetterbeobachtung .....	4
2.2. Hardware.....	5
3. Ergebnisse.....	7
3.1. Grafische Benutzeroberfläche (GUI).....	7
<b>3.1.1. Allgemeines</b> .....	7
<b>3.1.2. Startfenster</b> .....	7
<b>3.1.3. Messungs-Fenster</b> .....	8
<b>3.1.4. Auswertungs-Fenster</b> .....	8
<b>3.1.5. Menüs</b> .....	9
3.2. Genutzte Bibliotheken.....	9
3.3. Auswertung-Beispiele .....	10
4. Methoden .....	12
4.1. Messungen .....	12
4.2. Auswertungen .....	13
5. Diskussion.....	14
6. Abbildungen.....	16
7. Quellen.....	18
7.1. Literatur .....	18
8. Selbstständigkeitserklärung <sup>5</sup> .....	18

## Kurzreferat

Im Zeitraum des zweiten Schulhalbjahres wurde im Informatik-Unterricht von Leon Hübner der Klasse 10/1 eine Applikation für die Erfassung und Auswertung von Wetterdaten erstellt. Die Software ist in der Lage, regelmäßige Messungen in festgelegten Intervallen mittels eines geeigneten Messgerätes durchzuführen. Die Messungen beinhalten dabei für das Wetter relevante Größen, darunter Temperatur und Luftfeuchtigkeit. Bei der Auswertung wird ein Diagramm erstellt, das die Messwerte in Abhängigkeit von der Zeit darstellt. Das Programm zeichnet sich durch seine Einfachheit und große Nutzerfreundlichkeit aus. Es findet in vielerlei Bereichen Anwendungsmöglichkeiten, wobei es hauptsächlich für die Langzeitmessung von Wetter konzipiert wurde.

# 1. Einleitung

## 1.1. Themenwahl

Bei dem Projekt handelt es sich um eine Applikation für die naturwissenschaftliche Analyse, genauer gesagt der Analyse von Wetterdaten. Damit ist es in die Meteorologie, die Wissenschaft über die Prozesse der Atmosphäre, einzuordnen.

## 1.2. Zielstellung

Das Ziel war ein Programm, das in der Lage ist, physikalische Größen aufzunehmen und anschließend auszuwerten. Zu diesen Größen gehören für das Wetter relevante Eigenschaften, insbesondere Temperatur, Luftfeuchtigkeit und Luftdruck. Die Auswertung besteht aus einem Diagramm sowie statistischen Größen.

## 1.3. Motivation

Die meisten Menschen nutzen täglich einen Wetterbericht, den sie zum Beispiel im Radio hören. Er ist nützlich, um sich wettergerecht zu kleiden, aber auch für die Planung des Tagesablaufs und Aktivitäten, die im Freien stattfinden. Wetterberichte sind Prognosen für zukünftige Entwicklungen auf Grundlage von Messdaten. Möglichst solide Wetter-Daten zu erfassen und auszuwerten, ist daher ein zentraler Bestandteil der Wetterberichte. Aber auch wissenschaftlicher Experimente oder die Überwachung von Systemen wie Aquarien benötigen die Erfassung und Auswertung von physikalischen Größen.

Eine Software, die diese Aufgabe erfüllen kann, hat daher viele Einsatzbereiche, für die sich eine Entwicklung lohnt.

# 2. Vorbetrachtung

## 2.1. Meteorologie und Wetterbeobachtung

Die Meteorologie ist die Wissenschaft der physikalischen und chemischen Vorgänge der Atmosphäre. Das Projekt beschränkt sich auf die Aufnahme und Auswertung von physikalischen Größen, weshalb es in das Teilgebiet Wetterbeobachtung einzuordnen ist. Im Gegensatz zu anderen Teilgebieten sucht die Wetterbeobachtung keine wissenschaftlichen Erklärungen für die Messwerte. Sie ist damit aber auch die Grundlage für alle anderen Gebiete.

Bereits seit Jahrtausenden beschäftigen sich die Menschen mit der Beobachtung des Wetters, die vor allem für die Aussaat und Ernte wichtig war. Es entstanden die sogenannten Bauernregeln. 1781 begann ein Observatorium in Bayern mit der „professionellen“ Wetterbeobachtung und nahm regelmäßig Messwerte auf. Im 19. Jh. wurden erste Wetterballons für die Messung in großen Höhen entwickelt. Heute gibt es unzählige Messstationen auf dem Land, in Bojen auf dem Meer und in der Luft als Wetterballons. Sie werden von den Wetterdiensten betrieben und senden diesen regelmäßig auf digitalem Wege ihre Daten.<sup>1</sup>

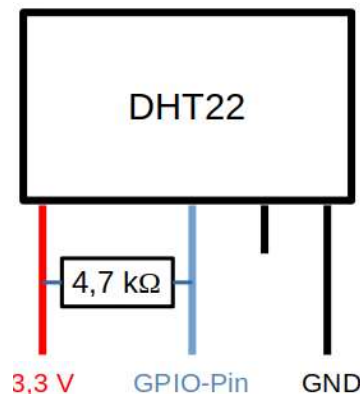
Man kann die Wetterbeobachtung in zwei Bereiche unterteilen. Der erste umfasst Eigenschaften, die man nicht direkt messen kann, sondern wortwörtlich beobachten muss. Dazu zählen zum Beispiel die Art der Wolken, deren Menge, evtl. Nebel oder auch die Verfärbung beim Sonnenaufgang und -untergang. Der zweite Bereich umfasst direkt messbare Größen, unter anderem Temperatur, Luftfeuchtigkeit und -druck, Helligkeit, Niederschlagsmenge und einige mehr. Für das Projekt wurden wenige Größen gewählt, die mit geeigneten Sensoren leicht zu messen sind.

## 2.2. Hardware

Die Hardware setzt sich aus zwei verschiedenen Komponenten zusammen. Sensoren (DHT22 und BMP280) dienen der direkten Messung der verschiedenen Größen. Für die Verarbeitung (Umwandlung in Daten) und die Speicherung wird ein Minicomputer (Raspberry Pi) verwendet.

Als Minicomputer wurde der Raspberry Pi 2 Model B genutzt. Dieser besitzt eine sogenannte GPIO-Leiste mit über 40 Anschlüssen, welche als digitale Ein- und Ausgänge verwendet werden. Sie können zwei Zustände („Low“ oder „High“) annehmen. In diesem Projekt werden die Sensoren über geeignete Kabel mit einem bzw. zwei dieser GPIO-Pins verbunden. Weil nur digitale Signale aufgenommen werden können, müssen Sensoren die analogen Messwerte durch einen integrierten AD-Wandler (Analog-Digital-Wandler) in digitale Signale „übersetzen“. Die Sensoren liefern daher eine Folge von Nullen und Einsen (= digitales Signal), die letztendlich durch den Raspberry Pi 2 in Messwerte übersetzt werden müssen. Dazu wurden Bibliotheken der jeweiligen Hersteller genutzt.<sup>3</sup>

Für die Aufnahme der Messwerte standen zwei verschiedene Sensoren zur Verfügung. Der Erste ist der DHT-22, auch bekannt als AM2302. Dieser Sensor ist in der Lage Temperatur (-40 bis 80 °C) und Luftfeuchtigkeit (0 – 100 %) zu messen. Die Genauigkeit der Temperaturmessung liegt bei  $\pm 0,5$  °C und die der Luftfeuchtigkeit bei  $\pm 2-5$  %.<sup>2</sup> Der Sensor verfügt über vier Anschlüsse, einen für eine Eingangsspannung von 3,3 V, einen für die Masse und einen für die Übertragung der Messwerte. Der vierte Anschluss wird für das Messen nicht benötigt. Angeschlossen wird das Ganze nach folgendem Schaltplan:



Der Widerstand von 4,7 kΩ dient als sogenannter „Pull-Up-Widerstand“, der die Spannung erhöht, so dass die Spannung des „Low“-Zustands am GPIO-Pin größer als Null ist.

Bei dem zweiten Sensor handelt es sich um den BMP280 der Firma Bosch. Neben der Temperatur (-40 °C –bis 85 °C, mit  $\pm 0,5$  °C) kann dieser Sensor den Luftdruck (300 bis 1100 hPa, mit  $\pm 0,12$  hPa) messen.<sup>4</sup> Der BMP280 hat sechs Anschlüsse, wobei nur vier benutzt werden. Hier entfällt der Pull-Up-Widerstand. Dafür gibt es zwei Anschlüsse, die Signale übertragen. Sie haben die Bezeichnung „SCL“, sowie „SDO“ und werden an zwei unterschiedliche GPIO-Pins angeschlossen. Insgesamt ist folgende Zuordnung einzuhalten:

<b>BMP280</b>	<b>Raspberry Pi</b>
VCC	3,3 V
GND	Masse/GND
SCL	GPIO-Pin x (beliebig)
SDO	GPIO-Pin y (beliebig, aber nicht x)

## 3. Ergebnisse

### 3.1. Grafische Benutzeroberfläche (GUI)

#### 3.1.1. Allgemeines

Das Projekt verfügt über eine GUI, die aus drei bedeutsamen Fenstern besteht. Diese drei Fenster tragen die Bezeichnung „Startfenster“, „Messungs-Fenster“ und „Auswertungs-Fenster“. Sie haben jeweils eine eigene Aufgabe, die im Folgenden erläutert wird. Zunächst sei jedoch darauf hingewiesen, dass allen Elementen/Widget-Namen der jeweiligen Fenster eine feste Bezeichnungsvorschrift zu Grunde liegt, die zur Übersichtlichkeit des Codes beiträgt. Je nach Klasse der jeweiligen Objekte beginnt die Bezeichnung mit unterschiedlichen Großbuchstaben bzw. teilweise Wörtern. Folgende Abkürzungen wurden häufig verwendet:

B – Button	L – Label (ohne Änderung des Textes)
E – Eingabe-Zeile	A – Label (mit Änderung des Textes)
Check – Checkbox	Combo – Combobox

Daraufhin folgt ein Unterstrich und die Funktion oder Text des Elementes (häufig bei Labels) oder auch der Text der zugehörigen Überschrift. So entstehen beispielsweise diese Bezeichnungen:

B\_pfadWaehlen (Button, der den Dateexplorer öffnet)

A\_Pfad (Label, das den ausgewählten Pfad anzeigt)

#### 3.1.2. Startfenster

Begrüßt wird der Nutzer zunächst durch das Startfenster, welches lediglich aus einer Abbildung, sowie zwei Buttons besteht. Die Abbildung zielt auf einen lebendigen und ansprechenden ersten Eindruck ab. An diesem Punkt hat der Nutzer die Entscheidung zu fällen, was er mit der Software anfangen möchte. Er kann entweder eine Messung starten (oberer Button) oder eine bereits getätigte Messung auswerten (unterer Button). Die strikte Trennung dieser beiden Bereiche dient der Übersichtlichkeit und Nutzerfreundlichkeit. Nach dem Klick auf einen der beiden Buttons öffnet sich das jeweilige Fenster zur Messung bzw. Auswertung.

### **3.1.3. Messungs-Fenster**

Wie der Name schon sagt, hat dieses Fenster die Funktion eine Messung zu starten. Dazu wird durch Klick auf einen Button zunächst über einen Dateiexplorer der gewünschte Speicherpfad ausgewählt. Unter diesem Button befindet sich eine Eingabezeile für den Namen der Messung. Später wird ein Ordner mit diesem Namen in dem ausgewählten Pfad für die Messwerte erstellt. Hinter der Eingabezeile befindet sich ein Ausgabe-Label, das den Nutzer daraufhin weist, ob der eingegebene Name verwendbar ist (enthält kein „/“, ist nicht „.“ und ist noch nicht vorhanden). Nach der Auswahl des Speicherortes und -namen wird der verwendete Sensor durch eine Combo-Box ausgewählt. Je nach Sensor werden darunter ein oder zwei Eingabefelder für die Pins angezeigt. Dies erleichtert dem Nutzer die Eingabe und beugt Fehlern vor. Im nächsten Abschnitt sind Angaben zur Messung einzugeben, von denen genau zwei notwendig sind und die dritte berechnet werden kann. Es reichen daher zwei ausgefüllte Felder. Sobald zwei Felder ausgefüllt sind, ist das dritte nicht mehr editierbar. Der darunterliegende Button startet schließlich die Messung, deren Ergebnisse auch in der ganz unten platzierten Text-Area ausgegeben werden. Mit einem weiteren Button kann die Messung bei Bedarf abgebrochen werden.

### **3.1.4. Auswertungs-Fenster**

Das Auswertungs-Fenster stellt die Minima, Maxima und Durchschnitte der gewählten Größen der Messung in einer Tabelle dar und zeichnet den Graphen der Größen in Abhängigkeit von der Zeit in ein Diagramm, das einem Layout hinzugefügt wird. Zunächst wird wie bei Messungen der Pfad gewählt. Anschließend wählt der Nutzer über Check-Boxen die auszuwertenden Größen. Darunter befindet sich eine weitere Checkbox, die dem Nutzer die Möglichkeit gibt, seine Auswertung auf ein bestimmtes Zeitfenster zu begrenzen. Erst wenn der Nutzer diese Box anklickt, werden die zuvor blassen Eingabefelder für die Zeitangaben von Jahr bis zu Sekunden editierbar. Um die Arbeit des Nutzers zu minimieren, sind die Eingabefelder bereits mit den Werten von Beginn und Ende der Messung ausgefüllt, sodass diese nur nach Bedarf geändert werden müssen. Nach Klick auf den Button „Auswerten“ wird entweder eine Fehlermeldung direkt über dem Button durch ein Label angezeigt oder die Tabelle und das Diagramm erscheinen. Unter dem Diagramm werden außerdem Optionen zur Anpassung des Diagramms, sowie



ein Button, der das Diagramm in einem neuen Fenster öffnet, sichtbar. Der Nutzer kann sich immer auf das Wesentliche fokussieren, denn die Elemente werden immer nur nach Bedarf ein- oder ausgeblendet.

### 3.1.5. Menüs

Messungs- und Auswertungsfenster verfügen jeweils über ein Menü. Beide haben dabei die Kategorie „Datei“ mit den beiden Optionen „Schließen“ und „Beenden“. Während ein Klick auf „Schließen“ nur das Fenster verschwinden lässt (Python Befehl: `self.close()`), schließt „Beenden“ alle Fenster und beendet somit das Programm (Python Befehl: `sys.exit()`). Des Weiteren bietet das Menü des Messungs-Fensters die Kategorie „Fenster“ an, mit der das Auswertungsfenster, sowie das Startfenster geöffnet werden können. Dies ist praktisch, um direkt nach einer Messung zur Auswertung zu gelangen. Zum Schluss gibt es für unerfahrene Nutzer eine „Hilfe“ Kategorie. Diese bietet für Messungs- und Auswertungsfenster jeweils eine eigene Kurzanleitung, sowie eine kurze Information über den Herausgeber der Software. Beim Messungsfenster findet der Nutzer außerdem verlinkte Tutorial-Seiten (Python Befehl: `webbrowser.open(url)`) für die Sensoren.

## 3.2. Genutzte Bibliotheken

Es wurden diverse Bibliotheken genutzt. Besonders bedeutsam war für die Auswertung das Diagramm, dass mit Hilfe von „matplotlib“ erstellt wurde. Dieser Bibliothek werden zwei Arrays für x- und y-Werte übergeben und diese erstellt dann das Diagramm, das in meinem Fall als Widget in ein Layout eingefügt wurde. Folgender Code erstellte den Graphen:

```
import matplotlib.pyplot as plt

plt.plot(xWerte, yWerte)
```

Das Einbetten in ein Layout erfolgt auf diese Weise:

```
self.figur = plt.figure()

self.canvas = FigureCanvas(self.figur)

self.ui.L_diagramm.addWidget(self.canvas)
```

Außerdem bietet „matplotlib“ verschiedene Funktionen, das Erscheinungsbild des Diagramms zu ändern. Farben der Graphen, Beschriftungen, Skalen der

Achsen und vieles mehr lässt sich durch die entsprechenden Funktionen anpassen.

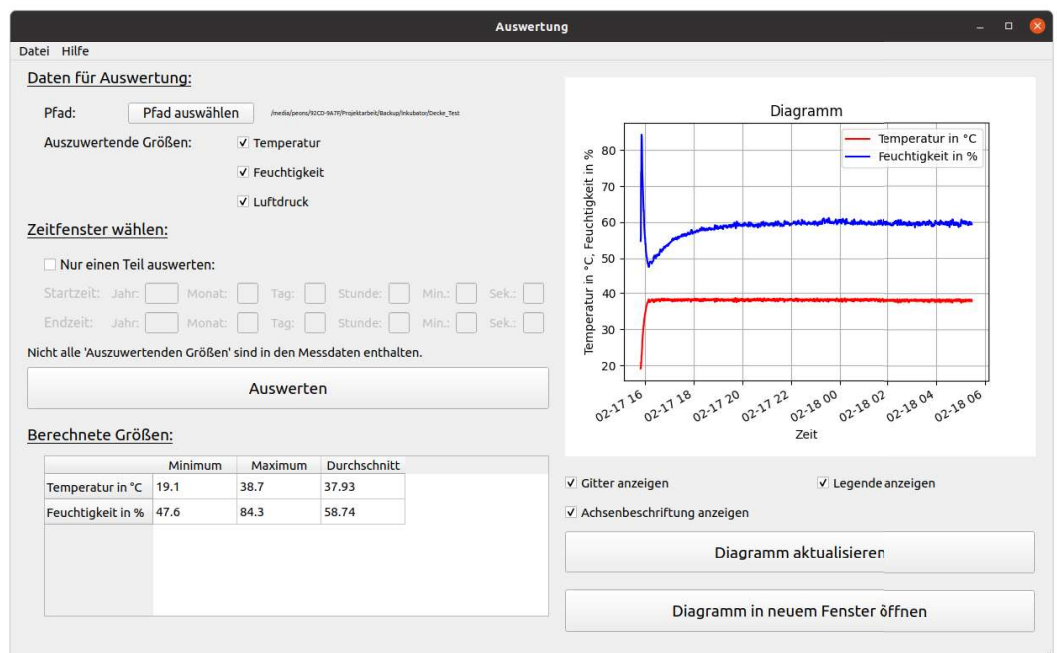
Für den Sensor DHT22 ist eine Bibliothek für das Einlesen der Messwerte notwendig. Genutzt wurde die Bibliothek Adafruit\_DHT. Diese ermöglicht das Einlesen der Messdaten bei korrektem Anschluss eines DHT22 an einem Raspberry Pi. Weil Raspberry Pis normalerweise eine Linux-Distribution bereitstellen, ist es nicht möglich, diese Bibliothek auf Windows zu installieren. Zu Testzwecken habe ich daher eine Erzeugung von zufälligen Werten eingebaut. Dasselbe gilt für die Bibliothek CIRCUIT, die für den Sensor BMP280 genutzt wird. Folgender Code würde auf einem Raspberry die Messwerte liefern:

```
DHT22: f,t=Adafruit_DHT.read_retry(sensor, gpio)
```

```
BMP280: t = bmp280.get_temperature()  
  
p = bmp280.get_pressure()
```

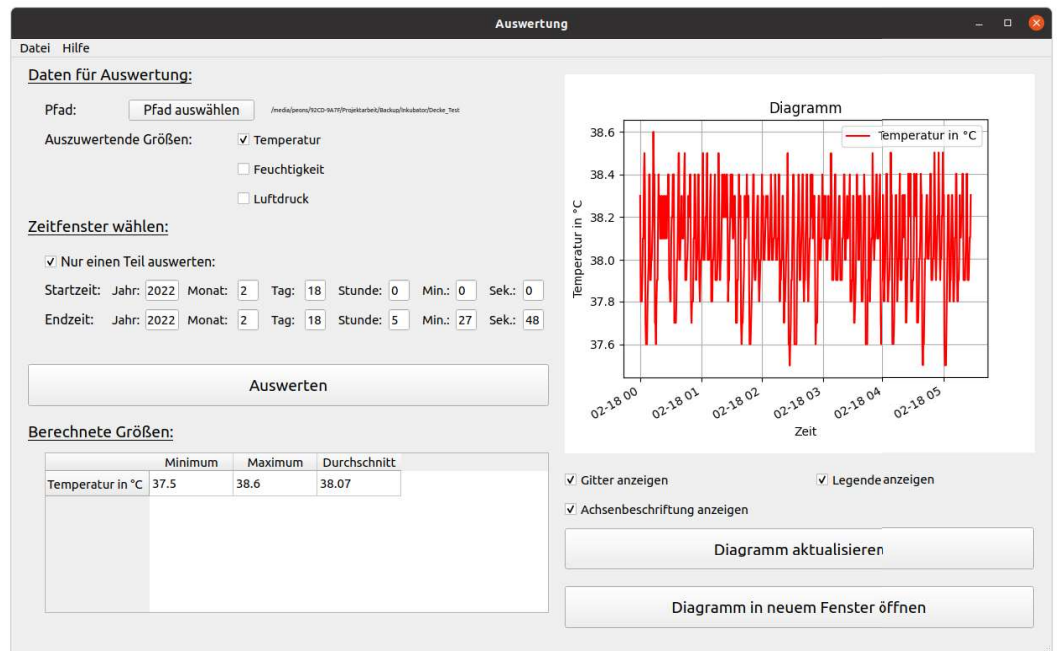
### 3.3. Auswertung-Beispiele

Das Messungsfenster erzeugt erfolgreich Dateien mit Messdaten. Der Nutzer könnte sich diese zwar durchlesen, aber die eigentliche Ausgabe des Programms sollten viel mehr das Diagramm und die dazu berechneten Größen des Auswertungsfensters sein, von denen hier einige Beispiele abgebildet sind:

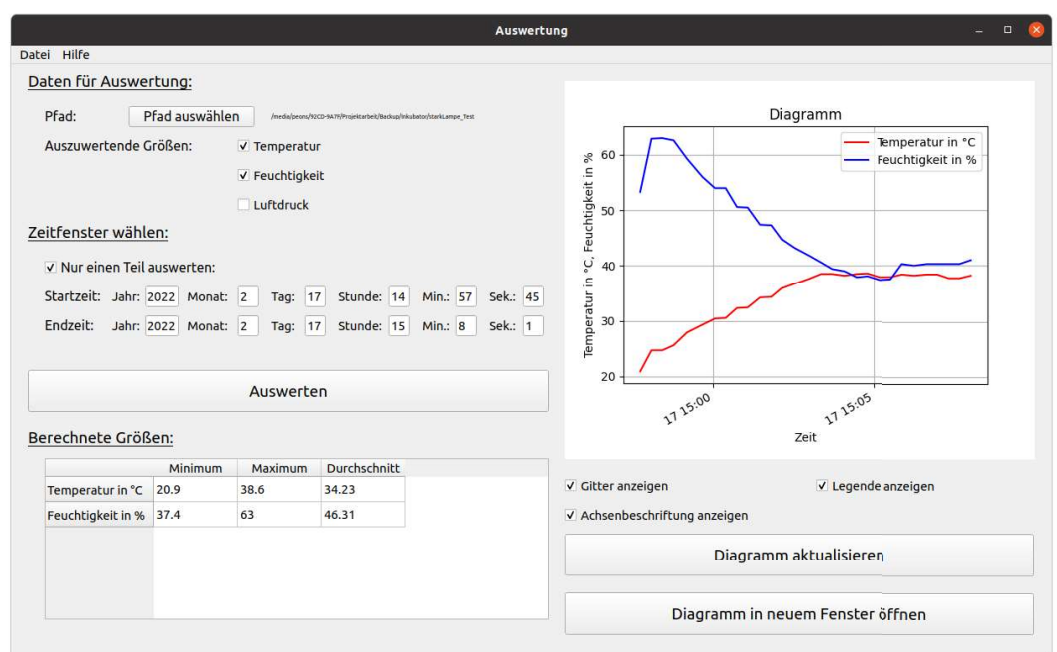


Bei diesem Bild handelt es sich um eine Messreihe von Temperatur und

Luftfeuchtigkeit über ca. 13 Stunden. Ziel war es, die Temperatur bei 38,3 °C und die Luftfeuchtigkeit bei 60 % zu halten. Es ist bereits am Durchschnitt zu sehen, dass dieses Ziel gut erreicht wurde. An den Graphen ist jedoch bereits zu erkennen, dass es Schwankungen gab. Um diese genauer zu untersuchen, kann man sich zum Beispiel nur die Temperatur anschauen und den Zeitraum begrenzen:



Zu diesem Zweck wurde das Häkchen unter „auszuwertende Größen“ nur bei „Temperatur“ gesetzt und der Zeitraum unter „Zeitfenster wählen“ auf den 18. Februar beschränkt.



Dieses Beispiel zeigt einen interessanteren Verlauf, bei dem von 20 °C auf 38 °C erwärmt wurde. Die Messung umfasst einen Zeitraum von ca. 10 Minuten. Der physikalische Zusammenhang zwischen Luftfeuchtigkeit und Temperatur lässt sich gut erkennen: Je höher die Temperatur, desto geringer die Luftfeuchtigkeit. Nach Einstellung des Zeitraums könnte man mit Hilfe der Tabelle (Minima und Maxima) außerdem die Temperaturdifferenz von Beginn bis Ende ermitteln und damit die Leistung der Wärmequelle errechnen.

## 4. Methoden

### 4.1. Messungen

Eine Messung lässt sich in drei wesentliche Arbeitsschritte aufteilen: Messen, Speichern und Warten. Nach Abschluss eines solchen Durchgangs wiederholt sich das Ganze. Beim Messen wird die entsprechende Bibliothek genutzt (siehe 3.2.), um die Messwerte in Variablen zwischen-zu-speichern. Daraufhin werden die Zahlen auf eine Nachkommastelle gerundet, da die Sensoren höher aufgelöste („längere“) Zahlen ausgeben, als es bei ihrer Genauigkeit sinnvoll wäre. Die „fertigen“ Messwerte werden dann zum Speichern weitergegeben. Dazu wird lediglich die jeweilige(n) Datei(en) geöffnet und der (/die) Messwert(e) angehängen. Wenn alle Messwerte gesichert sind, beginnt das Programm die festgelegte Zeit bis zur nächsten Messung zu warten. Dies wäre mit dem „sleep()“-Befehl möglich, der jedoch auch die gesamte GUI schlafen schickt. Daher wurde hierfür ein „QTimer“ genutzt, der eine Funktion mit den anderen Arbeitsschritten regelmäßig aufruft.

Für jeden Arbeitsschritt wurden unterschiedliche Python-Befehle genutzt:

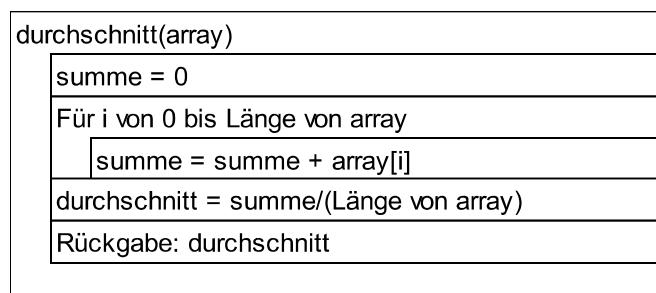
Arbeitsschritt	Python-Befehl	Erklärung
Messen	Siehe 3.2.  <code>round(zahl, n)</code>	Liefert Messwert  Rundet Zahl auf n Nachkommastellen
Speichern	<code>datei = open(pfad, „a“)</code>  <code>datei.write(messwerte)</code>	Öffnet Datei  Schreibt in Datei

	<code>datei.close()</code>	Schließt Datei
Warten	<code>timer = QTimer()</code> <code>timer.timeout.connect(messen)</code> <code>timer.start(n * 1000)</code>	Aufruf von „messen“ aller n Sekunden

#### 4.2. Auswertungen

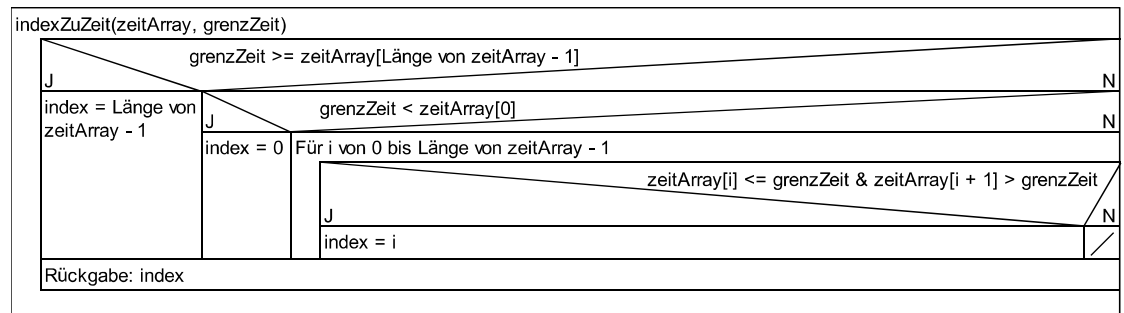
Bei der Auswertung steht zunächst das Einlesen der Messdaten im Vordergrund. Dies erfolgt analog zum Speichern, wobei „`datei.read()`“ zum Auslesen genutzt wird. Da alle Größen bereits in einzelnen Dateien gespeichert sind, müssen nur die einzelnen Messwerte jeder Größe durch den „`split()`“-Befehl getrennt werden. Daraufhin erfolgt einer Umwandlung der Messwerte von „Strings“ in „floats“ bzw. „datetimes“.

Für die Berechnung von Min- und Maxima gibt es bereits die in Python integrierten Funktionen „`min()`“ und „`max()`“. Für den Durchschnitt wurde jedoch eine Funktion nach folgendem Strukturgramm geschrieben:



Bei der Auswahl eines Zeitfensters ist es notwendig, nur einen Teil der Messdaten auswerten zu lassen. Dazu muss zu einer Grenz-Zeit (Start- bzw. Endzeit) der Index in dem Zeitarray gefunden werden, sodass alle Messwerte vor bzw. nach diesem Index nicht mehr ausgewertet werden. Dieser Index ist dabei der Index derjenigen Zeit im Zeitarray, die der Grenzzeit am nächsten liegt. Es müssen also alle Zeiten im Zeitarray iteriert werden, um zu prüfen, welches gerade noch kleiner bzw. größer als die Grenzzeit ist. Es könnte dabei aber auch vorkommen, dass die Grenzzeit deutlich kleiner/größer als das kleinste/größte Element des Zeitarrays ist. Folgendes Strukturgramm

verdeutlicht die Vorgehensweise:



## 5. Diskussion

Es ist zunächst festzustellen, dass alle Kann- und Soll-Kriterien der Planung vom 6.3.2022 erfüllt wurden. Es werden erfolgreich Messungen durchgeführt, die Messergebnisse gespeichert, wieder ausgelesen, wichtige Kenngrößen (Min-, Maxima und Durchschnitt) berechnet und ein Diagramm erstellt. Außerdem verfügt die Software über eine ansprechende GUI, kann den Zeitraum der Auswertung festlegen und nur bestimmte Größen auswerten. Des Weiteren wurde ein Kann-Kriterium mit den Kurzanleitungen, sowie den Links zu Tutorial-Seiten erfüllt. Auch das Exportieren des Diagramms ist über den Button „Diagramm in neuem Fenster öffnen“ und das erscheinende Fenster von „matplotlib“ möglich, wobei diese Funktionalität kaum als Ergebnis dieses Projektes angesehen werden kann. Das zweite Kann-Kriterium, weitere Größen zu den Messdaten auszurechnen, wurde nicht erreicht, da alle zunächst in Betracht gezogenen Größen als zu banal empfunden wurden. Die gewählten drei Größen scheinen am sinnvollsten zu sein. Eine Speicherung der Hardware-Konfiguration dagegen wurde aus Zeitgründen nicht implementiert.

Es gäbe noch einige Erweiterungsmöglichkeiten, die auch in der Planung nicht enthalten sind. Dazu zählt vor allem eine Funktion, die offensichtliche Messfehler erkennt. Gelegentlich kommt es vor, dass der Sensor kurzzeitig, meist nur für einen Messwert, einen vollkommen unrealistischen Wert liefert (am häufigsten:  $T = -50\text{ °C}$ ,  $F = 100\%$ ). Diese Messfehler können den Durchschnitt und Min- bzw. Maxima einer (kurzen) Messung vollkommen ändern, wogegen eine einfache Überprüfung Abhilfe schaffen würde.

Des Weiteren wäre es denkbar, die Software zur Steuerung von Heizelementen zu nutzen, die abhängig von der Temperatur ein- oder

ausgeschaltet werden. In Form eines Konsolen-Programmes wurde dies bereits umgesetzt (siehe 3.3., Beispiel 1 und 2).

Nichts desto trotz verlief das Projekt zur Erstellung einer Software für die Wetteranalyse äußerst erfolgreich. Das Programm hat die Erwartungen übertroffen und lässt sich in der Praxis vielfältig einsetzen, ist intuitiv zu bedienen und beugt Fehlern durch Eingaben vor.



## 6. Abbildungen

Startfenster:





Messungsfenster:

Wetteranalyse - Messungen

Datei Fenster Hilfe

Speicherort:

Pfad:

Pfad auswählen

/media/peons/92CD-9A7F/Projektarbeit/Test

Name:

Test

Messgerät:

☒ Test mit zufälligen Messwerten

Sensortyp:

DHT22

Sensor-Pin:

1

Messung:

Bitte zwei der drei Felder ausfüllen:

Intervall:

1

Abstand zwischen zwei Messungen (in sec.)

Dauer:

10

Dauer der gesamten Messreihe (in sec.)

Anzahl:

Anzahl an Messungen insgesamt.

Messung gestartet

Messung starten

Zeit: 2022-06-08 17:09:18.294011

Temperatur in °C: 35

Feuchtigkeit in %: 43

Zeit: 2022-06-08 17:09:19.293652

Temperatur in °C: 70

Feuchtigkeit in %: 76

Zeit: 2022-06-08 17:09:20.293797

Temperatur in °C: 63

Feuchtigkeit in %: 42

Messung abbrechen